Machine Learning

A Bayesian and Optimization Perspective

Academic Press, 2015

Sergios Theodoridis[1]

[1]Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece.

Spring 2017, Version III

Chapters 12 and 13
Bayesian Learning

Bayesian Learning

- According to the Bayesian path to machine learning, the unknown set of parameters are treated as random variables instead of a set of fixed (yet unknown) values.
- This was a revolutionary idea, at the time it was used by Laplace. Even now, after more than two centuries, it may seem strange to assume that a physical phenomenon/mechanism is controlled by a set of random parameters.
- However, there is a subtle point here.

Bayesian Learning

- According to the Bayesian path to machine learning, the unknown set of parameters are treated as random variables instead of a set of fixed (yet unknown) values.
- This was a revolutionary idea, at the time it was used by Laplace. Even now, after more than two centuries, it may seem strange to assume that a physical phenomenon/mechanism is controlled by a set of random parameters.
- However, there is a subtle point here.

Bayesian Learning

- According to the Bayesian path to machine learning, the unknown set of parameters are treated as random variables instead of a set of fixed (yet unknown) values.
- This was a revolutionary idea, at the time it was used by Laplace. Even now, after more than two centuries, it may seem strange to assume that a physical phenomenon/mechanism is controlled by a set of random parameters.
- However, there is a subtle point here.

Bayesian Learning

- A set of random parameters, θ, does not really imply a random nature for them.
- The associated randomness, in terms of a prior distribution, $p(\boldsymbol{\theta})$, encapsulates our **uncertainty** about their values, prior to receiving any measurements/observations.
- Put it in another way, the prior distribution represents our belief about the different possible values, although only one of them is actually true. From this perspective, probabilities are viewed in a more open-minded way, i.e., as measures of **uncertainty**.

Bayesian Learning

- A set of random parameters, θ, does not really imply a random nature for them.
- The associated randomness, in terms of a prior distribution, $p(\boldsymbol{\theta})$, encapsulates our **uncertainty** about their values, prior to receiving any measurements/observations.
- Put it in another way, the prior distribution represents our belief about the different possible values, although only one of them is actually true. From this perspective, probabilities are viewed in a more open-minded way, i.e., as measures of **uncertainty**.

Bayesian Learning

- A set of random parameters, θ, does not really imply a random nature for them.
- The associated randomness, in terms of a prior distribution, $p(\boldsymbol{\theta})$, encapsulates our **uncertainty** about their values, prior to receiving any measurements/observations.
- Put it in another way, the prior distribution represents our belief about the different possible values, although only one of them is actually true. From this perspective, probabilities are viewed in a more open-minded way, i.e., as measures of **uncertainty**.

Bayesian Learning

- Recall that parameter learning from data is an inverse problem. Basically, all we do is to deduce the "causes" (parameters) from the "effects" (observations).
- Bayes' theorem can be seen as such an inversion procedure expressed in a probabilistic context. Indeed, given the set of observations, say, $\mathcal{X}_o$, which are controlled by the unknown set of parameters, we write:

$$p(\boldsymbol{\theta}|\mathcal{X}_o) = \frac{p(\mathcal{X}_o|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{X}_o)}.$$

- All is needed for the above inversion is to have a **guess** about $p(\boldsymbol{\theta})$.

Bayesian Learning

- Recall that parameter learning from data is an inverse problem. Basically, all we do is to deduce the "causes" (parameters) from the "effects" (observations).
- Bayes' theorem can be seen as such an inversion procedure expressed in a probabilistic context. Indeed, given the set of observations, say, $\mathcal{X}_o$, which are controlled by the unknown set of parameters, we write:

$$p(\boldsymbol{\theta}|\mathcal{X}_o) = \frac{p(\mathcal{X}_o|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{X}_o)}.$$

- All is needed for the above inversion is to have a **guess** about $p(\boldsymbol{\theta})$.

Bayesian Learning

- Recall that parameter learning from data is an inverse problem. Basically, all we do is to deduce the "causes" (parameters) from the "effects" (observations).
- Bayes' theorem can be seen as such an inversion procedure expressed in a probabilistic context. Indeed, given the set of observations, say, $\mathcal{X}_o$, which are controlled by the unknown set of parameters, we write:

$$p(\boldsymbol{\theta}|\mathcal{X}_o) = \frac{p(\mathcal{X}_o|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{X}_o)}.$$

- All is needed for the above inversion is to have a **guess** about $p(\boldsymbol{\theta})$.

Bayesian Learning

- This term $p(\boldsymbol{\theta})$ has brought a lot of controversy. However, once a reasonable guess of the prior is available, a number of advantages associated with the Bayesian approach emerge, compared to the deterministic approaches, usually referred to as frequentist techniques.
- The term frequentist comes from the more classical view of probabilities as frequencies of occurrence of repeatable events.
- A typical example of this family of methods is the maximum likelihood approach, which estimates the values of the parameters by maximizing $p(\mathcal{X}_o|\theta)$; its value is solely controlled by the obtained observations in a sequence of experiments.

Bayesian Learning

- This term $p(\boldsymbol{\theta})$ has brought a lot of controversy. However, once a reasonable guess of the prior is available, a number of advantages associated with the Bayesian approach emerge, compared to the deterministic approaches, usually referred to as frequentist techniques.
- The term frequentist comes from the more classical view of probabilities as frequencies of occurrence of repeatable events.
- A typical example of this family of methods is the maximum likelihood approach, which estimates the values of the parameters by maximizing $p(\mathcal{X}_o|\theta)$; its value is solely controlled by the obtained observations in a sequence of experiments.

Bayesian Learning

- This term $p(\boldsymbol{\theta})$ has brought a lot of controversy. However, once a reasonable guess of the prior is available, a number of advantages associated with the Bayesian approach emerge, compared to the deterministic approaches, usually referred to as frequentist techniques.
- The term frequentist comes from the more classical view of probabilities as frequencies of occurrence of repeatable events.
- A typical example of this family of methods is the maximum likelihood approach, which estimates the values of the parameters by maximizing $p(\mathcal{X}_o|\boldsymbol{\theta})$; its value is solely controlled by the obtained observations in a sequence of experiments.

Regression: A Probabilistic Perspective

- Let us consider the (generalized) linear regression task, i.e.,

$$\mathrm{y} = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}) + \eta = \theta_0 + \sum_{k=1}^{K-1} \theta_k \phi_k(\mathbf{x}) + \eta,$$

where $\mathrm{y} \in \mathbb{R}$ is the output random variable, $\mathbf{x} \in \mathbb{R}^l$ is the input random vector, $\eta \in \mathbb{R}$ is the noise disturbance, $\boldsymbol{\theta} \in \mathbb{R}^K$ is the unknown parameter vector and

$$\boldsymbol{\phi}(\mathbf{x}) := [\phi_1(\mathbf{x}), \ldots, \phi_{K-1}(\mathbf{x}), 1]^T$$

where $\phi_k(\cdot),\ k = 1, \ldots, K-1$, are some (fixed) basis functions. We are given a set of N training points, $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, \ldots, N$.

- We assume that the respective (unobserved) noise samples, $\eta_n,\ n = 1, 2, \ldots, N$, correspond to a jointly Gaussian pdf with covariance matrix Σ_η, i.e.,

$$p(\boldsymbol{\eta}) = \frac{1}{(2\pi)^{N/2} |\Sigma_\eta|^{1/2}} \exp\left(-\frac{1}{2} \boldsymbol{\eta}^T \Sigma_\eta^{-1} \boldsymbol{\eta}\right).$$

Regression: A Probabilistic Perspective

- Let us consider the (generalized) linear regression task, i.e.,

$$\mathrm{y} = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}) + \eta = \theta_0 + \sum_{k=1}^{K-1} \theta_k \phi_k(\mathbf{x}) + \eta,$$

where $\mathrm{y} \in \mathbb{R}$ is the output random variable, $\mathbf{x} \in \mathbb{R}^l$ is the input random vector, $\eta \in \mathbb{R}$ is the noise disturbance, $\boldsymbol{\theta} \in \mathbb{R}^K$ is the unknown parameter vector and

$$\boldsymbol{\phi}(\mathbf{x}) := [\phi_1(\mathbf{x}), \ldots, \phi_{K-1}(\mathbf{x}), 1]^T$$

where $\phi_k(\cdot),\ k = 1, \ldots, K-1$, are some (fixed) basis functions. We are given a set of N training points, $(y_n, \boldsymbol{x}_n)$, $n = 1, 2, \ldots, N$.

- We assume that the respective (unobserved) noise samples, $\eta_n,\ n = 1, 2, \ldots, N$, correspond to a jointly Gaussian pdf with covariance matrix Σ_η, i.e.,

$$p(\boldsymbol{\eta}) = \frac{1}{(2\pi)^{N/2} |\Sigma_\eta|^{1/2}} \exp\left(-\frac{1}{2} \boldsymbol{\eta}^T \Sigma_\eta^{-1} \boldsymbol{\eta}\right).$$

The Maximum Likelihood Method

- According to the ML method, the unknown parameter is treated as a deterministic variable $\boldsymbol{\theta}$, which parameterizes the pdf that describes the output vector of observations,

$$\boldsymbol{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\eta},$$

where

$$\Phi = \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \boldsymbol{\phi}^T(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_N) \end{bmatrix}, \text{ and } \boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T.$$

- Thus, $p(\boldsymbol{\eta}) = p(\boldsymbol{y} - \Phi\boldsymbol{\theta})$. Optimizing $p(\boldsymbol{\eta})$, w.r. to $\boldsymbol{\theta}$, the ML estimate results as,

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \left(\Phi^T \Sigma_\eta^{-1} \Phi\right)^{-1} \Phi^T \Sigma_\eta^{-1} \boldsymbol{y}.$$

The Maximum Likelihood Method

- According to the ML method, the unknown parameter is treated as a deterministic variable $\boldsymbol{\theta}$, which parameterizes the pdf that describes the output vector of observations,

$$\boldsymbol{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\eta},$$

where

$$\Phi = \begin{bmatrix} \boldsymbol{\phi}^T(\boldsymbol{x}_1) \\ \boldsymbol{\phi}^T(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{\phi}^T(\boldsymbol{x}_N) \end{bmatrix}, \text{ and } \boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T.$$

- Thus, $p(\boldsymbol{\eta}) = p(\boldsymbol{y} - \Phi\boldsymbol{\theta})$. Optimizing $p(\boldsymbol{\eta})$, w.r. to $\boldsymbol{\theta}$, the ML estimate results as,

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \left(\Phi^T \Sigma_\eta^{-1} \Phi\right)^{-1} \Phi^T \Sigma_\eta^{-1} \boldsymbol{y}.$$

The Maximum Likelihood Method

- For the simple case of a white noise sequence of variance σ_η^2 ($\Sigma_\eta = \sigma_\eta^2 I$), we get the LS solution,

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \boldsymbol{y} = \hat{\boldsymbol{\theta}}_{\mathrm{LS}}.$$

- A major drawback of the ML approach is that it is vulnerable to overfitting, since no care is taken for complex models that try to "learn" the specificities of the particular training set.

The Maximum Likelihood Method

- For the simple case of a white noise sequence of variance σ_η^2 ($\Sigma_\eta = \sigma_\eta^2 I$), we get the LS solution,

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \boldsymbol{y} = \hat{\boldsymbol{\theta}}_{\mathrm{LS}}.$$

- A major drawback of the ML approach is that it is vulnerable to overfitting, since no care is taken for complex models that try to "learn" the specificities of the particular training set.

The MAP Estimator: A Revision

- According to the MAP, the unknown set of parameters are treated as a random vector, θ, and its posterior, for a given set of output observations, $\boldsymbol{y}$, is expressed as

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})},$$

  where $p(\boldsymbol{\theta})$ is the associated prior pdf. The notation has been relaxed on the dependence on the set of observations, $\mathcal{X}$. The input set, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, is considered fixed, so all the randomness associated with $\boldsymbol{y}$ is due to the **noise source**.

- In MAP, we are only interested in the maximum. Hence, since the denominator on the right hand side does not depend on θ, it can be ignored.

- Note that ignoring $p(\boldsymbol{y})$ is a major difference compared to the Bayesian approach, and important information that **resides** in the denominator is not exploited.

The MAP Estimator: A Revision

- According to the MAP, the unknown set of parameters are treated as a random vector, Θ, and its posterior, for a given set of output observations, $\boldsymbol{y}$, is expressed as

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})},$$

  where $p(\boldsymbol{\theta})$ is the associated prior pdf. The notation has been relaxed on the dependence on the set of observations, $\mathcal{X}$. The input set, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, is considered fixed, so all the randomness associated with $\boldsymbol{y}$ is due to the **noise source**.

- In MAP, we are only interested in the maximum. Hence, since the denominator on the right hand side does not depend on $\boldsymbol{\theta}$, it can be ignored.

- Note that ignoring $p(\boldsymbol{y})$ is a major difference compared to the Bayesian approach, and important information that **resides** in the denominator is not exploited.

The MAP Estimator: A Revision

- According to the MAP, the unknown set of parameters are treated as a random vector, Θ, and its posterior, for a given set of output observations, $\boldsymbol{y}$, is expressed as

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})},$$

  where $p(\boldsymbol{\theta})$ is the associated prior pdf. The notation has been relaxed on the dependence on the set of observations, $\mathcal{X}$. The input set, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, is considered fixed, so all the randomness associated with $\boldsymbol{y}$ is due to the **noise source**.

- In MAP, we are only interested in the maximum. Hence, since the denominator on the right hand side does not depend on $\boldsymbol{\theta}$, it can be ignored.

- Note that ignoring $p(\boldsymbol{y})$ is a major difference compared to the Bayesian approach, and important information that **resides** in the denominator is not exploited.

The MAP Estimator: A Revision

- Assuming both the prior as well as the conditional pdfs to be Gaussians, i.e.,

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_0, \Sigma_\theta) \text{ and } p(\boldsymbol{y}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\Phi\boldsymbol{\theta}, \Sigma_\eta),$$

  the posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$ turns out also to be Gaussian with mean vector,

$$\boldsymbol{\mu}_{\theta|y} := \mathbb{E}[\theta|\boldsymbol{y}] = \boldsymbol{\theta}_0 + \left(\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi\right)^{-1}\Phi^T\Sigma_\eta^{-1}\left(\boldsymbol{y} - \Phi\boldsymbol{\theta}_0\right).$$

- Since in the MAP, we are only interested in the maximum, for a Gaussian this coincides with its mean, and we have that

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \mathbb{E}[\theta|\boldsymbol{y}].$$

The MAP Estimator: A Revision

- Assuming both the prior as well as the conditional pdfs to be Gaussians, i.e.,

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_0, \Sigma_\theta) \text{ and } p(\boldsymbol{y}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\Phi\boldsymbol{\theta}, \Sigma_\eta),$$

  the posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$ turns out also to be Gaussian with mean vector,

$$\boldsymbol{\mu}_{\theta|y} := \mathbb{E}[\theta|\boldsymbol{y}] = \boldsymbol{\theta}_0 + \left(\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi\right)^{-1}\Phi^T\Sigma_\eta^{-1}\left(\boldsymbol{y} - \Phi\boldsymbol{\theta}_0\right).$$

- Since in the MAP, we are only interested in the maximum, for a Gaussian this coincides with its mean, and we have that

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \mathbb{E}[\theta|\boldsymbol{y}].$$

The Prior PDF Acts As A Regularizer

- Treating the parameters as random variables, regularization is achieved via $\boldsymbol{\theta}_0$ and Σ_θ, which are imposed by the prior $p(\boldsymbol{\theta})$.
- We can verify it by establishing a bridge with the ridge regression. Let us assume that $\Sigma_\theta = \sigma_\theta^2 I$, $\Sigma_\eta = \sigma_\eta^2 I$ and $\boldsymbol{\theta}_0 = \mathbf{0}$. Then the previous formula becomes,

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \left(\lambda I + \Phi^T \Phi\right)^{-1} \Phi^T \boldsymbol{y},$$

where we have set $\lambda := \frac{\sigma_\eta^2}{\sigma_\theta^2}$. This is the same as the solution resulting from the ridge regression, i.e.,

$$J(\boldsymbol{\theta}, \lambda) = \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\theta}\|^2.$$

The Prior PDF Acts As A Regularizer

- Treating the parameters as random variables, regularization is achieved via $\boldsymbol{\theta}_0$ and Σ_θ, which are imposed by the prior $p(\boldsymbol{\theta})$.
- We can verify it by establishing a bridge with the ridge regression. Let us assume that $\Sigma_\theta = \sigma_\theta^2 I$, $\Sigma_\eta = \sigma_\eta^2 I$ and $\boldsymbol{\theta}_0 = \mathbf{0}$. Then the previous formula becomes,

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \left(\lambda I + \Phi^T \Phi\right)^{-1} \Phi^T \boldsymbol{y},$$

where we have set $\lambda := \frac{\sigma_\eta^2}{\sigma_\theta^2}$. This is the same as the solution resulting from the ridge regression, i.e.,

$$J(\boldsymbol{\theta}, \lambda) = \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\theta}\|^2.$$

The Prior PDF Acts As A Regularizer

- Treating the parameters as random variables, regularization is achieved via $\boldsymbol{\theta}_0$ and Σ_θ, which are imposed by the prior $p(\boldsymbol{\theta})$.
- We can verify it by establishing a bridge with the ridge regression. Let us assume that $\Sigma_\theta = \sigma_\theta^2 I$, $\Sigma_\eta = \sigma_\eta^2 I$ and $\boldsymbol{\theta}_0 = \mathbf{0}$. Then the previous formula becomes,

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \left(\lambda I + \Phi^T \Phi\right)^{-1} \Phi^T \boldsymbol{y},$$

where we have set $\lambda := \frac{\sigma_\eta^2}{\sigma_\theta^2}$. This is the same as the solution resulting from the ridge regression, i.e.,

$$J(\boldsymbol{\theta}, \lambda) = \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2 + \lambda\|\boldsymbol{\theta}\|^2.$$

The Prior PDF Acts As A Regularizer

- Choosing the value of λ is critical to the performance of the estimator. The main issue now becomes how to choose a good value for λ, or equivalently for $\Sigma_\theta,\ \Sigma_\eta$ in the more general case.
- In practice, the cross-validation method is adopted; different values of λ are tested and the one that leads to the best MSE (or some other criterion) is selected.
- This is a computationally costly procedure. Note that cross-validation requires the use for training of only a fraction of the available data, so that to reserve the rest for testing.

The Prior PDF Acts As A Regularizer

- Choosing the value of λ is critical to the performance of the estimator. The main issue now becomes how to choose a good value for λ, or equivalently for $\Sigma_\theta,\ \Sigma_\eta$ in the more general case.
- In practice, the cross-validation method is adopted; different values of λ are tested and the one that leads to the best MSE (or some other criterion) is selected.
- This is a computationally costly procedure. Note that cross-validation requires the use for training of only a fraction of the available data, so that to reserve the rest for testing.

The Prior PDF Acts As A Regularizer

- Choosing the value of λ is critical to the performance of the estimator. The main issue now becomes how to choose a good value for λ, or equivalently for $\Sigma_\theta,\ \Sigma_\eta$ in the more general case.
- In practice, the cross-validation method is adopted; different values of λ are tested and the one that leads to the best MSE (or some other criterion) is selected.
- This is a computationally costly procedure. Note that cross-validation requires the use for training of only a fraction of the available data, so that to reserve the rest for testing.

The Prior PDF Acts As A Regularizer

- Choosing the value of λ is critical to the performance of the estimator. The main issue now becomes how to choose a good value for λ, or equivalently for $\Sigma_\theta,\ \Sigma_\eta$ in the more general case.
- In practice, the cross-validation method is adopted; different values of λ are tested and the one that leads to the best MSE (or some other criterion) is selected.
- This is a computationally costly procedure. Note that cross-validation requires the use for training of only a fraction of the available data, so that to reserve the rest for testing.

The Bayesian Approach

- In the Bayesian approach, all the involved parameters can be estimated on the training set. In this vein, the parameters will be treated as random variables.
- At the same time, the goal now becomes to **infer** the pdf that describes the unknown set of parameters, instead of obtaining a single vector estimate of parameters.
- Thus, one has more information at his/her disposal. Having said all that, it does not mean that Bayesian techniques are necessarily free from cross-validation; this is needed to assess their overall performance.

The Bayesian Approach

- In the Bayesian approach, all the involved parameters can be estimated on the training set. In this vein, the parameters will be treated as random variables.
- At the same time, the goal now becomes to **infer** the pdf that describes the unknown set of parameters, instead of obtaining a single vector estimate of parameters.
- Thus, one has more information at his/her disposal. Having said all that, it does not mean that Bayesian techniques are necessarily free from cross-validation; this is needed to assess their overall performance.

The Bayesian Approach

- In the Bayesian approach, all the involved parameters can be estimated on the training set. In this vein, the parameters will be treated as random variables.
- At the same time, the goal now becomes to **infer** the pdf that describes the unknown set of parameters, instead of obtaining a single vector estimate of parameters.
- Thus, one has more information at his/her disposal. Having said all that, it does not mean that Bayesian techniques are necessarily free from cross-validation; this is needed to assess their overall performance.

The Bayesian Approach

- The starting point is the same as that for MAP, that is,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})}.$$

- However, instead of taking just the maximum of the numerator, we will make use of $p(\boldsymbol{\theta}|\boldsymbol{y})$ as a whole. As a matter of fact, most of the secrets lie in the denominator, $p(\boldsymbol{y})$, which is basically the **normalizing constant**,

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}.$$

The Bayesian Approach

- The starting point is the same as that for MAP, that is,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})}.$$

- However, instead of taking just the maximum of the numerator, we will make use of $p(\boldsymbol{\theta}|\boldsymbol{y})$ as a whole. As a matter of fact, most of the secrets lie in the denominator, $p(\boldsymbol{y})$, which is basically the **normalizing constant**,

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}.$$

The Bayesian Approach

- The difficulty with $p(\boldsymbol{y})$ is that, in general, the evaluation of the corresponding integral cannot be performed analytically. In such cases, one has to resort to approximate techniques. To this end, a number of techniques are available. Various alternatives for the computation of this integral are:
  - The Laplacian approximation method.
  - The variational approximation method.
  - The variational bound approximation method.
  - Monte Carlo techniques for the evaluation of the integral.
  - Message passing algorithms in the context of probabilistic graphical models.

The Bayesian Approach

- The difficulty with $p(\boldsymbol{y})$ is that, in general, the evaluation of the corresponding integral cannot be performed analytically. In such cases, one has to resort to approximate techniques. To this end, a number of techniques are available. Various alternatives for the computation of this integral are:
  - The Laplacian approximation method.
  - The variational approximation method.
  - The variational bound approximation method.
  - Monte Carlo techniques for the evaluation of the integral.
  - Message passing algorithms in the context of probabilistic graphical models.

The Bayesian Approach

- The difficulty with $p(\boldsymbol{y})$ is that, in general, the evaluation of the corresponding integral cannot be performed analytically. In such cases, one has to resort to approximate techniques. To this end, a number of techniques are available. Various alternatives for the computation of this integral are:
  - The Laplacian approximation method.
  - The variational approximation method.
  - The variational bound approximation method.
  - Monte Carlo techniques for the evaluation of the integral.
  - Message passing algorithms in the context of probabilistic graphical models.

The Bayesian Approach

- The difficulty with $p(\boldsymbol{y})$ is that, in general, the evaluation of the corresponding integral cannot be performed analytically. In such cases, one has to resort to approximate techniques. To this end, a number of techniques are available. Various alternatives for the computation of this integral are:
  - The Laplacian approximation method.
  - The variational approximation method.
  - The variational bound approximation method.
  - Monte Carlo techniques for the evaluation of the integral.
  - Message passing algorithms in the context of probabilistic graphical models.

The Bayesian Approach

- The difficulty with $p(\boldsymbol{y})$ is that, in general, the evaluation of the corresponding integral cannot be performed analytically. In such cases, one has to resort to approximate techniques. To this end, a number of techniques are available. Various alternatives for the computation of this integral are:
  - The Laplacian approximation method.
  - The variational approximation method.
  - The variational bound approximation method.
  - Monte Carlo techniques for the evaluation of the integral.
  - Message passing algorithms in the context of probabilistic graphical models.

The Bayesian Approach

- The difficulty with $p(\boldsymbol{y})$ is that, in general, the evaluation of the corresponding integral cannot be performed analytically. In such cases, one has to resort to approximate techniques. To this end, a number of techniques are available. Various alternatives for the computation of this integral are:
  - The Laplacian approximation method.
  - The variational approximation method.
  - The variational bound approximation method.
  - Monte Carlo techniques for the evaluation of the integral.
  - Message passing algorithms in the context of probabilistic graphical models.

The Bayesian Approach

- As a first step, for pedagogical purposes, we start with the case where the integral can be computed analytically. Let us assume that $p(\boldsymbol{y}|\boldsymbol{\theta})$ and $p(\boldsymbol{\theta})$ are both Gaussians. Such an assumption renders $p(\boldsymbol{y})$ to be also a Gaussian one. Moreover, the respective mean and covariance matrix can be obtained **analytically**.

The Bayesian Approach to Regression: The Full Gaussian Case

- Assuming that the prior and the conditional are Gaussians, i.e.,

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_0, \Sigma_\theta) \text{ and } p(\boldsymbol{y}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\Phi\boldsymbol{\theta}, \Sigma_\eta),$$

it turns out that:

- The normalizing constant is given by,

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}|\Phi\boldsymbol{\theta}_0, \Sigma_\eta + \Phi\Sigma_\theta\Phi^T\right). \tag{1}$$

- The resulting posterior pdf is also Gaussian, i.e.,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y}\right), \tag{2}$$

where $\boldsymbol{\mu}_{\theta|y}$ is given as in the MAP estimator,

$$\boldsymbol{\mu}_{\theta|y} = \boldsymbol{\theta}_0 + \left(\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi\right)^{-1}\Phi^T\Sigma_\eta^{-1}\left(\boldsymbol{y} - \Phi\boldsymbol{\theta}_0\right),$$

and the corresponding covariance matrix is equal to,

$$\Sigma_{\theta|y} = \left(\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi\right)^{-1}.$$

The Bayesian Approach to Regression: The Full Gaussian Case

- Assuming that the prior and the conditional are Gaussians, i.e.,

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_0, \Sigma_\theta) \text{ and } p(\boldsymbol{y}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\Phi\boldsymbol{\theta}, \Sigma_\eta),$$

it turns out that:

- The normalizing constant is given by,

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}|\Phi\boldsymbol{\theta}_0, \Sigma_\eta + \Phi\Sigma_\theta\Phi^T\right). \tag{1}$$

- The resulting posterior pdf is also Gaussian, i.e.,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y}\right), \tag{2}$$

where $\boldsymbol{\mu}_{\theta|y}$ is given as in the MAP estimator,

$$\boldsymbol{\mu}_{\theta|y} = \boldsymbol{\theta}_0 + \left(\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi\right)^{-1}\Phi^T\Sigma_\eta^{-1}\left(\boldsymbol{y} - \Phi\boldsymbol{\theta}_0\right),$$

and the corresponding covariance matrix is equal to,

$$\Sigma_{\theta|y} = \left(\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi\right)^{-1}.$$

The Bayesian Approach to Regression: The Full Gaussian Case

- Assuming that the prior and the conditional are Gaussians, i.e.,

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_0, \Sigma_\theta) \text{ and } p(\boldsymbol{y}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\Phi\boldsymbol{\theta}, \Sigma_\eta),$$

  it turns out that:

  - The normalizing constant is given by,

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}|\Phi\boldsymbol{\theta}_0, \Sigma_\eta + \Phi\Sigma_\theta\Phi^T\right). \tag{1}$$

  - The resulting posterior pdf is also Gaussian, i.e.,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y}\right), \tag{2}$$

    where $\boldsymbol{\mu}_{\theta|y}$ is given as in the MAP estimator,

$$\boldsymbol{\mu}_{\theta|y} = \boldsymbol{\theta}_0 + \left(\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi\right)^{-1}\Phi^T\Sigma_\eta^{-1}\left(\boldsymbol{y} - \Phi\boldsymbol{\theta}_0\right),$$

    and the corresponding covariance matrix is equal to,

$$\Sigma_{\theta|y} = \left(\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi\right)^{-1}.$$

The Bayesian Approach to Regression: The Full Gaussian Case

- The posterior pdf provides our knowledge about θ, **after** the observations $\boldsymbol{y}$ have been obtained. Hence, our uncertainty about θ has been reduced.
- This explains why the posterior is different to the prior pdf; the latter represents only our initial guess. The covariance matrix of the posterior provides information about our uncertainty w.r. to θ.

The Bayesian Approach to Regression: The Full Gaussian Case

- The posterior pdf provides our knowledge about θ, **after** the observations y have been obtained. Hence, our uncertainty about θ has been reduced.
- This explains why the posterior is different to the prior pdf; the latter represents only our initial guess. The covariance matrix of the posterior provides information about our uncertainty w.r. to θ.

Regression: Inference On The Output Variable Directly

- Recall that the ultimate goal of a regression model is to predict the output value, $\hat{y}$, given the corresponding value of the input vector, $\boldsymbol{x}$. The Bayesian philosophy provides the means for a **direct** inference of the output variable.
- In such cases, estimating a value for the unknown θ is only the means to an end. To formulate the prediction task directly, without involving θ, one has to integrate out the contribution of θ.

Regression: Inference On The Output Variable Directly

- Recall that the ultimate goal of a regression model is to predict the output value, $\hat{y}$, given the corresponding value of the input vector, $\boldsymbol{x}$. The Bayesian philosophy provides the means for a **direct** inference of the output variable.
- In such cases, estimating a value for the unknown $\boldsymbol{\theta}$ is only the means to an end. To formulate the prediction task directly, without involving $\boldsymbol{\theta}$, one has to integrate out the contribution of $\boldsymbol{\theta}$.

Regression: Inference On The Output Variable Directly

- Having learned the posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$, then given a new input vector $\boldsymbol{x}$, the conditional pdf of the output variable, y, given the set of observations, $\boldsymbol{y}$, is written as,

$$p(y|\boldsymbol{x},\boldsymbol{y}) = \int p(y|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}. \tag{3}$$

Note that we have written $p(y|\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) = p(y|\boldsymbol{x},\boldsymbol{\theta})$ since y is conditionally independent of $\boldsymbol{y}$ given the value of $\boldsymbol{\theta}$. Strictly speaking, the posterior should have been denoted as $p(\boldsymbol{\theta}|\boldsymbol{y};\mathcal{X})$ to indicate the dependence on the input training samples. However, the dependence on $\mathcal{X}$ has been suppressed to unclutter notation.

Regression: Inference On The Output Variable Directly

- Having learned the posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$, then given a new input vector $\boldsymbol{x}$, the conditional pdf of the output variable, y, given the set of observations, $\boldsymbol{y}$, is written as,

$$p(y|\boldsymbol{x},\boldsymbol{y}) = \int p(y|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}. \tag{3}$$

Note that we have written $p(y|\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) = p(y|\boldsymbol{x},\boldsymbol{\theta})$ since y is conditionally independent of $\boldsymbol{y}$ given the value of $\boldsymbol{\theta}$. Strictly speaking, the posterior should have been denoted as $p(\theta|\boldsymbol{y};\mathcal{X})$ to indicate the dependence on the input training samples. However, the dependence on $\mathcal{X}$ has been suppressed to unclutter notation.

Regression: Inference On The Output Variable Directly

- Having learned the posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$, then given a new input vector $\boldsymbol{x}$, the conditional pdf of the output variable, y, given the set of observations, $\boldsymbol{y}$, is written as,

$$p(y|\boldsymbol{x},\boldsymbol{y}) = \int p(y|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}. \tag{3}$$

Note that we have written $p(y|\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) = p(y|\boldsymbol{x},\boldsymbol{\theta})$ since y is conditionally independent of $\boldsymbol{y}$ given the value of $\boldsymbol{\theta}$. Strictly speaking, the posterior should have been denoted as $p(\boldsymbol{\theta}|\boldsymbol{y};\mathcal{X})$ to indicate the dependence on the input training samples. However, the dependence on $\mathcal{X}$ has been suppressed to unclutter notation.

Regression: Inference On The Output Variable Directly

- In order to simplify algebra and focus on the concepts, assume that $\Sigma_\eta = \sigma_\eta^2 I$. Also for the prior pdf, $\Sigma_\theta = \sigma_\theta^2 I$. Then, the conditional of the output variable takes the form,

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}), \sigma_\eta^2).$$

- Also, the mean and covariance matrix of the respective (Gaussian) posterior are simplified to,

$$\boldsymbol{\mu}_{\theta|y} = \boldsymbol{\theta}_0 + \frac{1}{\sigma_\eta^2}\left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1} \Phi^T (\boldsymbol{y} - \Phi\boldsymbol{\theta}_0), \tag{4}$$

$$\Sigma_{\theta|y} = \left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1}. \tag{5}$$

Regression: Inference On The Output Variable Directly

- In order to simplify algebra and focus on the concepts, assume that $\Sigma_\eta = \sigma_\eta^2 I$. Also for the prior pdf, $\Sigma_\theta = \sigma_\theta^2 I$. Then, the conditional of the output variable takes the form,

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}), \sigma_\eta^2).$$

- Also, the mean and covariance matrix of the respective (Gaussian) posterior are simplified to,

$$\boldsymbol{\mu}_{\theta|y} = \boldsymbol{\theta}_0 + \frac{1}{\sigma_\eta^2}\left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1} \Phi^T\left(\boldsymbol{y} - \Phi\boldsymbol{\theta}_0\right), \tag{4}$$

$$\Sigma_{\theta|y} = \left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1}. \tag{5}$$

Regression: Inference On The Output Variable Directly

- Plugging the above in (3), results in

$$p(y|\boldsymbol{x},\boldsymbol{y}) = \mathcal{N}\left(y|\mu_y,\sigma_y^2\right),$$

where

$$\mu_y = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\mu}_{\theta|y},\ \ \sigma_y^2 = \sigma_\eta^2+\sigma_\eta^2\sigma_\theta^2\boldsymbol{\phi}^T(\boldsymbol{x})\left(\sigma_\eta^2 I+\sigma_\theta^2\Phi^T\Phi\right)^{-1}\boldsymbol{\phi}(\boldsymbol{x}). \tag{6}$$

- Hence, given x, one can predict the respective value of y using the most probable value, i.e., μ_y. Note that the same prediction value would result via the MAP estimate if $\theta_0 = 0$. Have we then gained anything extra by adopting the Bayesian approach? YES!
- More information concerning the predicted value is now available, since we have an estimate of the respective variance, which quantifies the associated uncertainty of the prediction.

Regression: Inference On The Output Variable Directly

- Plugging the above in (3), results in

$$p(y|\boldsymbol{x},\boldsymbol{y}) = \mathcal{N}\left(y|\mu_y,\sigma_y^2\right),$$

where

$$\mu_y = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\mu}_{\theta|y},\ \ \sigma_y^2 = \sigma_\eta^2+\sigma_\eta^2\sigma_\theta^2\boldsymbol{\phi}^T(\boldsymbol{x})\left(\sigma_\eta^2 I+\sigma_\theta^2\Phi^T\Phi\right)^{-1}\boldsymbol{\phi}(\boldsymbol{x}). \tag{6}$$

- Hence, given $\boldsymbol{x}$, one can predict the respective value of y using the most probable value, i.e., μ_y. Note that the same prediction value would result via the MAP estimate if $\boldsymbol{\theta_0} = \mathbf{0}$. Have we then gained anything extra by adopting the Bayesian approach? YES!
- More information concerning the predicted value is now available, since we have an estimate of the respective variance, which quantifies the associated uncertainty of the prediction.

Regression: Inference On The Output Variable Directly

- Plugging the above in (3), results in

$$p(y|\boldsymbol{x},\boldsymbol{y}) = \mathcal{N}\left(y|\mu_y,\sigma_y^2\right),$$

where

$$\mu_y = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\mu}_{\theta|y},\ \ \sigma_y^2 = \sigma_\eta^2+\sigma_\eta^2\sigma_\theta^2\boldsymbol{\phi}^T(\boldsymbol{x})\left(\sigma_\eta^2 I+\sigma_\theta^2\Phi^T\Phi\right)^{-1}\boldsymbol{\phi}(\boldsymbol{x}). \tag{6}$$

- Hence, given $\boldsymbol{x}$, one can predict the respective value of y using the most probable value, i.e., μ_y. Note that the same prediction value would result via the MAP estimate if $\boldsymbol{\theta_0} = \mathbf{0}$. Have we then gained anything extra by adopting the Bayesian approach? YES!
- More information concerning the predicted value is now available, since we have an estimate of the respective variance, which quantifies the associated uncertainty of the prediction.

Regression: Inference On The Output Variable Directly

- To investigate our task further, let us simplify it via the following approximation, in terms of the autocorrelation matrix of $\boldsymbol{\phi}(\mathbf{x})$, R_ϕ, i.e.,

$$R_\phi := \mathbb{E}[\boldsymbol{\phi}(\mathbf{x})\boldsymbol{\phi}^T(\mathbf{x})] \simeq \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}^T(\boldsymbol{x}_n) = \frac{1}{N}\Phi^T\Phi,$$

or

$$\Phi^T\Phi \simeq NR_\phi.$$

- Plugging this approximation into the variance formula, readily results in

$$\sigma_y^2 \simeq \sigma_\eta^2\left(1+\sigma_\theta^2\boldsymbol{\phi}^T(\boldsymbol{x})\left(\sigma_\eta^2 I + N\sigma_\theta^2 R_\phi\right)^{-1}\boldsymbol{\phi}(\boldsymbol{x})\right),$$

which for large values of N becomes

$$\sigma_y^2 \simeq \sigma_\eta^2\left(1+\frac{1}{N}\boldsymbol{\phi}^T(\boldsymbol{x})R_\phi^{-1}\boldsymbol{\phi}(\boldsymbol{x})\right).$$

Regression: Inference On The Output Variable Directly

- To investigate our task further, let us simplify it via the following approximation, in terms of the autocorrelation matrix of $\boldsymbol{\phi}(\mathbf{x})$, R_ϕ, i.e.,

$$R_\phi := \mathbb{E}[\boldsymbol{\phi}(\mathbf{x})\boldsymbol{\phi}^T(\mathbf{x})] \simeq \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\boldsymbol{\phi}^T(\boldsymbol{x}_n) = \frac{1}{N}\Phi^T\Phi,$$

or

$$\Phi^T\Phi \simeq NR_\phi.$$

- Plugging this approximation into the variance formula, readily results in

$$\sigma_y^2 \simeq \sigma_\eta^2\left(1+\sigma_\theta^2\boldsymbol{\phi}^T(\boldsymbol{x})\left(\sigma_\eta^2 I + N\sigma_\theta^2 R_\phi\right)^{-1}\boldsymbol{\phi}(\boldsymbol{x})\right),$$

which for large values of N becomes

$$\sigma_y^2 \simeq \sigma_\eta^2\left(1+\frac{1}{N}\boldsymbol{\phi}^T(\boldsymbol{x})R_\phi^{-1}\boldsymbol{\phi}(\boldsymbol{x})\right).$$

Regression: Inference On The Output Variable Directly

- Thus, for a large number of observations, $\sigma_y^2 \to \sigma_\eta^2$, and our uncertainty is contributed by the noise source, which **cannot be reduced further**. For smaller values of N, there is extra uncertainty, which is associated with the parameter θ, measured by σ_θ^2.

Regression: Inference On The Output Variable Directly

- So far, we dealt with Gaussians, which led to tractable and analytically computed integrals. Moreover, even in the case of Gaussian pdfs, we have assumed the covariance matrices $\Sigma_\theta, \Sigma_\eta$ to be known. In practice, they are not. Can one select the related parameters via an optimization process?

Regression: Inference On The Output Variable Directly

- If the answer is yes, can this optimization be carried out on the training set, or one would necessarily run into problems similar to the ones we faced with the regularization approach? We will indulge in all these challenges soon.

Example on Bayesian Regression

- In this example, we focus on inferring the output directly, after integrating out the parameters. The simplified full Gaussian case will be considered. Data are generated based on the following nonlinear model,

$$y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \theta_3 x_n^3 + \theta_5 x_n^5 + \eta_n, \ n = 1, 2, \ldots, N,$$

  where η_n are samples i.i.d. drawn form a zero mean Gaussian with variance σ_η^2. Samples x_n are equidistant points in the interval $[0, 2]$. The goal of the task is to predict the value y given a measured value x, using (6). The parameter values used to generate the data were equal to,

$$\theta_0 = 0.2, \ \theta_1 = -1, \ \theta_2 = 0.9, \ \theta_3 = 0.7, \ \theta_5 = -0.2.$$

- I) In the first set of experiments, a Gaussian prior for the unknown θ was used with mean θ_0 equal to the previous true set of parameters and $\Sigma_\theta = 0.1I$. Also, the true model structure was used to construct the matrix Φ. The following figures provide the graphical illustration of the obtained simulation results.

Example on Bayesian Regression

- In this example, we focus on inferring the output directly, after integrating out the parameters. The simplified full Gaussian case will be considered. Data are generated based on the following nonlinear model,

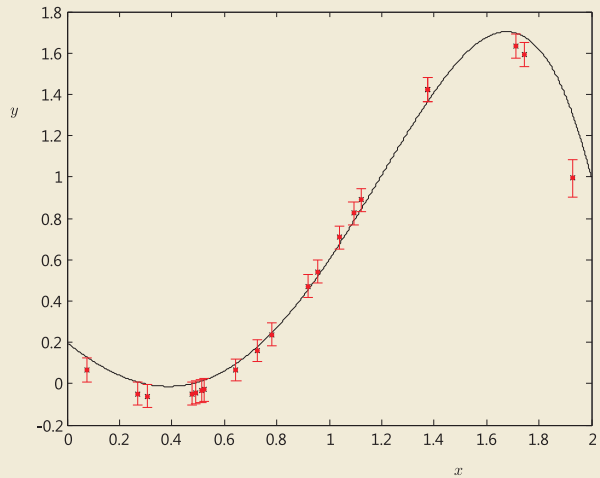
$$y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \theta_3 x_n^3 + \theta_5 x_n^5 + \eta_n, \ n = 1, 2, \ldots, N,$$

  where η_n are samples i.i.d. drawn form a zero mean Gaussian with variance σ_η^2. Samples x_n are equidistant points in the interval $[0, 2]$. The goal of the task is to predict the value y given a measured value x, using (6). The parameter values used to generate the data were equal to,

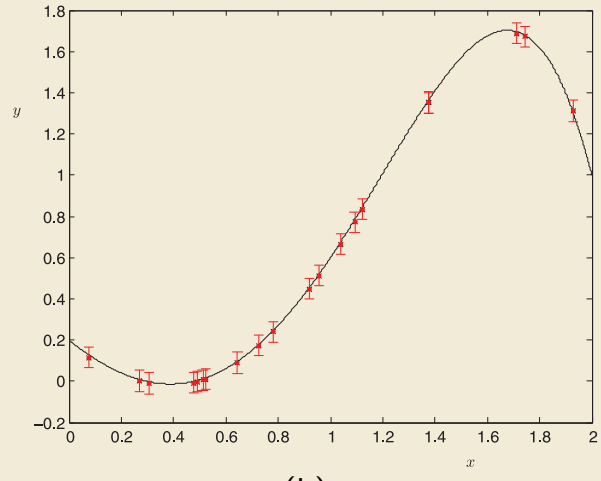
$$\theta_0 = 0.2, \ \theta_1 = -1, \ \theta_2 = 0.9, \ \theta_3 = 0.7, \ \theta_5 = -0.2.$$

- **I**) In the first set of experiments, a Gaussian prior for the unknown θ was used with mean $\boldsymbol{\theta}_0$ equal to the previous true set of parameters and $\Sigma_\theta = 0.1I$. Also, the true model structure was used to construct the matrix Φ. The following figures provide the graphical illustration of the obtained simulation results.

Example on Bayesian Regression



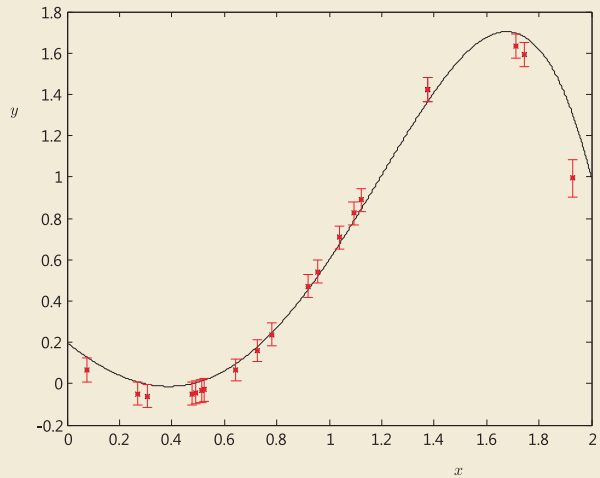
(a)



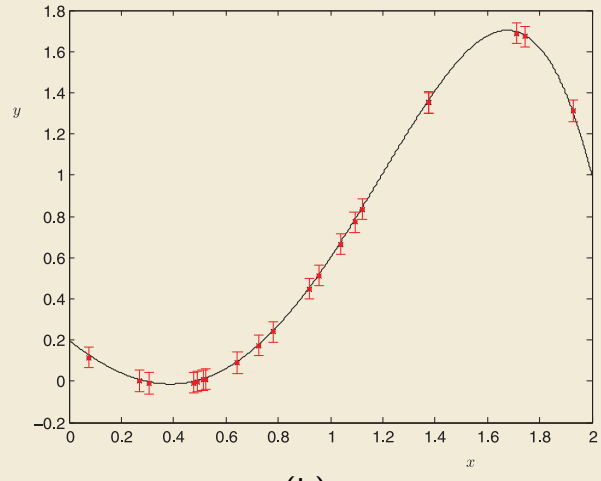
(b)

Each one of the red points, (y, x), indicates the prediction $(\hat{y})$ corresponding to the input value, x. The error bars are dictated by the computed variance, σ_y^2. The mean values used in the Gaussian prior are equal to the true values of the unknown model. (a) $\sigma_\eta^2 = 0.05$, $N = 20$, $\sigma_\theta^2 = 0.1$. (b) $\sigma_\eta^2 = 0.05$, $N = 500$, $\sigma_\theta^2 = 0.1$.

Example on Bayesian Regression

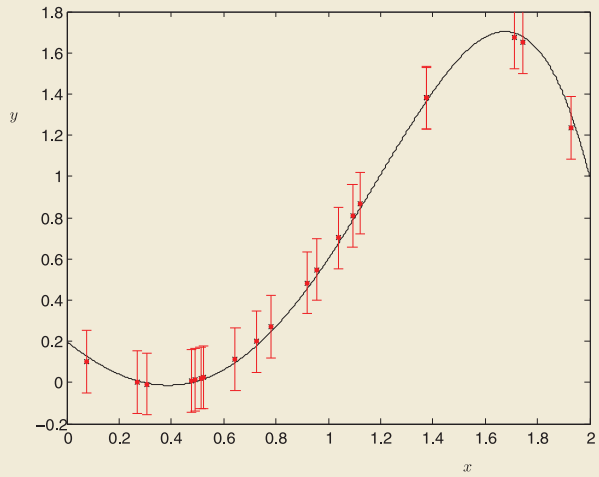


(a)



(b)

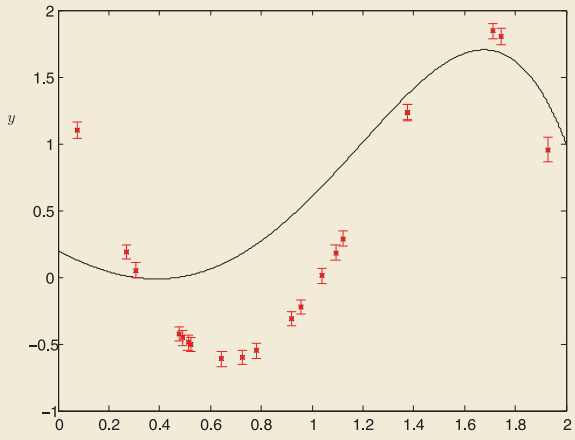
Each one of the red points, (y, x), indicates the prediction $(\hat{y})$ corresponding to the input value, x. The error bars are dictated by the computed variance, σ_y^2. The mean values used in the Gaussian prior are equal to the true values of the unknown model. (a) $\sigma_\eta^2 = 0.05$, $N = 20$, $\sigma_\theta^2 = 0.1$. (b) $\sigma_\eta^2 = 0.05$, $N = 500$, $\sigma_\theta^2 = 0.1$.



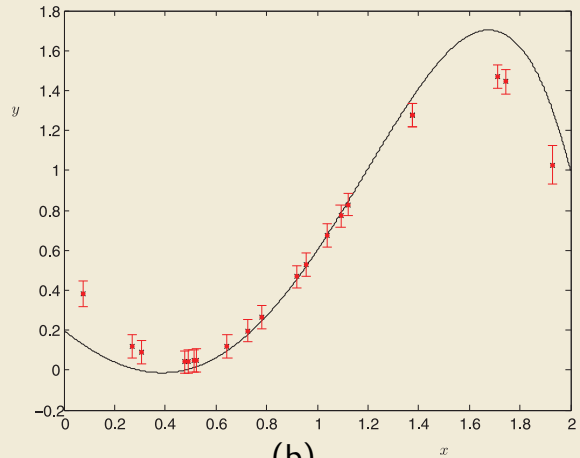
$\sigma_\eta^2 = 0.15$, $N = 500$, $\sigma_\theta^2 = 0.1$. Observe that the larger the data set is the better the predictions are and the larger the noise variance is the larger the error bars become.

Example on Bayesian Regression

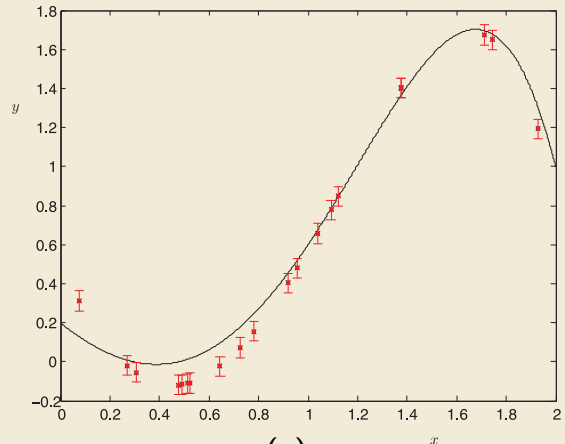
- **II**) In the second set of experiments, we kept the correct model, however, the mean of the prior was given a different value to that of the true model, namely: $\boldsymbol{\theta}_0 = [-10.54, 0.465, 0.0087, -0.093, -0.004]^T$.



(a)



(b)



(c)

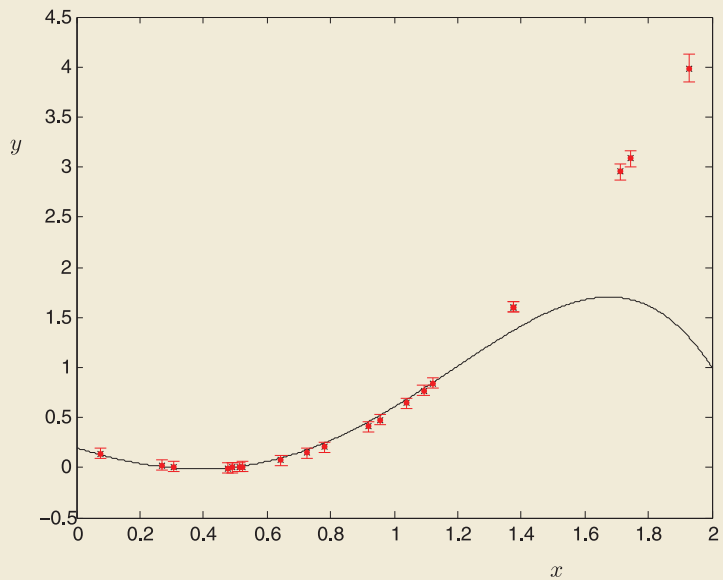
In this set of figures, the mean values of the prior are different than that of the true model. (a) $\sigma_\eta^2 = 0.05$, $N = 20$, $\sigma_\theta^2 = 0.1$. (b) $\sigma_\eta^2 = 0.05$, $N = 20$, $\sigma_\theta^2 = 2$; observe the effect of using larger variance for the prior. (c) $\sigma_\eta^2 = 0.05$, $N = 500$, $\sigma_\theta^2 = 0.1$; observe the effect of the larger training data set.

Example on Bayesian Regression

- **III**) The third set of experiments corresponds to the case where the adopted model for prediction is a wrong one, i.e.,

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \eta.$$

The adopted values were $\sigma_\eta^2 = 0.05$, $N = 500$ and $\sigma_\theta^2 = 2$. From the figure below, observe that once a wrong model has been adopted, one must not have "high expectations" for good prediction performance.



The Evidence Function and Occam's Razor Rule

- Bringing the model explicitly into the scene: The discussion will evolve around the marginal, $p(\boldsymbol{y})$; the latter does depend on the particular model used. Even for Gaussian models, it depends on the model parameters defining the respective pdfs. Hence, it is more natural to write the respective defining equation as,

$$p(\boldsymbol{y}|\mathcal{M}_i) = \int p(\boldsymbol{y}|\mathcal{M}_i, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{M}_i) d\boldsymbol{\theta}, \tag{7}$$

  where $\mathcal{M}_i$ denotes the corresponding model. The quantity $p(\boldsymbol{y}|\mathcal{M}_i)$ is known as the evidence function or simply the **evidence**.

- Assuming the choice of a model to be random and $P(\mathcal{M}_i)$ being the corresponding prior pdf, then mobilizing Bayes theorem, we obtain,

$$P(\mathcal{M}_i|\boldsymbol{y}) = \frac{P(\mathcal{M}_i) p(\boldsymbol{y}|\mathcal{M}_i)}{p(\boldsymbol{y})}, \text{ where } p(\boldsymbol{y}) := \sum_i P(\mathcal{M}_i) p(\boldsymbol{y}|\mathcal{M}_i).$$

The Evidence Function and Occam's Razor Rule

- Bringing the model explicitly into the scene: The discussion will evolve around the marginal, $p(\boldsymbol{y})$; the latter does depend on the particular model used. Even for Gaussian models, it depends on the model parameters defining the respective pdfs. Hence, it is more natural to write the respective defining equation as,

$$p(\boldsymbol{y}|\mathcal{M}_i) = \int p(\boldsymbol{y}|\mathcal{M}_i, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{M}_i) d\boldsymbol{\theta}, \tag{7}$$

  where $\mathcal{M}_i$ denotes the corresponding model. The quantity $p(\boldsymbol{y}|\mathcal{M}_i)$ is known as the evidence function or simply the **evidence**.

- Assuming the choice of a model to be random and $P(\mathcal{M}_i)$ being the corresponding prior pdf, then mobilizing Bayes theorem, we obtain,

$$P(\mathcal{M}_i|\boldsymbol{y}) = \frac{P(\mathcal{M}_i) p(\boldsymbol{y}|\mathcal{M}_i)}{p(\boldsymbol{y})}, \text{ where } p(\boldsymbol{y}) := \sum_i P(\mathcal{M}_i) p(\boldsymbol{y}|\mathcal{M}_i).$$

The Evidence Function and Occam's Razor Rule

- The probability $P(\mathcal{M}_i)$ provides a measure of the **subjective** prior over all possible models, which expresses our guess on how plausible a model is with respect to alternative ones, **prior** to the data arrival.
- We can now obtain the most probable model, after observing $\boldsymbol{y}$, by maximizing the numerator (denominator is independent of the model).
- If one assigns to all possible models equal probabilities, then detecting the most probable model, under the given set of observations becomes a task of maximizing $p(\boldsymbol{y}|\mathcal{M}_i)$. This is the reason that we called this pdf the evidence function for the model.

The Evidence Function and Occam's Razor Rule

- The probability $P(\mathcal{M}_i)$ provides a measure of the **subjective** prior over all possible models, which expresses our guess on how plausible a model is with respect to alternative ones, **prior** to the data arrival.
- We can now obtain the most probable model, after observing $\boldsymbol{y}$, by maximizing the numerator (denominator is independent of the model).
- If one assigns to all possible models equal probabilities, then detecting the most probable model, under the given set of observations becomes a task of maximizing $p(\boldsymbol{y}|\mathcal{M}_i)$. This is the reason that we called this pdf the evidence function for the model.

The Evidence Function and Occam's Razor Rule

- The probability $P(\mathcal{M}_i)$ provides a measure of the **subjective** prior over all possible models, which expresses our guess on how plausible a model is with respect to alternative ones, **prior** to the data arrival.
- We can now obtain the most probable model, after observing $\boldsymbol{y}$, by maximizing the numerator (denominator is independent of the model).
- If one assigns to all possible models equal probabilities, then detecting the most probable model, under the given set of observations becomes a task of maximizing $p(\boldsymbol{y}|\mathcal{M}_i)$. This is the reason that we called this pdf the evidence function for the model.

The Evidence Function and Occam's Razor Rule

- In practice, we content ourselves with using the most probable model, although the most orthodox Bayesian would suggest to average all obtained quantities over all possible models.
- In an ideal Bayesian setting, one does not choose among models; predictions are performed by summing over all possible models, each one weighted by the respective probability.
- However, in many practical problems, we may have reasons to suggest that the evidence function is strongly peaked around a specific model; after all, such an assumption may simplify the task considerably.

The Evidence Function and Occam's Razor Rule

- In practice, we content ourselves with using the most probable model, although the most orthodox Bayesian would suggest to average all obtained quantities over all possible models.
- In an ideal Bayesian setting, one does not choose among models; predictions are performed by summing over all possible models, each one weighted by the respective probability.
- However, in many practical problems, we may have reasons to suggest that the evidence function is strongly peaked around a specific model; after all, such an assumption may simplify the task considerably.

The Evidence Function and Occam's Razor Rule

- In practice, we content ourselves with using the most probable model, although the most orthodox Bayesian would suggest to average all obtained quantities over all possible models.
- In an ideal Bayesian setting, one does not choose among models; predictions are performed by summing over all possible models, each one weighted by the respective probability.
- However, in many practical problems, we may have reasons to suggest that the evidence function is strongly peaked around a specific model; after all, such an assumption may simplify the task considerably.

The Evidence Function and Occam's Razor Rule

- The evidence function: One may wonder whether maximizing $p(\boldsymbol{y}|\mathcal{M}_i)$, w.r. to different models, is any different from maximizing the likelihood, $p(\boldsymbol{y};\boldsymbol{\theta})$ (ML method). As a matter of fact, the two cases belong to two different worlds.
- ML maximizes w.r. to a single (vector) parameter **within an adopted model**, and this is the weak point that makes ML vulnerable to overfitting.
- On the other hand, maximizing the evidence is an optimization task w.r. to the **model itself**; this is a wise alternative that guards us against overfitting, as it will be unravelled next.

The Evidence Function and Occam's Razor Rule

- The evidence function: One may wonder whether maximizing $p(\boldsymbol{y}|\mathcal{M}_i)$, w.r. to different models, is any different from maximizing the likelihood, $p(\boldsymbol{y};\boldsymbol{\theta})$ (ML method). As a matter of fact, the two cases belong to two different worlds.
- ML maximizes w.r. to a single (vector) parameter **within an adopted model**, and this is the weak point that makes ML vulnerable to overfitting.
- On the other hand, maximizing the evidence is an optimization task w.r. to the **model itself**; this is a wise alternative that guards us against overfitting, as it will be unravelled next.

The Evidence Function and Occam's Razor Rule

- The evidence function: One may wonder whether maximizing $p(\boldsymbol{y}|\mathcal{M}_i)$, w.r. to different models, is any different from maximizing the likelihood, $p(\boldsymbol{y};\boldsymbol{\theta})$ (ML method). As a matter of fact, the two cases belong to two different worlds.
- ML maximizes w.r. to a single (vector) parameter **within an adopted model**, and this is the weak point that makes ML vulnerable to overfitting.
- On the other hand, maximizing the evidence is an optimization task w.r. to the **model itself**; this is a wise alternative that guards us against overfitting, as it will be unravelled next.

The Evidence Function and Occam's Razor Rule

- The evidence function: One may wonder whether maximizing $p(\boldsymbol{y}|\mathcal{M}_i)$, w.r. to different models, is any different from maximizing the likelihood, $p(\boldsymbol{y};\boldsymbol{\theta})$ (ML method). As a matter of fact, the two cases belong to two different worlds.
- ML maximizes w.r. to a single (vector) parameter **within an adopted model**, and this is the weak point that makes ML vulnerable to overfitting.
- On the other hand, maximizing the evidence is an optimization task w.r. to the **model itself**; this is a wise alternative that guards us against overfitting, as it will be unravelled next.

The Evidence Function and Occam's Razor Rule

- Let us assume, for simplicity, that $\boldsymbol{\theta}$ is a scalar; i.e., $\theta \in \mathbb{R}$. Furthermore, assume that the posterior, $p(\theta|\boldsymbol{y}, \mathcal{M}_i)$, peaks around a value of θ, within a width of values $\Delta\theta_{\theta|y}$; yet, the posterior is analogous to the integrand in

$$p(\boldsymbol{y}|\mathcal{M}_i) = \int \underbrace{p(\boldsymbol{y}|\mathcal{M}_i, \theta)p(\theta|\mathcal{M}_i)}\, d\theta.$$

  The peak is obviously the value that would result as the MAP estimate, $\hat{\theta}_{MAP}$.

- Then, the evidence can be approximated as,

$$p(\boldsymbol{y}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i, \hat{\theta}_{MAP})p(\hat{\theta}_{MAP}|\mathcal{M}_i)\Delta\theta_{\theta|y}.$$

- To simplify further, assume that the prior pdf is (almost) uniform (of width $\Delta\theta$). Then, we can write,

$$p(\boldsymbol{y}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i, \hat{\theta}_{MAP})\frac{\Delta\theta_{\theta|y}}{\Delta\theta}.$$

The Evidence Function and Occam's Razor Rule

- Let us assume, for simplicity, that $\boldsymbol{\theta}$ is a scalar; i.e., $\theta \in \mathbb{R}$. Furthermore, assume that the posterior, $p(\theta|\boldsymbol{y}, \mathcal{M}_i)$, peaks around a value of θ, within a width of values $\Delta\theta_{\theta|y}$; yet, the posterior is analogous to the integrand in

$$p(\boldsymbol{y}|\mathcal{M}_i) = \int \underbrace{p(\boldsymbol{y}|\mathcal{M}_i, \theta)p(\theta|\mathcal{M}_i)}\, d\theta.$$

  The peak is obviously the value that would result as the MAP estimate, $\hat{\theta}_{MAP}$.

- Then, the evidence can be approximated as,

$$p(\boldsymbol{y}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i, \hat{\theta}_{MAP})p(\hat{\theta}_{MAP}|\mathcal{M}_i)\Delta\theta_{\theta|y}.$$

- To simplify further, assume that the prior pdf is (almost) uniform (of width $\Delta\theta$). Then, we can write,

$$p(\boldsymbol{y}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i, \hat{\theta}_{MAP})\frac{\Delta\theta_{\theta|y}}{\Delta\theta}.$$

The Evidence Function and Occam's Razor Rule

- Let us assume, for simplicity, that $\boldsymbol{\theta}$ is a scalar; i.e., $\theta \in \mathbb{R}$. Furthermore, assume that the posterior, $p(\theta|\boldsymbol{y}, \mathcal{M}_i)$, peaks around a value of θ, within a width of values $\Delta\theta_{\theta|y}$; yet, the posterior is analogous to the integrand in

$$p(\boldsymbol{y}|\mathcal{M}_i) = \int \underbrace{p(\boldsymbol{y}|\mathcal{M}_i, \theta)p(\theta|\mathcal{M}_i)}\, d\theta.$$

  The peak is obviously the value that would result as the MAP estimate, $\hat{\theta}_{MAP}$.

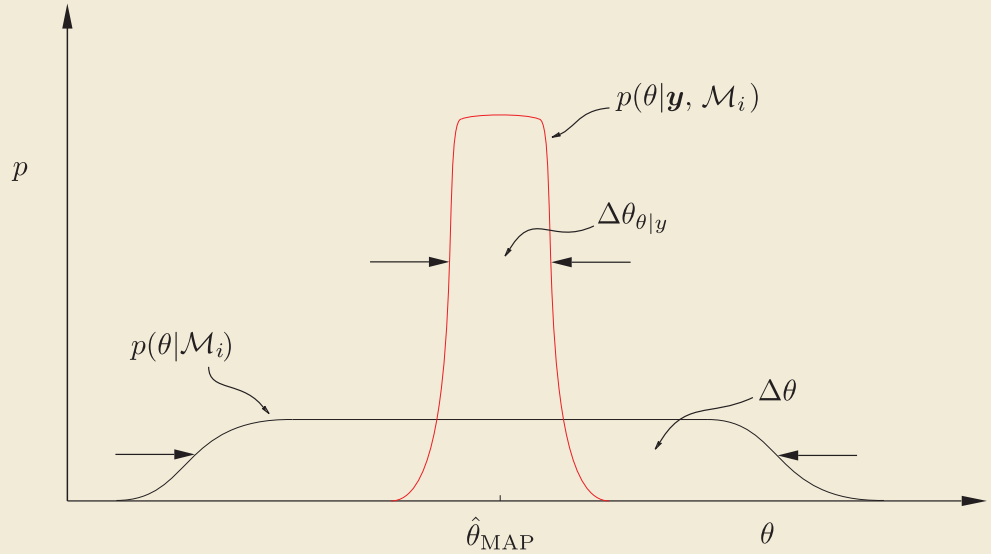
- Then, the evidence can be approximated as,

$$p(\boldsymbol{y}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i, \hat{\theta}_{MAP})p(\hat{\theta}_{MAP}|\mathcal{M}_i)\Delta\theta_{\theta|y}.$$

- To simplify further, assume that the prior pdf is (almost) uniform (of width $\Delta\theta$). Then, we can write,

$$p(\boldsymbol{y}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i, \hat{\theta}_{MAP})\frac{\Delta\theta_{\theta|y}}{\Delta\theta}.$$

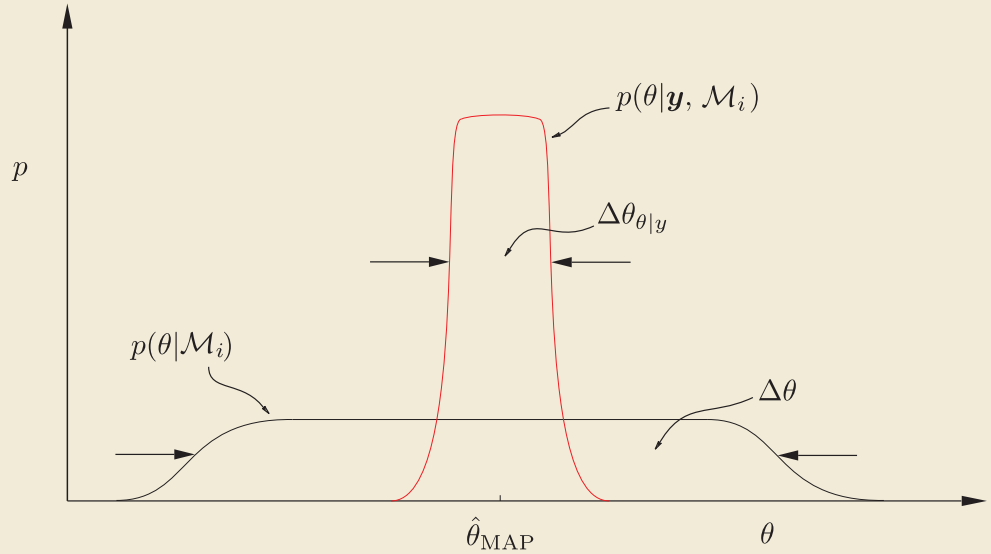
The Evidence Function and Occam's Razor Rule



The posterior peaks around the value $\hat{\theta}_{\text{MAP}}$ and the posterior pdf can be approximated by $p(\hat{\theta}_{\text{MAP}}|\boldsymbol{y}; \mathcal{M}_i)$ over an interval of values equal to $\Delta\theta_{\theta|y}$.

- There two factors involved in the last formula:
  - The factor $p(\boldsymbol{y}|\mathcal{M}_i, \hat{\theta}_{\text{MAP}})$ coincides with the likelihood function at its optimal value, since for this case of uniform prior, $\hat{\theta}_{\text{MAP}} = \hat{\theta}_{\text{ML}}$. In other words, this factor provides us with the best fit that model $\mathcal{M}_i$ can achieve on the given set of observations.
  - However, in contrast to the ML method, the evidence function depends also on the second factor, $\frac{\Delta\theta_{\theta|y}}{\Delta\theta}$. This term accounts for the **complexity** of the model and it is known as the Occam factor.

The Evidence Function and Occam's Razor Rule



The posterior peaks around the value $\hat{\theta}_{\text{MAP}}$ and the posterior pdf can be approximated by $p(\hat{\theta}_{\text{MAP}}|\boldsymbol{y}; \mathcal{M}_i)$ over an interval of values equal to $\Delta\theta_{\theta|y}$.

- There two factors involved in the last formula:
  - The factor $p(\boldsymbol{y}|\mathcal{M}_i, \hat{\theta}_{\text{MAP}})$ coincides with the likelihood function at its optimal value, since for this case of uniform prior, $\hat{\theta}_{\text{MAP}} = \hat{\theta}_{\text{ML}}$. In other words, this factor provides us with the best fit that model $\mathcal{M}_i$ can achieve on the given set of observations.
  - However, in contrast to the ML method, the evidence function depends also on the second factor, $\frac{\Delta\theta_{\theta|y}}{\Delta\theta}$. This term accounts for the **complexity** of the model and it is known as the Occam factor.

The Evidence Function and Occam's Razor Rule

- The Occam factor penalizes these models which are **finely tuned** to the received observations.
- As an example, if two different models $\mathcal{M}_i$ and $\mathcal{M}_j$ have a similar range of values for their prior pdfs, then if, say, $\Delta\theta_{\theta|y}(\mathcal{M}_i) \ll \Delta\theta_{\theta|y}(\mathcal{M}_j)$ then $\mathcal{M}_i$ will be penalized more; only a small range of values for θ survive (i.e., correspond to high probability values) after the reception of y.

The Evidence Function and Occam's Razor Rule

- The Occam factor penalizes these models which are **finely tuned** to the received observations.
- As an example, if two different models $\mathcal{M}_i$ and $\mathcal{M}_j$ have a similar range of values for their prior pdfs, then if, say, $\Delta\theta_{\theta|y}(\mathcal{M}_i) \ll \Delta\theta_{\theta|y}(\mathcal{M}_j)$ then $\mathcal{M}_i$ will be penalized more; only a small range of values for θ survive (i.e., correspond to high probability values) after the reception of $\boldsymbol{y}$.

The Evidence Function and Occam's Razor Rule

- So, if this fine-tuned (to the data) model, $\mathcal{M}_i$, had resulted in a large value of the ML term, it is not certain that the evidence would be maximized for it, since the Occam factor would be small.
- Which model, between the two, finally wins depends on the product of the two involved terms.
- Soon, we will see that the Occam term is also related to the number of parameters; that is, to the complexity of the adopted model.

The Evidence Function and Occam's Razor Rule

- So, if this fine-tuned (to the data) model, $\mathcal{M}_i$, had resulted in a large value of the ML term, it is not certain that the evidence would be maximized for it, since the Occam factor would be small.
- Which model, between the two, finally wins depends on the product of the two involved terms.
- Soon, we will see that the Occam term is also related to the number of parameters; that is, to the complexity of the adopted model.

The Evidence Function and Occam's Razor Rule

- So, if this fine-tuned (to the data) model, $\mathcal{M}_i$, had resulted in a large value of the ML term, it is not certain that the evidence would be maximized for it, since the Occam factor would be small.
- Which model, between the two, finally wins depends on the product of the two involved terms.
- Soon, we will see that the Occam term is also related to the number of parameters; that is, to the complexity of the adopted model.

Laplacian Approximation and the Evidence Function

- Laplacian approximation: To investigate the evidence function for the general multiparameter case, we will employ the method of Laplacian approximation of a pdf. This is a general methodology that approximates **any** pdf **locally** in terms of a **Gaussian** one. To this end, define

$$g(\boldsymbol{\theta}) := \ln\left(p(\boldsymbol{y}|\mathcal{M}_i, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{M}_i)\right).$$

- Use Taylor's expansion around $\hat{\theta}_{\text{MAP}}$ and keep terms up to the second order,

$$\begin{aligned} g(\boldsymbol{\theta}) &= g(\hat{\boldsymbol{\theta}}_{\text{MAP}}) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}})^T \frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}} \\ &+ \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}})^T \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}}) \\ &= g(\hat{\boldsymbol{\theta}}_{\text{MAP}}) - \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}})^T \Sigma^{-1} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}}), \end{aligned}$$

where

$$\Sigma^{-1} := -\frac{\partial^2 g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}}.$$

Laplacian Approximation and the Evidence Function

- Laplacian approximation: To investigate the evidence function for the general multiparameter case, we will employ the method of Laplacian approximation of a pdf. This is a general methodology that approximates **any** pdf **locally** in terms of a **Gaussian** one. To this end, define

$$g(\boldsymbol{\theta}) := \ln\left(p(\boldsymbol{y}|\mathcal{M}_i, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{M}_i)\right).$$

- Use Taylor's expansion around $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ and keep terms up to the second order,

$$\begin{aligned} g(\boldsymbol{\theta}) &= g(\hat{\boldsymbol{\theta}}_{\text{MAP}}) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}})^T \frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}} \\ &+ \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}})^T \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}}) \\ &= g(\hat{\boldsymbol{\theta}}_{\text{MAP}}) - \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}})^T \Sigma^{-1} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\text{MAP}}), \end{aligned}$$

where

$$\Sigma^{-1} := -\frac{\partial^2 g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}}.$$

Laplacian Approximation and the Evidence Function

- The last equation readily leads, by a simple inspection, to the following approximation,

$$p(\boldsymbol{y}|\mathcal{M}_i,\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i,\hat{\boldsymbol{\theta}}_{\text{MAP}})p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\mathcal{M}_i) \times \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}_{\text{MAP}})^T\Sigma^{-1}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}_{\text{MAP}})\right).$$

- Plugging the last equation into the defining integral of (7) we obtain

$$p(\boldsymbol{y}|\mathcal{M}_i) = p(\boldsymbol{y}|\mathcal{M}_i,\hat{\theta}_{\text{MAP}})p(\hat{\theta}_{\text{MAP}}|\mathcal{M}_i)(2\pi)^{\frac{K}{2}}|\Sigma|^{1/2},$$

and taking the logarithms, we have

$$\underbrace{\ln p(\boldsymbol{y}|\mathcal{M}_i)}_{\text{Evidence}} = \underbrace{\ln p(\boldsymbol{y}|\mathcal{M}_i,\hat{\boldsymbol{\theta}}_{\text{MAP}})}_{\text{Best likelihood fit}} + \underbrace{\ln p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\mathcal{M}_i) + \frac{K}{2}\ln(2\pi) + \frac{1}{2}\ln|\Sigma|}_{\text{Occam factor}}.$$

Laplacian Approximation and the Evidence Function

- The last equation readily leads, by a simple inspection, to the following approximation,

$$p(\boldsymbol{y}|\mathcal{M}_i,\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i,\hat{\boldsymbol{\theta}}_{\text{MAP}})p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\mathcal{M}_i) \times \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}_{\text{MAP}})^T\Sigma^{-1}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}_{\text{MAP}})\right).$$

- Plugging the last equation into the defining integral of (7) we obtain

$$p(\boldsymbol{y}|\mathcal{M}_i) = p(\boldsymbol{y}|\mathcal{M}_i,\hat{\boldsymbol{\theta}}_{\text{MAP}})p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\mathcal{M}_i)(2\pi)^{\frac{K}{2}}|\Sigma|^{1/2},$$

and taking the logarithms, we have

$$\underbrace{\ln p(\boldsymbol{y}|\mathcal{M}_i)}_{\text{Evidence}} = \underbrace{\ln p(\boldsymbol{y}|\mathcal{M}_i,\hat{\boldsymbol{\theta}}_{\text{MAP}})}_{\text{Best likelihood fit}} + \underbrace{\ln p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\mathcal{M}_i) + \frac{K}{2}\ln(2\pi) + \frac{1}{2}\ln|\Sigma|}_{\text{Occam factor}}.$$

Laplacian Approximation and the Evidence Function

- The last equation readily leads, by a simple inspection, to the following approximation,

$$p(\boldsymbol{y}|\mathcal{M}_i,\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{M}_i) \simeq p(\boldsymbol{y}|\mathcal{M}_i,\hat{\boldsymbol{\theta}}_{\text{MAP}})p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\mathcal{M}_i) \times \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}_{\text{MAP}})^T\Sigma^{-1}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}_{\text{MAP}})\right).$$

- Plugging the last equation into the defining integral of (7) we obtain

$$p(\boldsymbol{y}|\mathcal{M}_i) = p(\boldsymbol{y}|\mathcal{M}_i,\hat{\boldsymbol{\theta}}_{\text{MAP}})p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\mathcal{M}_i)(2\pi)^{\frac{K}{2}}|\Sigma|^{1/2},$$

and taking the logarithms, we have

$$\underbrace{\ln p(\boldsymbol{y}|\mathcal{M}_i)}_{\text{Evidence}} = \underbrace{\ln p(\boldsymbol{y}|\mathcal{M}_i,\hat{\boldsymbol{\theta}}_{\text{MAP}})}_{\text{Best likelihood fit}} + \underbrace{\ln p(\hat{\boldsymbol{\theta}}_{\text{MAP}}|\mathcal{M}_i) + \frac{K}{2}\ln(2\pi) + \frac{1}{2}\ln|\Sigma|}_{\text{Occam factor}}.$$

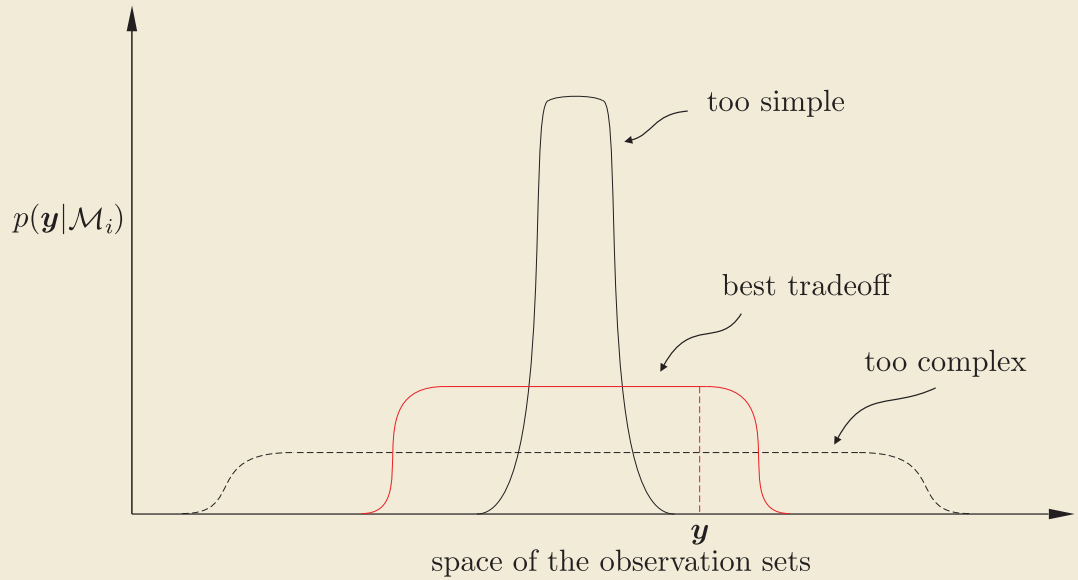
Laplacian Approximation and the Evidence Function

- The dependence on the complexity (number of basis functions) of the model is readily spotted. Moreover, the Occam term, quantifying complexity, depends on the prior **and** the second derivatives (via Σ) of the posterior; that is, it depends on how "sharp" its shape is.
- Hence, in a single equation, besides the number of parameters and the associated best-fit term, the evidence takes into account also information related to the associated variance; maximizing the evidence leads to the best tradeoff.

Laplacian Approximation and the Evidence Function

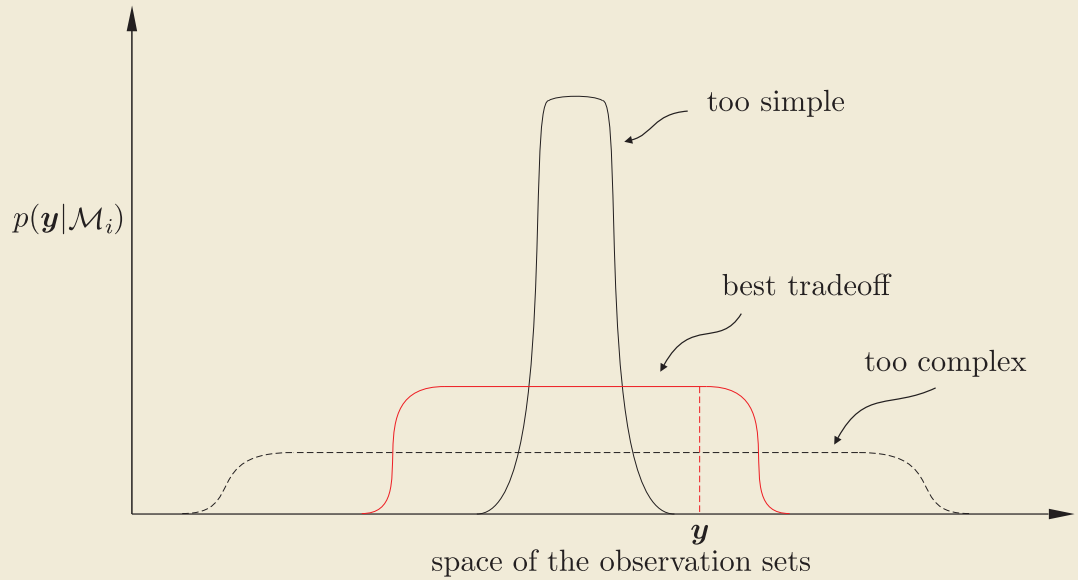
- The dependence on the complexity (number of basis functions) of the model is readily spotted. Moreover, the Occam term, quantifying complexity, depends on the prior **and** the second derivatives (via Σ) of the posterior; that is, it depends on how "sharp" its shape is.
- Hence, in a single equation, besides the number of parameters and the associated best-fit term, the evidence takes into account also information related to the associated variance; maximizing the evidence leads to the best tradeoff.

Laplacian Approximation and the Evidence Function



- If the model is too complex, it can fit well a wide range of data sets, and since $p(\boldsymbol{y}|\mathcal{M}_i)$ has to integrate to one, its value for any value of $\boldsymbol{y}$ is expected to be low. The opposite is true for models that are too simple; such models can model well some data sets and consequently the evidence function peaks sharply around a value in the space of observation sets. Thus, selecting a data set at random, it is rather unlikely that this has been generated by such a model.

Laplacian Approximation and the Evidence Function



- Note that, the Occam term does not depend solely on the number of parameters; hence, complexity here should be interpreted in a more "open-minded" way. This robustness against overfitting, which is intrinsic in the Bayesian inference approach, is the consequence of integrating out the parameters for any specific model in (7); this integration penalizes models of **high complexity**, because such models can model a **large range** of data.

Bayesian Learning: Some Remarks

- In the Bayesian approach, one makes all the modeling assumptions **explicit** and it is then left to the rules of probability theory to provide the answers. One has not to worry about the choice of an optimizing criterion, where different criteria lead to different estimators and there is not an objective systematic way to decide which criterion is best.
- On the other hand, in the Bayesian approach, one has to make sure that selects the **prior** that explains the data in the best possible way.

Bayesian Learning: Some Remarks

- In the Bayesian approach, one makes all the modeling assumptions **explicit** and it is then left to the rules of probability theory to provide the answers. One has not to worry about the choice of an optimizing criterion, where different criteria lead to different estimators and there is not an objective systematic way to decide which criterion is best.
- On the other hand, in the Bayesian approach, one has to make sure that selects the **prior** that explains the data in the best possible way.

Bayesian Learning: Some Remarks

- The choice of the prior pdf is very critical in the performance of Bayesian methods and must be carried out in such a way so that to encapsulate prior knowledge as fully as possible.

Bayesian Learning: Some Remarks

- Note, however, that the Bayesian approach is not free from the cross-validation phase. Maximizing the evidence, which at the same time guards against overfitting, does not necessarily mean that the performance of the designed estimator is optimized.
- There is no reason to suggest that the evidence may be a reliable predictor of the generalization performance. The generalization performance depends very much on whether the adopted prior matches the "true" distribution of the unknown parameters. Thus, the performance has to be tested on data.

Bayesian Learning: Some Remarks

- Note, however, that the Bayesian approach is not free from the cross-validation phase. Maximizing the evidence, which at the same time guards against overfitting, does not necessarily mean that the performance of the designed estimator is optimized.
- There is no reason to suggest that the evidence may be a reliable predictor of the generalization performance. The generalization performance depends very much on whether the adopted prior matches the "true" distribution of the unknown parameters. Thus, the performance has to be tested on data.

Bayesian Learning: Some Remarks

- The Laplacian approximation to the evidence function is closely related to the Bayesian Information Criterion (BIC) for model selection, which is expressed as,

$$\ln p(\boldsymbol{y}|\mathcal{M}_i) \approx \ln p(\boldsymbol{y}|\mathcal{M}_i, \hat{\boldsymbol{\theta}}_{\text{MAP}}) - \frac{1}{2} K \ln N.$$

BIC is obtained as a limiting form for large N of the Laplacian approximation of the evidence function, as discussed before, assuming a broad enough Gaussian prior, and manipulating a bit on the determinant involved in the last term.

Bayesian Learning: Some Remarks

- The Laplacian approximation to the evidence function is closely related to the Bayesian Information Criterion (BIC) for model selection, which is expressed as,

$$\ln p(\boldsymbol{y}|\mathcal{M}_i) \approx \ln p(\boldsymbol{y}|\mathcal{M}_i, \hat{\boldsymbol{\theta}}_{\text{MAP}}) - \frac{1}{2} K \ln N.$$

  BIC is obtained as a limiting form for large N of the Laplacian approximation of the evidence function, as discussed before, assuming a broad enough Gaussian prior, and manipulating a bit on the determinant involved in the last term.

Bayesian Learning: Some Remarks

- The Bayesian framework is also closely related to the Minimum Description Length (MDL) methods. The log-evidence is associated to the number of bits in the shortest message that encodes the data via model $\mathcal{M}_i$.

Bayesian Learning: Some Remarks

- Type II Maximum Likelihood: Note that the evidence is the marginal likelihood function after **integrating out** the parameters θ.
- To distinguish it from the MAP method, when the evidence function is maximized, with respect to a set of some unknown parameters, it is usually referred to as Generalized Maximum Likelihood or Type II Maximum Likelihood and sometimes as Empirical Bayes. In contrast, the MAP estimator is sometimes called Type I estimator.

Bayesian Learning: Some Remarks

- Type II Maximum Likelihood: Note that the evidence is the marginal likelihood function after **integrating out** the parameters θ.
- To distinguish it from the MAP method, when the evidence function is maximized, with respect to a set of some unknown parameters, it is usually referred to as Generalized Maximum Likelihood or Type II Maximum Likelihood and sometimes as Empirical Bayes. In contrast, the MAP estimator is sometimes called Type I estimator.

Latent Variables And The EM Algorithm

- Adopting the Gaussian for the prior as well as the conditional pdfs, renders the analytical computation of the evidence function possible, i.e.,

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}|\Phi\boldsymbol{\theta}_0, \Sigma_\eta + \Phi\Sigma_\theta\Phi^T\right).$$

- Assume that $\Sigma_\eta = \sigma_\eta^2 I$, $\Sigma_\theta = \sigma_\theta^2 I$ and $\boldsymbol{\theta_0} = \mathbf{0}$. Then, the evidence function depends on two user-defined parameters, i.e., $\boldsymbol{\xi} := [\sigma_\eta^2, \sigma_\theta^2]^T$. Let us make this dependence explicit into the notation and write $p(\boldsymbol{y}; \boldsymbol{\xi})$.

- We can now compute the parameter vector $\boldsymbol{\xi}$ by maximizing the evidence function. For such cases, this is just an instance of the maximum likelihood method.

Latent Variables And The EM Algorithm

- Adopting the Gaussian for the prior as well as the conditional pdfs, renders the analytical computation of the evidence function possible, i.e.,

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}|\Phi\boldsymbol{\theta}_0, \Sigma_\eta + \Phi\Sigma_\theta\Phi^T\right).$$

- Assume that $\Sigma_\eta = \sigma_\eta^2 I$, $\Sigma_\theta = \sigma_\theta^2 I$ and $\boldsymbol{\theta_0} = \mathbf{0}$. Then, the evidence function depends on two user-defined parameters, i.e., $\boldsymbol{\xi} := [\sigma_\eta^2, \sigma_\theta^2]^T$. Let us make this dependence explicit into the notation and write $p(\boldsymbol{y}; \boldsymbol{\xi})$.

- We can now compute the parameter vector ξ by maximizing the evidence function. For such cases, this is just an instance of the maximum likelihood method.

Latent Variables And The EM Algorithm

- Adopting the Gaussian for the prior as well as the conditional pdfs, renders the analytical computation of the evidence function possible, i.e.,

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{y}|\Phi\boldsymbol{\theta}_0, \Sigma_\eta + \Phi\Sigma_\theta\Phi^T\right).$$

- Assume that $\Sigma_\eta = \sigma_\eta^2 I$, $\Sigma_\theta = \sigma_\theta^2 I$ and $\boldsymbol{\theta_0} = \mathbf{0}$. Then, the evidence function depends on two user-defined parameters, i.e., $\boldsymbol{\xi} := [\sigma_\eta^2, \sigma_\theta^2]^T$. Let us make this dependence explicit into the notation and write $p(\boldsymbol{y}; \boldsymbol{\xi})$.

- We can now compute the parameter vector $\boldsymbol{\xi}$ by maximizing the evidence function. For such cases, this is just an instance of the maximum likelihood method.

Latent Variables And The EM Algorithm

- In general, such closed-form expressions for the evidence function are **not** possible, and the integration in the respective equation is intractable.
- The source of difficulty is that our model is described by two random variables, i.e., $\mathbf{y}$ and $\boldsymbol{\theta}$, yet **only one** of them, $\mathbf{y}$, can be directly observed. The other one, $\boldsymbol{\theta}$, cannot be observed and this is the reason that the Bayesian philosophy tries to **integrate** it out of the joint pdf, $p(\boldsymbol{y}, \boldsymbol{\theta})$.
- If $\boldsymbol{\theta}$ could be observed, the set of parameters, $\boldsymbol{\xi}$, could be obtained by maximizing the likelihood $p(\boldsymbol{y}, \boldsymbol{\theta}; \boldsymbol{\xi})$, given a set of (joint) observations $(\boldsymbol{y}, \boldsymbol{\theta})$. Because it cannot be observed, the random variables in $\boldsymbol{\theta}$ are known as latent or hidden variables.

Latent Variables And The EM Algorithm

- In general, such closed-form expressions for the evidence function are **not** possible, and the integration in the respective equation is intractable.
- The source of difficulty is that our model is described by two random variables, i.e., $\mathbf{y}$ and $\boldsymbol{\theta}$, yet **only one** of them, $\mathbf{y}$, can be directly observed. The other one, $\boldsymbol{\theta}$, cannot be observed and this is the reason that the Bayesian philosophy tries to **integrate** it out of the joint pdf, $p(\boldsymbol{y}, \boldsymbol{\theta})$.
- If $\boldsymbol{\theta}$ could be observed, the set of parameters, $\boldsymbol{\xi}$, could be obtained by maximizing the likelihood $p(\boldsymbol{y}, \boldsymbol{\theta}; \boldsymbol{\xi})$, given a set of (joint) observations $(\boldsymbol{y}, \boldsymbol{\theta})$. Because it cannot be observed, the random variables in $\boldsymbol{\theta}$ are known as latent or hidden variables.

Latent Variables And The EM Algorithm

- In general, such closed-form expressions for the evidence function are **not** possible, and the integration in the respective equation is intractable.
- The source of difficulty is that our model is described by two random variables, i.e., $\mathbf{y}$ and $\boldsymbol{\theta}$, yet **only one** of them, $\mathbf{y}$, can be directly observed. The other one, $\boldsymbol{\theta}$, cannot be observed and this is the reason that the Bayesian philosophy tries to **integrate** it out of the joint pdf, $p(\boldsymbol{y}, \boldsymbol{\theta})$.
- If $\boldsymbol{\theta}$ could be observed, the set of parameters, $\boldsymbol{\xi}$, could be obtained by maximizing the likelihood $p(\boldsymbol{y}, \boldsymbol{\theta}; \boldsymbol{\xi})$, given a set of (joint) observations $(\boldsymbol{y}, \boldsymbol{\theta})$. Because it cannot be observed, the random variables in $\boldsymbol{\theta}$ are known as latent or hidden variables.

The Expectation Maximization Algorithm

- Latent variables occur very often in a number of problems in probability and statistics. In a number of cases, from a larger set of jointly distributed random variables only some can be observed and the rest remain hidden. Also, it is often useful to build hidden variables into a model by design. These variables are meant to represent latent **causes** that influence the observed variables and their introduction may facilitate the analysis.
- The EM algorithm: The Expectation-Maximization algorithm (EM) is an elegant tool to maximize the likelihood function for problems with latent variables. The problem is stated next in a general formulation.

The Expectation Maximization Algorithm

- Latent variables occur very often in a number of problems in probability and statistics. In a number of cases, from a larger set of jointly distributed random variables only some can be observed and the rest remain hidden. Also, it is often useful to build hidden variables into a model by design. These variables are meant to represent latent **causes** that influence the observed variables and their introduction may facilitate the analysis.
- The EM algorithm: The Expectation-Maximization algorithm (EM) is an elegant tool to maximize the likelihood function for problems with latent variables. The problem is stated next in a general formulation.

The Expectation Maximization Algorithm

- Let $\mathbf{x}$ be a random vector and let $\mathcal{X}$ be the respective set of observations. Let $\mathcal{X}^l := \{\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_N^l\}$ be the corresponding set of latent variables; these can be either of a discrete or of a continuous nature.
- Each observation in $\mathcal{X}$ is associated with a latent vector, $\mathbf{x}^l$, in $\mathcal{X}^l$. We refer to the set $\{\mathcal{X}, \mathcal{X}^l\}$ as the complete data set and to the set of observations, $\mathcal{X}$, as the incomplete one. Let their joint distribution be parameterized in terms of a set of unknown parameters, $\boldsymbol{\xi}$.
- Note that, everything to be said, also, applies if in addition to or instead of $\mathcal{X}^l$ the set of hidden variables contains parameters of **fixed** size, say θ, independent of the size N.

The Expectation Maximization Algorithm

- Let $\mathbf{x}$ be a random vector and let $\mathcal{X}$ be the respective set of observations. Let $\mathcal{X}^l := \{\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_N^l\}$ be the corresponding set of latent variables; these can be either of a discrete or of a continuous nature.
- Each observation in $\mathcal{X}$ is associated with a latent vector, $\mathbf{x}^l$, in $\mathcal{X}^l$. We refer to the set $\{\mathcal{X}, \mathcal{X}^l\}$ as the complete data set and to the set of observations, $\mathcal{X}$, as the incomplete one. Let their joint distribution be parameterized in terms of a set of unknown parameters, $\boldsymbol{\xi}$.
- Note that, everything to be said, also, applies if in addition to or instead of $\mathcal{X}^l$ the set of hidden variables contains parameters of **fixed** size, say θ, independent of the size N.

The Expectation Maximization Algorithm

- Let $\mathbf{x}$ be a random vector and let $\mathcal{X}$ be the respective set of observations. Let $\mathcal{X}^l := \{\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_N^l\}$ be the corresponding set of latent variables; these can be either of a discrete or of a continuous nature.
- Each observation in $\mathcal{X}$ is associated with a latent vector, $\mathbf{x}^l$, in $\mathcal{X}^l$. We refer to the set $\{\mathcal{X}, \mathcal{X}^l\}$ as the complete data set and to the set of observations, $\mathcal{X}$, as the incomplete one. Let their joint distribution be parameterized in terms of a set of unknown parameters, $\boldsymbol{\xi}$.
- Note that, everything to be said, also, applies if in addition to or instead of $\mathcal{X}^l$ the set of hidden variables contains parameters of **fixed** size, say θ, independent of the size N.

The Expectation Maximization Algorithm

- If the complete log-likelihood $\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})$ were available, then the problem would be a typical ML one.
- However, since no observations for the latent variables are available, the EM algorithm considers the **expectation** of the complete log-likelihood w.r. to $\mathcal{X}^l$; this operation is possible, only if the posterior distribution $p(\mathcal{X}^l|\mathcal{X}; \boldsymbol{\xi})$ is **assumed** to be known, provided that ξ is known, too.
- To this end, the EM algorithm builds upon an iterative philosophy, initialized by an arbitrary value $\boldsymbol{\xi}^{(0)}$. Then it proceeds along the following steps:

The Expectation Maximization Algorithm

- If the complete log-likelihood $\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})$ were available, then the problem would be a typical ML one.
- However, since no observations for the latent variables are available, the EM algorithm considers the **expectation** of the complete log-likelihood w.r. to $\mathcal{X}^l$; this operation is possible, only if the posterior distribution $p(\mathcal{X}^l|\mathcal{X}; \boldsymbol{\xi})$ is **assumed** to be known, provided that $\boldsymbol{\xi}$ is known, too.
- To this end, the EM algorithm builds upon an iterative philosophy, initialized by an arbitrary value $\boldsymbol{\xi}^{(0)}$. Then it proceeds along the following steps:

The Expectation Maximization Algorithm

- If the complete log-likelihood $\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})$ were available, then the problem would be a typical ML one.
- However, since no observations for the latent variables are available, the EM algorithm considers the **expectation** of the complete log-likelihood w.r. to $\mathcal{X}^l$; this operation is possible, only if the posterior distribution $p(\mathcal{X}^l|\mathcal{X}; \boldsymbol{\xi})$ is **assumed** to be known, provided that $\boldsymbol{\xi}$ is known, too.
- To this end, the EM algorithm builds upon an iterative philosophy, initialized by an arbitrary value $\boldsymbol{\xi}^{(0)}$. Then it proceeds along the following steps:

- **The EM Algorithm**

  1. Expectation E-step: at the $(j+1)$ iteration, compute $p(\mathcal{X}^l|\mathcal{X}, \boldsymbol{\xi}^{(j)})$ and

$$\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\Big[\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})\Big], \tag{8}$$

     where the expectation is taken with respect to $p(\mathcal{X}^l|\mathcal{X}; \boldsymbol{\xi}^{(j)})$.

  2. Maximization M-step: Determine $\boldsymbol{\xi}^{(j+1)}$ so that

$$\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}). \tag{9}$$

  3. Check for convergence according to a criterion. If it is not satisfied go to step 1.

- A possible convergence criterion is to check whether $\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}^{(j)}\| < \epsilon$, for some user-defined constant ϵ. The use of the EM algorithm presupposes that working with the joint pdf, $p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})$, is computationally tractable.

- **The EM Algorithm**

  1. Expectation E-step: at the $(j+1)$ iteration, compute $p(\mathcal{X}^l|\mathcal{X}, \boldsymbol{\xi}^{(j)})$ and

$$\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\Big[\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})\Big], \tag{8}$$

     where the expectation is taken with respect to $p(\mathcal{X}^l|\mathcal{X}; \boldsymbol{\xi}^{(j)})$.

  2. Maximization M-step: Determine $\boldsymbol{\xi}^{(j+1)}$ so that

$$\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}). \tag{9}$$

  3. Check for convergence according to a criterion. If it is not satisfied go to step 1.

- A possible convergence criterion is to check whether $\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}^{(j)}\| < \epsilon$, for some user-defined constant ϵ. The use of the EM algorithm presupposes that working with the joint pdf, $p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})$, is computationally tractable.

- **The EM Algorithm**

  1. Expectation E-step: at the $(j+1)$ iteration, compute $p(\mathcal{X}^l|\mathcal{X}, \boldsymbol{\xi}^{(j)})$ and

$$\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\Big[\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})\Big], \tag{8}$$

     where the expectation is taken with respect to $p(\mathcal{X}^l|\mathcal{X}; \boldsymbol{\xi}^{(j)})$.

  2. Maximization M-step: Determine $\boldsymbol{\xi}^{(j+1)}$ so that

$$\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}). \tag{9}$$

  3. Check for convergence according to a criterion. If it is not satisfied go to step 1.

- A possible convergence criterion is to check whether $\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}^{(j)}\| < \epsilon$, for some user-defined constant ϵ. The use of the EM algorithm presupposes that working with the joint pdf, $p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})$, is computationally tractable.

- The Bayesian viewpoint to the regression has already been considered via the Gaussian model assumption for the conditional, $p(\boldsymbol{y}|\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$. This in turn led to a Gaussian posterior, $p(\boldsymbol{\theta}|\boldsymbol{y})$. Assume, for simplicity that, $\Sigma_\eta = \sigma_\eta^2 I$, as well as for the respective prior, $\Sigma_\theta = \sigma_\theta^2 I$. Let also for the prior that $\theta_0 = \mathbf{0}$. Hence, the posterior is the Gaussian $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta|y}, \Sigma_\theta|y)$ where (Eqs. (4) and (5))

$$\boldsymbol{\mu}_{\theta|y} = \frac{1}{\sigma_\eta^2}\left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1}\Phi^T\boldsymbol{y},$$

$$\Sigma_{\theta|y} = \left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1}.$$

- Our goal now becomes to consider σ_η^2 and σ_θ^2 as (non-random) parameters and to obtain their values by maximizing the evidence function in (1). We will employ the EM algorithm.

- The Bayesian viewpoint to the regression has already been considered via the Gaussian model assumption for the conditional, $p(\boldsymbol{y}|\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$. This in turn led to a Gaussian posterior, $p(\boldsymbol{\theta}|\boldsymbol{y})$. Assume, for simplicity that, $\Sigma_\eta = \sigma_\eta^2 I$, as well as for the respective prior, $\Sigma_\theta = \sigma_\theta^2 I$. Let also for the prior that $\boldsymbol{\theta}_0 = \mathbf{0}$. Hence, the posterior is the Gaussian $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta|y}, \Sigma_\theta|y)$ where (Eqs. (4) and (5))

$$\boldsymbol{\mu}_{\theta|y} = \frac{1}{\sigma_\eta^2}\left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1}\Phi^T\boldsymbol{y},$$

$$\Sigma_{\theta|y} = \left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1}.$$

- Our goal now becomes to consider σ_η^2 and σ_θ^2 as (non-random) parameters and to obtain their values by maximizing the evidence function in (1). We will employ the EM algorithm.

- The Bayesian viewpoint to the regression has already been considered via the Gaussian model assumption for the conditional, $p(\boldsymbol{y}|\boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$. This in turn led to a Gaussian posterior, $p(\boldsymbol{\theta}|\boldsymbol{y})$. Assume, for simplicity that, $\Sigma_\eta = \sigma_\eta^2 I$, as well as for the respective prior, $\Sigma_\theta = \sigma_\theta^2 I$. Let also for the prior that $\boldsymbol{\theta}_0 = \mathbf{0}$. Hence, the posterior is the Gaussian $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta|y}, \Sigma_\theta|y)$ where (Eqs. (4) and (5))

$$\boldsymbol{\mu}_{\theta|y} = \frac{1}{\sigma_\eta^2}\left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1}\Phi^T\boldsymbol{y},$$

$$\Sigma_{\theta|y} = \left(\frac{1}{\sigma_\theta^2} I + \frac{1}{\sigma_\eta^2}\Phi^T\Phi\right)^{-1}.$$

- Our goal now becomes to consider σ_η^2 and σ_θ^2 as (non-random) parameters and to obtain their values by maximizing the evidence function in (1). We will employ the EM algorithm.

Linear Regression And The EM Algorithm

- In the current context, the set of observations, which in the general EM formulation were denoted as $\mathcal{X}$, will be our familiar observation vector, $\boldsymbol{y}$. Also, the place of the set of the latent variables, denoted as $\mathcal{X}^l$, is replaced by $\boldsymbol{\theta}$.
- Recall that a prerequisite in order to apply the EM algorithm is the knowledge of the posterior, which for this case is known, given the value of the parameters, σ_η^2 and σ_θ^2.
- We will work with the precision variables and the parameter vector of the unknown variables becomes

$$\boldsymbol{\xi} = [\alpha, \beta]^T, \ \alpha := \frac{1}{\sigma_\theta^2} \text{ and } \beta := \frac{1}{\sigma_\eta^2}.$$

- The EM algorithm is initialized with some arbitrary positive values $\alpha^{(0)}$ and $\beta^{(0)}$. The resulting algorithm proceeds as follows:

Linear Regression And The EM Algorithm

- In the current context, the set of observations, which in the general EM formulation were denoted as $\mathcal{X}$, will be our familiar observation vector, $\boldsymbol{y}$. Also, the place of the set of the latent variables, denoted as $\mathcal{X}^l$, is replaced by $\boldsymbol{\theta}$.
- Recall that a prerequisite in order to apply the EM algorithm is the knowledge of the posterior, which for this case is known, given the value of the parameters, σ_η^2 and σ_θ^2.
- We will work with the precision variables and the parameter vector of the unknown variables becomes

$$\boldsymbol{\xi} = [\alpha, \beta]^T, \ \alpha := \frac{1}{\sigma_\theta^2} \text{ and } \beta := \frac{1}{\sigma_\eta^2}.$$

- The EM algorithm is initialized with some arbitrary positive values $\alpha^{(0)}$ and $\beta^{(0)}$. The resulting algorithm proceeds as follows:

Linear Regression And The EM Algorithm

- In the current context, the set of observations, which in the general EM formulation were denoted as $\mathcal{X}$, will be our familiar observation vector, $\boldsymbol{y}$. Also, the place of the set of the latent variables, denoted as $\mathcal{X}^l$, is replaced by $\boldsymbol{\theta}$.
- Recall that a prerequisite in order to apply the EM algorithm is the knowledge of the posterior, which for this case is known, given the value of the parameters, σ_η^2 and σ_θ^2.
- We will work with the precision variables and the parameter vector of the unknown variables becomes

$$\boldsymbol{\xi} = [\alpha, \beta]^T, \ \alpha := \frac{1}{\sigma_\theta^2} \text{ and } \beta := \frac{1}{\sigma_\eta^2}.$$

- The EM algorithm is initialized with some arbitrary positive values $\alpha^{(0)}$ and $\beta^{(0)}$. The resulting algorithm proceeds as follows:

Linear Regression And The EM Algorithm

- In the current context, the set of observations, which in the general EM formulation were denoted as $\mathcal{X}$, will be our familiar observation vector, $\boldsymbol{y}$. Also, the place of the set of the latent variables, denoted as $\mathcal{X}^l$, is replaced by $\boldsymbol{\theta}$.
- Recall that a prerequisite in order to apply the EM algorithm is the knowledge of the posterior, which for this case is known, given the value of the parameters, σ_η^2 and σ_θ^2.
- We will work with the precision variables and the parameter vector of the unknown variables becomes

$$\boldsymbol{\xi} = [\alpha, \beta]^T, \ \alpha := \frac{1}{\sigma_\theta^2} \text{ and } \beta := \frac{1}{\sigma_\eta^2}.$$

- The EM algorithm is initialized with some arbitrary positive values $\alpha^{(0)}$ and $\beta^{(0)}$. The resulting algorithm proceeds as follows:

- **Algorithm For Optimizing The Unknown Parameters,** $\alpha,\ \beta$.
  - Initialization.
    - Assign $\alpha^{(0)}$ and $\beta^{(0)}$ some positive values.
  - **For** $j = 0, 1, \ldots,$ **Do**
    - Compute:

$$\Sigma_{\theta|y}^{(j)} = \left(\alpha^{(j)} I + \beta^{(j)} \Phi^T \Phi\right)^{-1},$$
$$\boldsymbol{\mu}_{\theta|y}^{(j)} = \beta^{(j)} \Sigma_{\theta|y}^{(j)} \Phi^T \boldsymbol{y}.$$

    - Compute:

$$\alpha^{(j+1)} = \frac{K}{\|\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2 + \text{trace}\{\Sigma_{\theta|y}^{(j)}\}},$$
$$\beta^{(j+1)} = \frac{N}{\|\boldsymbol{y} - \Phi\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2 + \text{trace}\{\Phi\Sigma_{\theta|y}^{(j)}\Phi^T\}}.$$

  - **End For**
  - Stop If a stopping criterion is met.

Proof of the algorithm.

- **E-Step**: This step comprises the computation of the expectation of the complete log-likelihood with respect to the latent variables. The expectation is taken with respect to the posterior. The log-likelihood associated with the complete data set is given by,

$$\ln p(\boldsymbol{y}, \boldsymbol{\theta}; \boldsymbol{\xi}) := \ln p(\boldsymbol{y}, \boldsymbol{\theta}; \alpha, \beta) = \ln \Big(p(\boldsymbol{y}|\boldsymbol{\theta}; \beta) p(\boldsymbol{\theta}; \alpha) \Big),$$

  which for the case of the involved Gaussians becomes,

$$\ln p(\boldsymbol{y}, \boldsymbol{\theta}; \alpha, \beta) = \frac{N}{2} \ln \beta + \frac{K}{2} \ln \alpha - \frac{\beta}{2} \|\boldsymbol{y} - \Phi \boldsymbol{\theta}\|^2 - \frac{\alpha}{2} \boldsymbol{\theta}^T \boldsymbol{\theta} - \left(\frac{N}{2} + \frac{K}{2} \right) \ln(2\pi).$$

- Treating the latent variables as random ones, the expected value of the above, w.r. to θ, is carried out via the posterior, $\mathcal{N}(\boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y})$. To this end, the following computations are in order:

Proof of the algorithm.

- **E-Step**: This step comprises the computation of the expectation of the complete log-likelihood with respect to the latent variables. The expectation is taken with respect to the posterior. The log-likelihood associated with the complete data set is given by,

$$\ln p(\boldsymbol{y}, \boldsymbol{\theta}; \boldsymbol{\xi}) := \ln p(\boldsymbol{y}, \boldsymbol{\theta}; \alpha, \beta) = \ln \Big(p(\boldsymbol{y}|\boldsymbol{\theta}; \beta) p(\boldsymbol{\theta}; \alpha) \Big),$$

  which for the case of the involved Gaussians becomes,

$$\ln p(\boldsymbol{y}, \boldsymbol{\theta}; \alpha, \beta) = \frac{N}{2} \ln \beta + \frac{K}{2} \ln \alpha - \frac{\beta}{2} \|\boldsymbol{y} - \Phi \boldsymbol{\theta}\|^2 - \frac{\alpha}{2} \boldsymbol{\theta}^T \boldsymbol{\theta} - \left(\frac{N}{2} + \frac{K}{2} \right) \ln(2\pi).$$

- Treating the latent variables as random ones, the expected value of the above, w.r. to θ, is carried out via the posterior, $\mathcal{N}(\boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y})$. To this end, the following computations are in order:

- E-Step continued:
  1. To compute $\mathbb{E}[\theta^T\theta]$, recall the definition of the respective covariance matrix,

$$\Sigma_{\theta|y}^{(j)} = \mathbb{E}\left[(\theta - \boldsymbol{\mu}_{\theta|y}^{(j)})(\theta - \boldsymbol{\mu}_{\theta|y}^{(j)})^T\right]$$

     or

$$\mathbb{E}[\theta\theta^T] = \Sigma_{\theta|y}^{(j)} + \boldsymbol{\mu}_{\theta|y}^{(j)}\boldsymbol{\mu}_{\theta|y}^{(j)T},$$

     which results to

$$\begin{aligned} A := \mathbb{E}[\theta^T\theta] &= \mathbb{E}[\mathrm{trace}\{\theta\theta^T\}] \\ &= \mathrm{trace}\{\boldsymbol{\mu}_{\theta|y}^{(j)}\boldsymbol{\mu}_{\theta|y}^{(j)T} + \Sigma_{\theta|y}^{(j)}\} \\ &= \|\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2 + \mathrm{trace}\{\Sigma_{\theta|y}^{(j)}\}. \end{aligned}$$

  2. To compute $\mathbb{E}[\|\boldsymbol{y} - \Phi\theta\|^2]$, define $\boldsymbol{\psi} := \boldsymbol{y} - \Phi\theta$, and use the previous rationale to compute $\mathbb{E}[\boldsymbol{\psi}^T\boldsymbol{\psi}]$, which leads to

$$B := \mathbb{E}[\|\boldsymbol{y} - \Phi\theta\|^2] = \|\boldsymbol{y} - \Phi\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2 + \mathrm{trace}\{\Phi\Sigma_{\theta|y}^{(j)}\Phi^T\}.$$

     Hence,

$$\mathcal{Q}(\alpha, \beta; \alpha^{(j)}, \beta^{(j)}) = \frac{N}{2}\ln\beta + \frac{K}{2}\ln\alpha - \frac{\beta}{2}B - \frac{\alpha}{2}A - \left(\frac{N}{2} + \frac{K}{2}\right)\ln(2\pi).$$

- **M-Step**: In this step, maximization of the $\mathcal{Q}$ function with respect to α and β is performed to provide their updated estimates. Thus,

$$\alpha^{(j+1)} : \frac{\partial}{\partial \alpha} \mathcal{Q}(\alpha, \beta; \alpha^{(j)}, \beta^{(j)}) = 0$$
$$\beta^{(j+1)} : \frac{\partial}{\partial \beta} \mathcal{Q}(\alpha, \beta; \alpha^{(j)}, \beta^{(j)}) = 0,$$

  which trivially lead to the two algorithmic steps, i.e.,

$$\alpha^{(j+1)} = \frac{K}{\|\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2 + \text{trace}\{\Sigma_{\theta|y}^{(j)}\}},$$
$$\beta^{(j+1)} = \frac{N}{\|\boldsymbol{y} - \Phi\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2 + \text{trace}\{\Phi\Sigma_{\theta|y}^{(j)}\Phi^T\}}.$$

Linear Regression Example Via The EM Algorithm

- We return to the same example, which we treated already, concerning the regression model,

$$y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \theta_3 x_n^3 + \theta_5 x_n^5 + \eta_n, \; n = 1, 2, \ldots, N,$$

- The variance of the Gaussian noise used in the model to generate the data was set equal to $\sigma_\eta^2 = 0.05$. The number of training points was $N = 500$. For the EM algorithm, both initial values $\alpha^{(0)}$ and $\beta^{(0)}$ were set equal to one. The correct dimensionality for the unknown parameter vector, $\boldsymbol{\theta}$, was used.
- The recovered values after the convergence of the EM were, $\alpha = 1.32$ corresponding to $\sigma_\theta^2 = 0.756$ and $\beta = 19.96$ corresponding to $\sigma_\eta^2 = 0.0501$. Note that the latter is very close to the true variance of the noise.
- Having obtained the optimal values for σ_η^2 and σ_θ^2, we can use them to perform predictions of the output variable y at twenty points, using (6) and the value of $\boldsymbol{\mu}_{\theta|y}$ as computed by the EM algorithm.
- The obtained results are summarized by the following figures in the next slide:

Linear Regression Example Via The EM Algorithm

- We return to the same example, which we treated already, concerning the regression model,

$$y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \theta_3 x_n^3 + \theta_5 x_n^5 + \eta_n, \; n = 1, 2, \ldots, N,$$

- The variance of the Gaussian noise used in the model to generate the data was set equal to $\sigma_\eta^2 = 0.05$. The number of training points was $N = 500$. For the EM algorithm, both initial values $\alpha^{(0)}$ and $\beta^{(0)}$ were set equal to one. The correct dimensionality for the unknown parameter vector, $\boldsymbol{\theta}$, was used.
- The recovered values after the convergence of the EM were, $\alpha = 1.32$ corresponding to $\sigma_\theta^2 = 0.756$ and $\beta = 19.96$ corresponding to $\sigma_\eta^2 = 0.0501$. Note that the latter is very close to the true variance of the noise.
- Having obtained the optimal values for σ_η^2 and σ_θ^2, we can use them to perform predictions of the output variable y at twenty points, using (6) and the value of $\boldsymbol{\mu}_{\theta|y}$ as computed by the EM algorithm.
- The obtained results are summarized by the following figures in the next slide:

Linear Regression Example Via The EM Algorithm

- We return to the same example, which we treated already, concerning the regression model,

$$y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \theta_3 x_n^3 + \theta_5 x_n^5 + \eta_n, \; n = 1, 2, \ldots, N,$$

- The variance of the Gaussian noise used in the model to generate the data was set equal to $\sigma_\eta^2 = 0.05$. The number of training points was $N = 500$. For the EM algorithm, both initial values $\alpha^{(0)}$ and $\beta^{(0)}$ were set equal to one. The correct dimensionality for the unknown parameter vector, $\boldsymbol{\theta}$, was used.
- The recovered values after the convergence of the EM were, $\alpha = 1.32$ corresponding to $\sigma_\theta^2 = 0.756$ and $\beta = 19.96$ corresponding to $\sigma_\eta^2 = 0.0501$. Note that the latter is very close to the true variance of the noise.
- Having obtained the optimal values for σ_η^2 and σ_θ^2, we can use them to perform predictions of the output variable y at twenty points, using (6) and the value of $\boldsymbol{\mu}_{\theta|y}$ as computed by the EM algorithm.
- The obtained results are summarized by the following figures in the next slide:

Linear Regression Example Via The EM Algorithm

- We return to the same example, which we treated already, concerning the regression model,

$$y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \theta_3 x_n^3 + \theta_5 x_n^5 + \eta_n, \; n = 1, 2, \ldots, N,$$

- The variance of the Gaussian noise used in the model to generate the data was set equal to $\sigma_\eta^2 = 0.05$. The number of training points was $N = 500$. For the EM algorithm, both initial values $\alpha^{(0)}$ and $\beta^{(0)}$ were set equal to one. The correct dimensionality for the unknown parameter vector, $\boldsymbol{\theta}$, was used.
- The recovered values after the convergence of the EM were, $\alpha = 1.32$ corresponding to $\sigma_\theta^2 = 0.756$ and $\beta = 19.96$ corresponding to $\sigma_\eta^2 = 0.0501$. Note that the latter is very close to the true variance of the noise.
- Having obtained the optimal values for σ_η^2 and σ_θ^2, we can use them to perform predictions of the output variable y at twenty points, using (6) and the value of $\boldsymbol{\mu}_{\theta|y}$ as computed by the EM algorithm.
- The obtained results are summarized by the following figures in the next slide:

Linear Regression Example Via The EM Algorithm

- We return to the same example, which we treated already, concerning the regression model,

$$y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \theta_3 x_n^3 + \theta_5 x_n^5 + \eta_n, \; n = 1, 2, \ldots, N,$$

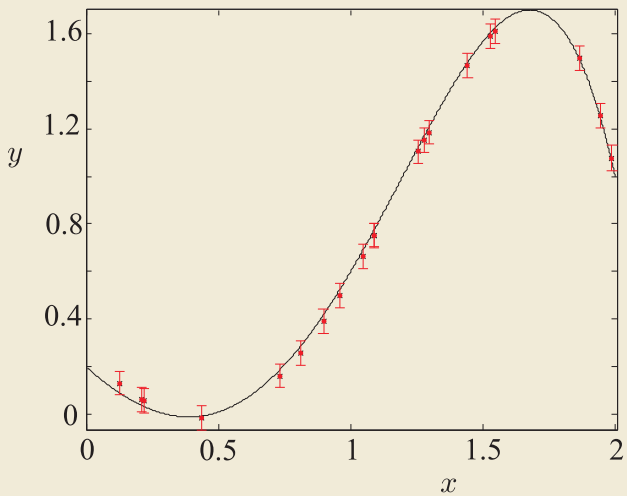
- The variance of the Gaussian noise used in the model to generate the data was set equal to $\sigma_\eta^2 = 0.05$. The number of training points was $N = 500$. For the EM algorithm, both initial values $\alpha^{(0)}$ and $\beta^{(0)}$ were set equal to one. The correct dimensionality for the unknown parameter vector, $\boldsymbol{\theta}$, was used.
- The recovered values after the convergence of the EM were, $\alpha = 1.32$ corresponding to $\sigma_\theta^2 = 0.756$ and $\beta = 19.96$ corresponding to $\sigma_\eta^2 = 0.0501$. Note that the latter is very close to the true variance of the noise.
- Having obtained the optimal values for σ_η^2 and σ_θ^2, we can use them to perform predictions of the output variable y at twenty points, using (6) and the value of $\boldsymbol{\mu}_{\theta|y}$ as computed by the EM algorithm.
- The obtained results are summarized by the following figures in the next slide:

Linear Regression Example Via The EM Algorithm

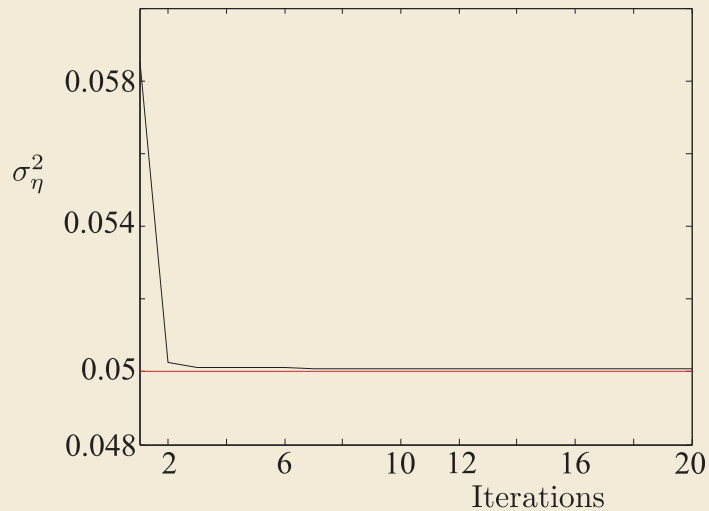
Recall that, for Gaussian prior and conditional, the pdf for the predicted value of y, associated with the observed vector $\boldsymbol{x}$, is given by:

$$p(y|\boldsymbol{x},\boldsymbol{y}) = \mathcal{N}\left(y|\mu_y,\sigma_y^2\right), \text{ where}$$

$$\mu_y = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\mu}_{\theta|y}, \quad \sigma_y^2 = \sigma_\eta^2 + \sigma_\eta^2\sigma_\theta^2\boldsymbol{\phi}^T(\boldsymbol{x})\left(\sigma_\eta^2 I + \sigma_\theta^2\Phi^T\Phi\right)^{-1}\boldsymbol{\phi}(\boldsymbol{x}).$$



(a)



(b)

a) The original graph from which the training points were sampled. In red, the respective predictions $\hat{y}$ and associated error bars for twenty randomly chosen points. b) The convergence curve for σ_η^2 as a function of the iterations of the EM algorithm. The red line corresponds to the true value.

Gaussian Mixture Models

- Often in practice, existing probability distributions models (e.g., Gaussian, gamma, exponential, Dirichlet) are not sufficient to provide a good enough description of the randomness that underlies the data at hand. An alternative path is via mixture models.
- Mixture modeling offers the freedom to model an unknown pdf, $p(\boldsymbol{x})$, as a linear combination of different distributions, i.e.,

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k p(\boldsymbol{x}|k),$$

where P_k are the respective weighting parameters associated with the corresponding contributing pdf, $p(\boldsymbol{x}|k)$. In order to guarantee that $p(\boldsymbol{x})$ is a pdf, the weighting parameters must be non-negative and add to one ($\sum_{k=1}^{K} P_k = 1$).

Gaussian Mixture Models

- Often in practice, existing probability distributions models (e.g., Gaussian, gamma, exponential, Dirichlet) are not sufficient to provide a good enough description of the randomness that underlies the data at hand. An alternative path is via mixture models.
- Mixture modeling offers the freedom to model an unknown pdf, $p(\boldsymbol{x})$, as a linear combination of different distributions, i.e.,

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k p(\boldsymbol{x}|k),$$

where P_k are the respective weighting parameters associated with the corresponding contributing pdf, $p(\boldsymbol{x}|k)$. In order to guarantee that $p(\boldsymbol{x})$ is a pdf, the weighting parameters must be non-negative and add to one ($\sum_{k=1}^{K} P_k = 1$).

Gaussian Mixture Models

- The physical interpretation of the previous combination is the following:
- We are given a set of K distributions, $p(\boldsymbol{x}|k),\ k = 1, 2, \ldots, K$. Each observation, $\boldsymbol{x}_n,\ n = 1, 2, \ldots, N$, is drawn from **one** of these K distributions, but we are not told from which one. All we know is a set of parameters, $P_k,\ 1, 2, \ldots, K$, each one providing the probability that a sample has been drawn from the corresponding pdf, $p(\boldsymbol{x}|k)$.
- It can be shown that, for large enough number of mixtures, K, and appropriate choice of the involved parameters, one can approximate arbitrary close **any** continuous pdf.

Gaussian Mixture Models

- The physical interpretation of the previous combination is the following:
- We are given a set of K distributions, $p(\boldsymbol{x}|k),\ k = 1, 2, \ldots, K$. Each observation, $\boldsymbol{x}_n,\ n = 1, 2, \ldots, N$, is drawn from **one** of these K distributions, but we are not told from which one. All we know is a set of parameters, $P_k,\ 1, 2, \ldots, K$, each one providing the probability that a sample has been drawn from the corresponding pdf, $p(\boldsymbol{x}|k)$.
- It can be shown that, for large enough number of mixtures, K, and appropriate choice of the involved parameters, one can approximate arbitrary close **any** continuous pdf.

Gaussian Mixture Models

- The physical interpretation of the previous combination is the following:
- We are given a set of K distributions, $p(\boldsymbol{x}|k),\ k = 1, 2, \ldots, K$. Each observation, $\boldsymbol{x}_n,\ n = 1, 2, \ldots, N$, is drawn from **one** of these K distributions, but we are not told from which one. All we know is a set of parameters, $P_k,\ 1, 2, \ldots, K$, each one providing the probability that a sample has been drawn from the corresponding pdf, $p(\boldsymbol{x}|k)$.
- It can be shown that, for large enough number of mixtures, K, and appropriate choice of the involved parameters, one can approximate arbitrary close **any** continuous pdf.

Gaussian Mixture Models

- Mixture modeling is a typical task involving hidden variables; these are the labels, k, of the pdf from which an obtained observation has originated. In practice, each $p(\boldsymbol{x}|k)$ is chosen from a known pdf family, parameterized via a set of parameters, $\boldsymbol{\xi}_k$, and we can write

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k p(\boldsymbol{x}|k; \boldsymbol{\xi}_k),$$

- The learning task is to estimate $(P_k, \boldsymbol{\xi}_k),\ k = 1, 2, \ldots, K$, based on a set of observations $\boldsymbol{x}_n,\ n = 1, 2, \ldots, N$.

Gaussian Mixture Models

- Mixture modeling is a typical task involving hidden variables; these are the labels, k, of the pdf from which an obtained observation has originated. In practice, each $p(\boldsymbol{x}|k)$ is chosen from a known pdf family, parameterized via a set of parameters, $\boldsymbol{\xi}_k$, and we can write

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k p(\boldsymbol{x}|k; \boldsymbol{\xi}_k),$$

- The learning task is to estimate $(P_k, \boldsymbol{\xi}_k),\ k = 1, 2, \ldots, K$, based on a set of observations $\boldsymbol{x}_n,\ n = 1, 2, \ldots, N$.

Gaussian Mixture Models

- The set of observations, $\mathcal{X}$, forms the incomplete set while the complete set $\{\mathcal{X}, \mathcal{K}\}$ comprises the samples $(\boldsymbol{x}_n, k_n)$, $n = 1, \ldots, N$, with k_n being the label of the distribution from which $\boldsymbol{x}_n$ was drawn.
- Parameter estimation for such a problem naturally lends itself to be treated via the EM algorithm. We will demonstrate the procedure via the use of Gaussian mixtures.

Gaussian Mixture Models

- The set of observations, $\mathcal{X}$, forms the incomplete set while the complete set $\{\mathcal{X}, \mathcal{K}\}$ comprises the samples $(\boldsymbol{x}_n, k_n)$, $n = 1, \ldots, N$, with k_n being the label of the distribution from which $\boldsymbol{x}_n$ was drawn.
- Parameter estimation for such a problem naturally lends itself to be treated via the EM algorithm. We will demonstrate the procedure via the use of Gaussian mixtures.

Gaussian Mixture Models

- Let

$$p(\boldsymbol{x}|k;\boldsymbol{\xi}_k) = p(\boldsymbol{x}|k;\boldsymbol{\mu}_k,\Sigma_k) = \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_k,\Sigma_k\right),$$

where for simplicity we will assume that $\Sigma_k = \sigma_k^2 I,\ k = 1,\ldots,K$. We will further assume that the observations are i.i.d. For such a modeling, the following holds true:

  - The log-likelihood of the complete data set is given by,

$$\ln p\left(\mathcal{X},\mathcal{K};\Xi,\boldsymbol{P}\right) = \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n,k_n;\boldsymbol{\xi}_{k_n}) = \sum_{n=1}^{N} \ln\Big(p(\boldsymbol{x}_n|k_n;\boldsymbol{\xi}_{k_n})P_{k_n}\Big).$$

    We have used the notation,

$$\Xi = [\boldsymbol{\xi}_1^T,\ldots,\boldsymbol{\xi}_K^T]^T,\ \boldsymbol{P} = [P_1,P_2,\ldots,P_K]^T,\ \text{and}\ \boldsymbol{\xi}_k = [\boldsymbol{\mu}_k^T,\sigma_k^2]^T.$$

Gaussian Mixture Models

- Let

$$p(\boldsymbol{x}|k;\boldsymbol{\xi}_k) = p(\boldsymbol{x}|k;\boldsymbol{\mu}_k,\Sigma_k) = \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_k,\Sigma_k\right),$$

where for simplicity we will assume that $\Sigma_k = \sigma_k^2 I,\ k = 1,\ldots,K$. We will further assume that the observations are i.i.d. For such a modeling, the following holds true:

- The log-likelihood of the complete data set is given by,

$$\ln p\left(\mathcal{X},\mathcal{K};\boldsymbol{\Xi},\boldsymbol{P}\right) = \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n,k_n;\boldsymbol{\xi}_{k_n}) = \sum_{n=1}^{N} \ln\Big(p(\boldsymbol{x}_n|k_n;\boldsymbol{\xi}_{k_n})P_{k_n}\Big).$$

We have used the notation,

$$\boldsymbol{\Xi} = [\boldsymbol{\xi}_1^T,\ldots,\boldsymbol{\xi}_K^T]^T,\ \ \boldsymbol{P} = [P_1,P_2,\ldots,P_K]^T,\ \text{ and }\ \boldsymbol{\xi}_k = [\boldsymbol{\mu}_k^T,\sigma_k^2]^T.$$

Gaussian Mixture Models

- For the EM, we need to know the posterior probabilities of the discrete hidden variables.
  - These are given by

$$P(k|\boldsymbol{x};\boldsymbol{\Xi},\boldsymbol{P})=\frac{p(\boldsymbol{x},k;\boldsymbol{\Xi},\boldsymbol{P})}{p(\boldsymbol{x};\boldsymbol{\Xi},\boldsymbol{P})}=\frac{p(\boldsymbol{x}|k;\boldsymbol{\xi}_k)P_k}{p(\boldsymbol{x};\boldsymbol{\Xi},\boldsymbol{P})}, \tag{10}$$

    where

$$p(\boldsymbol{x};\boldsymbol{\Xi},\boldsymbol{P})=\sum_{k=1}^{K}P_k p(\boldsymbol{x}|k;\boldsymbol{\xi}_k).$$

- We have now all the ingredients required by the EM algorithm. Starting from $\Xi^{(0)}$ and $\boldsymbol{P}^{(0)}$, the following algorithm results for the computation of the unknown parameters, $\boldsymbol{\mu}_k$, σ_k^2, P_k, $k=1,2,\ldots,K$.

Gaussian Mixture Models

- For the EM, we need to know the posterior probabilities of the discrete hidden variables.
  - These are given by

$$P(k|\boldsymbol{x};\boldsymbol{\Xi},\boldsymbol{P})=\frac{p(\boldsymbol{x},k;\boldsymbol{\Xi},\boldsymbol{P})}{p(\boldsymbol{x};\boldsymbol{\Xi},\boldsymbol{P})}=\frac{p(\boldsymbol{x}|k;\boldsymbol{\xi}_k)P_k}{p(\boldsymbol{x};\boldsymbol{\Xi},\boldsymbol{P})}, \tag{10}$$

    where

$$p(\boldsymbol{x};\boldsymbol{\Xi},\boldsymbol{P})=\sum_{k=1}^{K}P_k p(\boldsymbol{x}|k;\boldsymbol{\xi}_k).$$

- We have now all the ingredients required by the EM algorithm. Starting from $\boldsymbol{\Xi}^{(0)}$ and $\boldsymbol{P}^{(0)}$, the following algorithm results for the computation of the unknown parameters, $\boldsymbol{\mu}_k$, σ_k^2, P_k, $k=1,2,\ldots,K$.

- **Algorithm For The Gaussian Mixture Model**
  - Initialization.
    - Assign values to $\boldsymbol{\mu}_k^{(0)},\ k = 1, 2, \ldots, K$.
    - Assign positive values to $\sigma_k^{2(0)},\ k = 1, 2, \ldots, K$.
    - Assign values to $P_k^{(0)},\ k = 1, 2, \ldots, K$, such as $\sum_{k=1}^{K} P_k^{(0)} = 1$.
  - **For** $j = 1, 2, \ldots,$ **Do**
    - Set
$$\gamma_{kn} := P(k|\boldsymbol{x}_n; \boldsymbol{\Xi}^{(j)}, \boldsymbol{P}^{(j)}).$$
    - Compute:
$$\boldsymbol{\mu}_k^{(j+1)} = \frac{\sum_{n=1}^{N} \gamma_{kn} \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma_{kn}}, \tag{11}$$
$$\sigma_k^{2(j+1)} = \frac{\sum_{n=1}^{N} \gamma_{kn} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(j+1)}\|^2}{l \sum_{n=1}^{N} \gamma_{kn}},$$
$$P_k^{(j+1)} = \frac{1}{N} \sum_{n=1}^{N} \gamma_{kn}.$$
  - **End For**
  - Stop if a stopping criterion is met.

Gaussian Mixture Models

- The extension to the case of a general covariance matrix is straightforward by replacing the variance update equation by,

$$\Sigma_k^{(j+1)} = \frac{\sum_{n=1}^{N} \gamma_{kn}(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(j+1)})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(j+1)})^T}{\sum_{n=1}^{N} \gamma_{kn}}.$$

- Some Remarks
  - In order to get good initialization for the EM algorithm, sometimes a simpler clustering algorithm, e.g., the k-means (to be discussed soon) is run to provide an initial estimate of the means and shapes of clusters (covariance matrices), by associating each mixture with a cluster in the input space. Another simpler way is to select K points randomly from the data set. A more elaborate technique, which is commonly used, is to select them randomly but in such a way so that to make sure that the whole data set is represented in a balanced way.
  - The number of mixtures, K, is usually determined by cross-validation.

Gaussian Mixture Models

- The extension to the case of a general covariance matrix is straightforward by replacing the variance update equation by,

$$\Sigma_k^{(j+1)} = \frac{\sum_{n=1}^{N} \gamma_{kn}(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(j+1)})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(j+1)})^T}{\sum_{n=1}^{N} \gamma_{kn}}.$$

- Some Remarks
  - In order to get good initialization for the EM algorithm, sometimes a simpler clustering algorithm, e.g., the k-means (to be discussed soon) is run to provide an initial estimate of the means and shapes of clusters (covariance matrices), by associating each mixture with a cluster in the input space. Another simpler way is to select K points randomly from the data set. A more elaborate technique, which is commonly used, is to select them randomly but in such a way so that to make sure that the whole data set is represented in a balanced way.
  - The number of mixtures, K, is usually determined by cross-validation.

Gaussian Mixture Models

- The extension to the case of a general covariance matrix is straightforward by replacing the variance update equation by,

$$\Sigma_k^{(j+1)} = \frac{\sum_{n=1}^{N} \gamma_{kn}(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(j+1)})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(j+1)})^T}{\sum_{n=1}^{N} \gamma_{kn}}.$$

- Some Remarks
  - In order to get good initialization for the EM algorithm, sometimes a simpler clustering algorithm, e.g., the k-means (to be discussed soon) is run to provide an initial estimate of the means and shapes of clusters (covariance matrices), by associating each mixture with a cluster in the input space. Another simpler way is to select K points randomly from the data set. A more elaborate technique, which is commonly used, is to select them randomly but in such a way so that to make sure that the whole data set is represented in a balanced way.
  - The number of mixtures, K, is usually determined by cross-validation.

Proof of the algorithm

- **E-Step**: Combining the log-likelihood and the posterior in the form of (10), the corresponding expectation results in

$$\begin{aligned}\mathcal{Q}(\Xi, \boldsymbol{P}; \Xi^{(j)}, \boldsymbol{P}^{(j)}) &= \sum_{n=1}^{N} \mathbb{E}\Big[\ln\Big(p(\boldsymbol{x}_n|k_n;\boldsymbol{\xi}_{k_n})P_{k_n}\Big)\Big] \\ &:= \sum_{n=1}^{N}\sum_{k=1}^{K} P(k|\boldsymbol{x}_n;\Xi^{(j)},\boldsymbol{P}^{(j)})\Big(\ln P_k - \frac{l}{2}\ln\sigma_k^2 \\ &\quad -\frac{1}{2\sigma_k^2}\|\boldsymbol{x}_n-\boldsymbol{\mu}_k\|^2\Big) + C,\end{aligned}$$

  where C includes all the constant terms. Note that we have finally relaxed the notation from k_n to k, since we sum up over all k, which does not depend on n.

- **M-Step**: Maximization of $\mathcal{Q}(\Xi, \boldsymbol{P}; \Xi^{(j)}, \boldsymbol{P}^{(j)})$ w.r. to all the involved parameters results in the set of recursions given in the algorithm before. Note that maximizing with respect to P_k, $k = 1, 2, \ldots, K$, is a constrained optimization task, because probabilities have to add to one.

Proof of the algorithm

- **E-Step**: Combining the log-likelihood and the posterior in the form of (10), the corresponding expectation results in

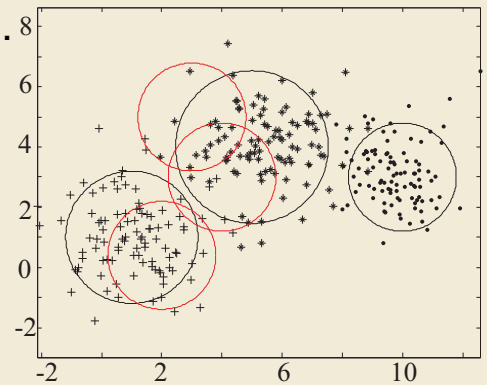
$$\begin{aligned}\mathcal{Q}(\boldsymbol{\Xi}, \boldsymbol{P}; \boldsymbol{\Xi}^{(j)}, \boldsymbol{P}^{(j)}) &= \sum_{n=1}^{N} \mathbb{E}\Big[\ln \Big(p(\boldsymbol{x}_n | k_n; \boldsymbol{\xi}_{k_n}) P_{k_n} \Big) \Big] \\ &:= \sum_{n=1}^{N} \sum_{k=1}^{K} P(k|\boldsymbol{x}_n; \boldsymbol{\Xi}^{(j)}, \boldsymbol{P}^{(j)}) \Big(\ln P_k - \frac{l}{2} \ln \sigma_k^2 \\ &\quad - \frac{1}{2\sigma_k^2} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \Big) + C,\end{aligned}$$

  where C includes all the constant terms. Note that we have finally relaxed the notation from k_n to k, since we sum up over all k, which does not depend on n.

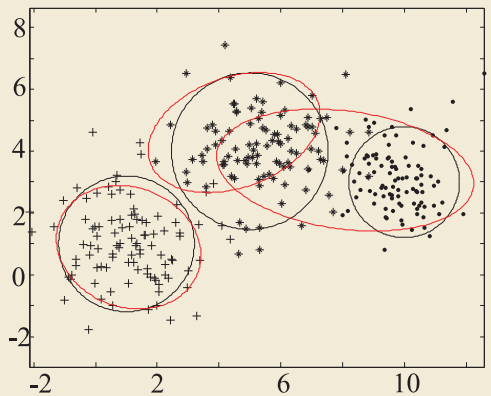
- **M-Step**: Maximization of $\mathcal{Q}(\boldsymbol{\Xi}, \boldsymbol{P}; \boldsymbol{\Xi}^{(j)}, \boldsymbol{P}^{(j)})$ w.r. to all the involved parameters results in the set of recursions given in the algorithm before. Note that maximizing with respect to P_k, $k = 1, 2, \ldots, K$, is a constrained optimization task, because probabilities have to add to one.

Example on Gaussian Mixture Modeling

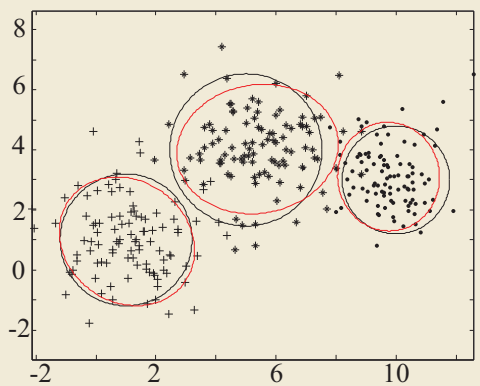
- The data are generated according to three (equiprobable) Gaussians. Each Gaussian has different mean and covariance matrix, with values reported in the book. The number of the generated points is 300 with 100 points per mixture. The points are shown in the figures below together with the gray circles, which indicate the 80% probability regions for each one of the clusters.



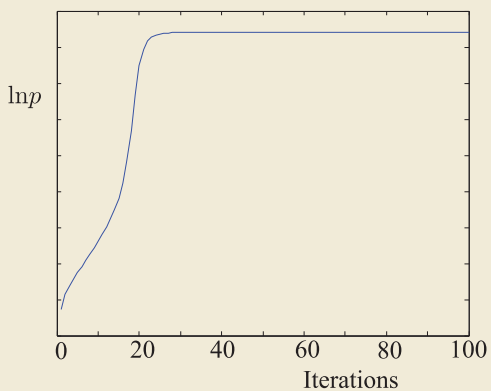
(a)



(b)



(c)

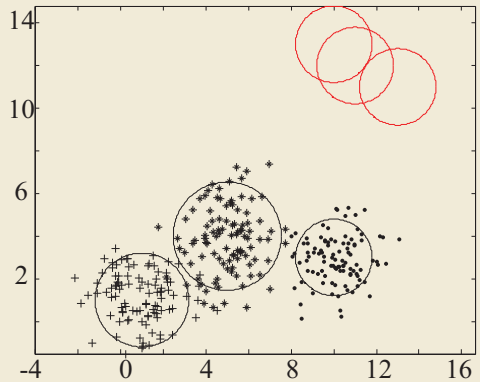


(d)

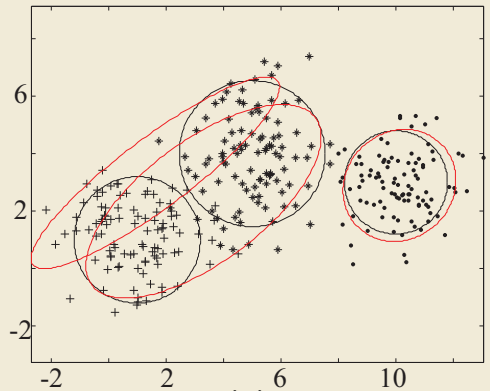
The curves (ellipses) indicate the 80% probability regions. The gray curves correspond to the **true** Gaussian clusters. The red curves correspond to a) the initial values for the mean and covariance matrices, (b) to the recovered by the EM algorithm mixtures after 5 iterations and (c) after 30 iterations. (d) The log-likelihood as a function of the number of iterations. Probabilities were initialized to their true (equal) values.

Example on Gaussian Mixture Modeling

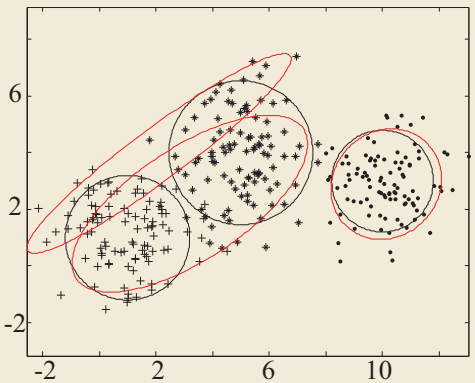
- The figures below correspond to a different setup. The number of points remains the same as before, but the clusters were initialized with mean values very far from the true ones. The covariances and probabilities were initialized as before. Observe that in this case, the EM algorithm fails to capture the true nature of the problem, having been trapped in a local minimum.



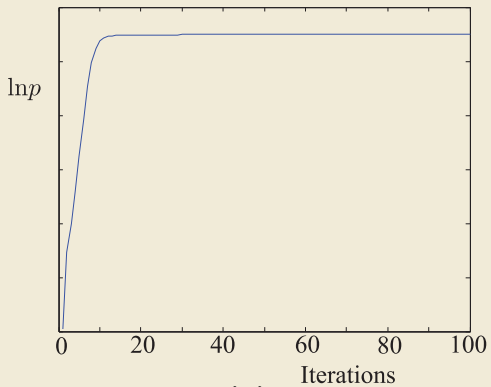
(a)



(b)



(c)



(d)

As before, the red curves correspond to a) the initial values for the mean, covariance matrices, (b) to the recovered by the EM algorithm mixtures after 5 iterations and (c) after 30 iterations. (d) The log-likelihood as a functions of the number of iterations. The EM fails to recover the clusters.

Mixture Modeling and Clustering

- The task of clustering is to assign a number of points, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, into K groups or clusters. Points which are assigned to the same cluster must be more "similar" than points which are assigned to other clusters.
- A major issue in clustering is to quantify "similarity". Different definitions end up with different clusterings. A clustering is a **specific** allocation of the points to clusters.
- In general, assigning points to clusters, according to an optimality criterion, is an NP-hard task. Thus, in general, any clustering algorithm provides a suboptimal solution.

Mixture Modeling and Clustering

- The task of clustering is to assign a number of points, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, into K groups or clusters. Points which are assigned to the same cluster must be more "similar" than points which are assigned to other clusters.
- A major issue in clustering is to quantify "similarity". Different definitions end up with different clusterings. A clustering is a **specific** allocation of the points to clusters.
- In general, assigning points to clusters, according to an optimality criterion, is an NP-hard task. Thus, in general, any clustering algorithm provides a suboptimal solution.

Mixture Modeling and Clustering

- The task of clustering is to assign a number of points, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, into K groups or clusters. Points which are assigned to the same cluster must be more "similar" than points which are assigned to other clusters.
- A major issue in clustering is to quantify "similarity". Different definitions end up with different clusterings. A clustering is a **specific** allocation of the points to clusters.
- In general, assigning points to clusters, according to an optimality criterion, is an NP-hard task. Thus, in general, any clustering algorithm provides a suboptimal solution.

Mixture Modeling and Clustering

- Gaussian mixture modeling is one among the popular clustering algorithms. The main assumption is that the points, which belong to the same cluster, are distributed according to the same Gaussian distribution (this is how similarity is defined in this case), of unknown mean and covariance matrix. Each mixture component defines a different cluster.
- Thus, the goal is to obtain estimates, via the EM, of the posterior probabilities, $P(k|\boldsymbol{x}_n),\ k = 1, 2, \ldots, K,\ n = 1, 2, \ldots, N$, where each k corresponds to a cluster (mixture). Then, each point is assigned to cluster k according to the rule,

$$\text{assign } \boldsymbol{x}_n \text{ to cluster } k = \arg\max_i P(i|\boldsymbol{x}_n),\ i = 1, 2, \ldots, K.$$

Mixture Modeling and Clustering

- Gaussian mixture modeling is one among the popular clustering algorithms. The main assumption is that the points, which belong to the same cluster, are distributed according to the same Gaussian distribution (this is how similarity is defined in this case), of unknown mean and covariance matrix. Each mixture component defines a different cluster.
- Thus, the goal is to obtain estimates, via the EM, of the posterior probabilities, $P(k|\boldsymbol{x}_n),\ k = 1, 2, \ldots, K,\ n = 1, 2, \ldots, N$, where each k corresponds to a cluster (mixture). Then, each point is assigned to cluster k according to the rule,

$$\text{assign } \boldsymbol{x}_n \text{ to cluster } k = \arg\max_i P(i|\boldsymbol{x}_n),\ i = 1, 2, \ldots, K.$$

The k-Means Or Isodata Clustering Algorithm

- In the EM algorithm, the posterior probability of each point, $\boldsymbol{x}_n$, with respect to each one of the clusters, k, is computed recursively. Moreover, the mean value $\boldsymbol{\mu}_k$, of the points associated with cluster k, is computed as a weighted average of all the training points (11).
- In contrast, in the k-means algorithm, at each iteration, the posterior probability gets a binary value in $\{1, 0\}$; for each point, $\boldsymbol{x}_n$, the Euclidean distance from all the currently available estimates of the mean values is computed, and the posterior probability is estimated according to the following rule,

$$P(k|\boldsymbol{x}_n) = \begin{cases} 1 & \text{if } ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 < ||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2, \ j \neq k, \\ 0 & \text{otherwise.} \end{cases}$$

The k-Means Or Isodata Clustering Algorithm

- In the EM algorithm, the posterior probability of each point, $\boldsymbol{x}_n$, with respect to each one of the clusters, k, is computed recursively. Moreover, the mean value $\boldsymbol{\mu}_k$, of the points associated with cluster k, is computed as a weighted average of all the training points (11).
- In contrast, in the k-means algorithm, at each iteration, the posterior probability gets a binary value in $\{1, 0\}$; for each point, $\boldsymbol{x}_n$, the Euclidean distance from all the currently available estimates of the mean values is computed, and the posterior probability is estimated according to the following rule,

$$P(k|\boldsymbol{x}_n) = \begin{cases} 1 & \text{if } ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 < ||\boldsymbol{x}_n - \boldsymbol{\mu}_j||^2, \ j \neq k, \\ 0 & \text{otherwise.} \end{cases}$$

The k-Means Or Isodata Clustering Algorithm

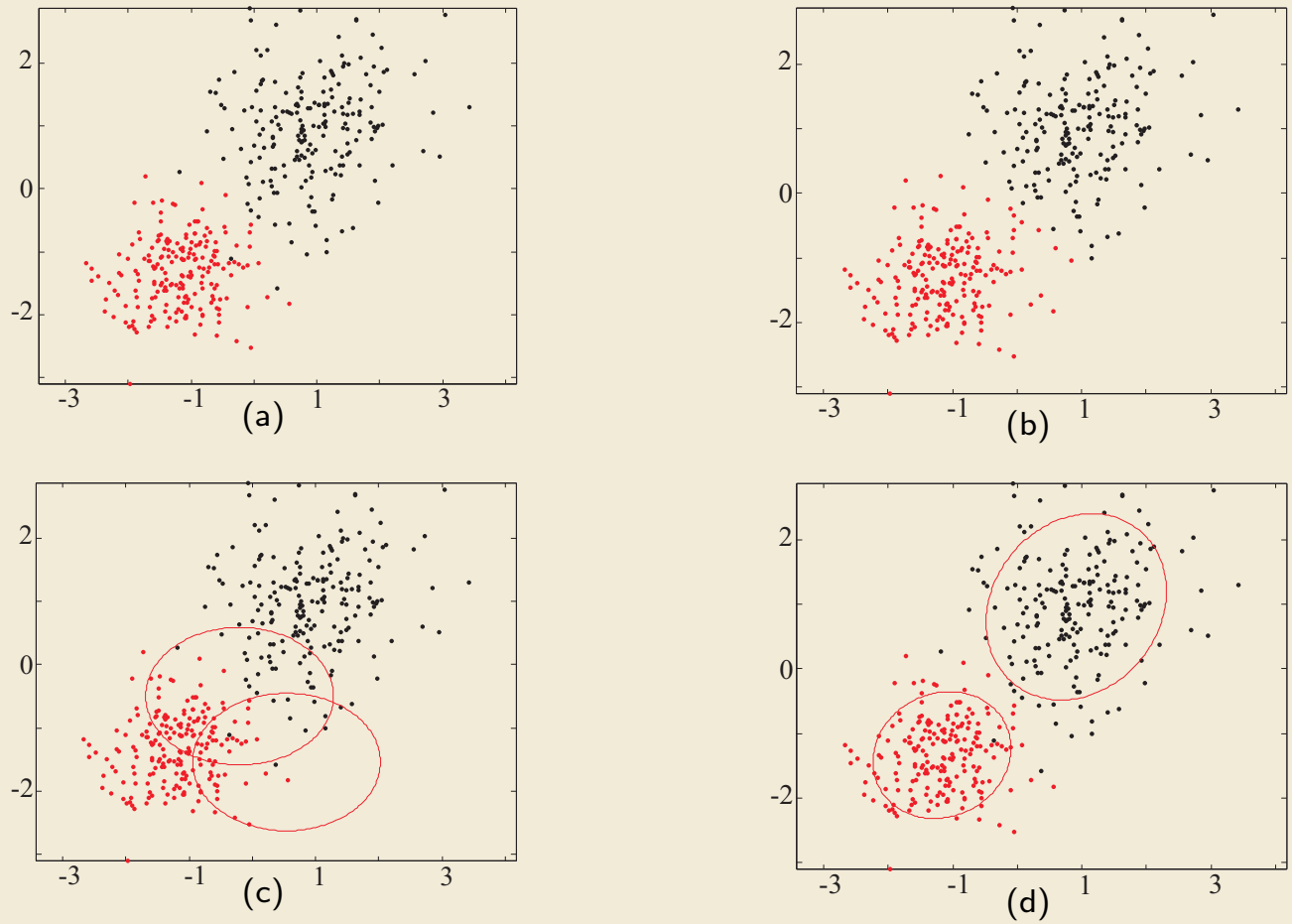
- **The k-Means or Isodata Clustering Algorithm**
  - Initialize:
    - Select the number of clusters K.
    - Set $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, to arbitrarily values.
  - **For** $n = 1, 2, \ldots, N$, **Do**
    - Determine the closest cluster mean, say, $\boldsymbol{\mu}_k$, to $\boldsymbol{x}_n$.
    - Set $b(n) = k$.
  - **End For**
  - **For** $k = 1, 2, \ldots, K$, **Do**
    - Update $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, as the mean of all the points with $b(n) = k,\ n = 1, 2, \ldots, N$.
  - **End For**
  - Until no change in $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, occurs between two successive iterations.
- Note that both the EM algorithms as well as the k-means algorithms can only recover compact clusters. For example, if the points are distributed in ring-shaped clusters, then this type of clustering algorithms is not appropriate.

The k-Means Or Isodata Clustering Algorithm

- **The k-Means or Isodata Clustering Algorithm**
  - Initialize:
    - Select the number of clusters K.
    - Set $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, to arbitrarily values.
  - **For** $n = 1, 2, \ldots, N$, **Do**
    - Determine the closest cluster mean, say, $\boldsymbol{\mu}_k$, to $\boldsymbol{x}_n$.
    - Set $b(n) = k$.
  - **End For**
  - **For** $k = 1, 2, \ldots, K$, **Do**
    - Update $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, as the mean of all the points with $b(n) = k,\ n = 1, 2, \ldots, N$.
  - **End For**
  - Until no change in $\boldsymbol{\mu}_k,\ k = 1, 2, \ldots, K$, occurs between two successive iterations.
- Note that both the EM algorithms as well as the k-means algorithms can only recover compact clusters. For example, if the points are distributed in ring-shaped clusters, then this type of clustering algorithms is not appropriate.

The k-Means and Gaussian Mixtures: Some Examples

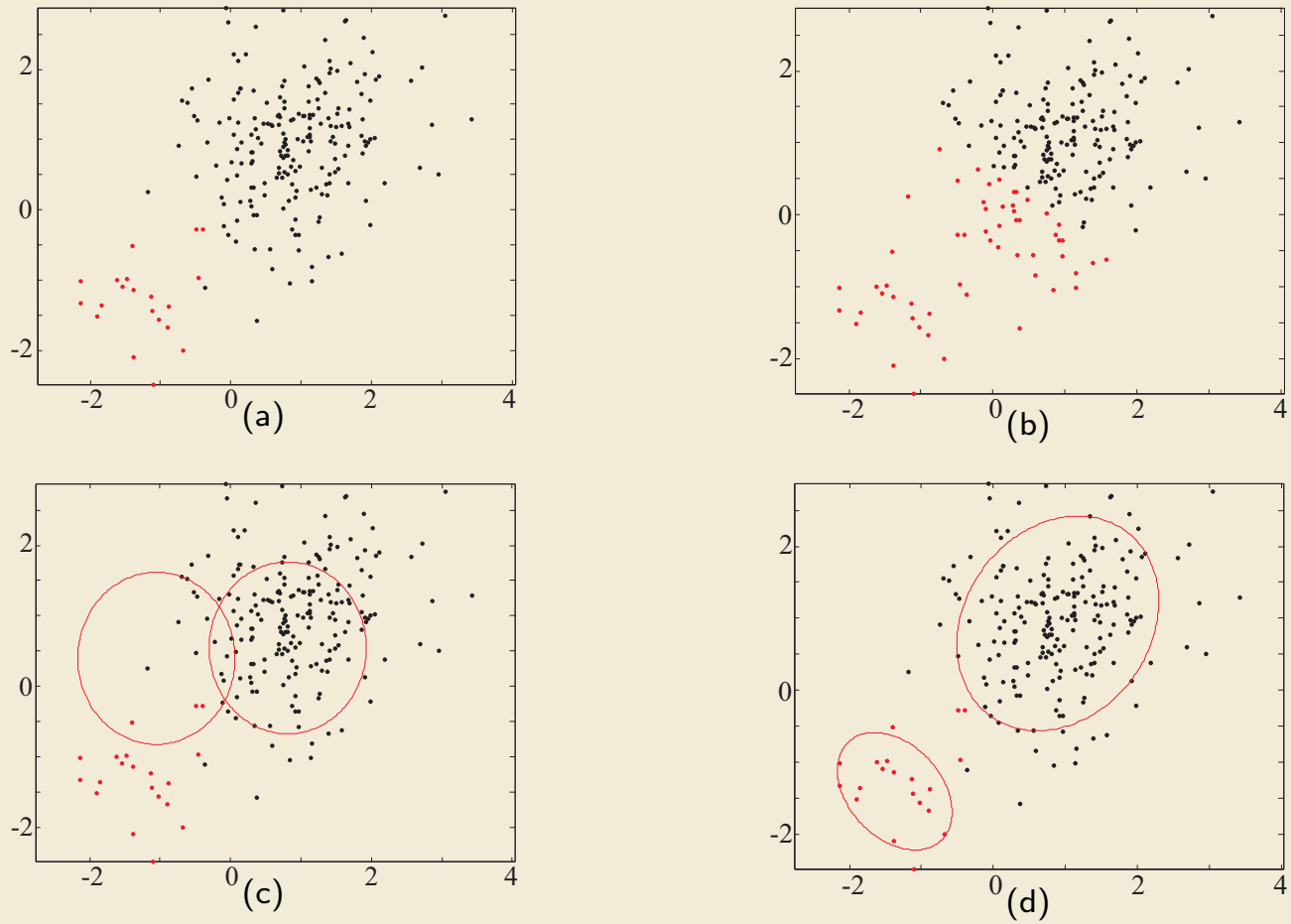
- Figure (a) shows the data points generated by two Gaussians; 200 points from each one. The points are shown by red and gray colors, depending on the Gaussian which generated them. For both, the EM and the k-means algorithm, the correct number of clusters ($K = 2$) was given. The k-means was initialized with zero mean values.



b) the recovered clusters by the k-means (red and gray), c) The 80% probability curves for the initialization of the the EM algorithm and d) the final obtained by the EM algorithm Gaussians with the respective clusters.

The k-Means and Gaussian Mixtures: Some Examples

- The figures correspond to the same Gaussians as before; however, now, there is an imbalance to the number of the points, where only 20 points spring from the first one and 200 points from the second. Observe that the k-means has a problem and it attempts to recover more equally sized clusters.



b) The recovered clusters by the k-means (red and gray). Observe that the algorithm has not identified the correct clusters, by assigning more points to the "smaller" one. c) The 80% probability curves for the initialization of the the EM algorithm and d) the final Gaussians, obtained by the EM algorithm, with the respective clusters.

Looking Deeper: A Lower Bound Maximization View of the EM

- Let us consider the functional

$$\mathcal{F}(q, \boldsymbol{\xi}) := \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l, \tag{12}$$

where $q(\mathcal{X}^l)$ is any nonnegative function that integrates to one; that is, it is a pdf defined over the latent variables. The functional $\mathcal{F}(\cdot, \cdot)$, depends on $\boldsymbol{\xi}$ and on $q(\cdot)$, and its definition bears a strong similarity with the notion of free energy, used in statistical physics. Indeed, the previous can be written as,

$$\mathcal{F}(q, \boldsymbol{\xi}) = \int q(\mathcal{X}^l) \ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi}) d\mathcal{X}^l + H,$$

where,

$$H = -\int q(\mathcal{X}^l) \ln q(\mathcal{X}^l) d\mathcal{X}^l,$$

is the entropy associated with $q(\mathcal{X}^l)$.

- If one defines $-\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})$ as the energy of the system, $(\mathcal{X}, \mathcal{X}^l)$, then $\mathcal{F}(q, \boldsymbol{\xi})$, represents the negative of the so-called free energy.

Looking Deeper: A Lower Bound Maximization View of the EM

- Let us consider the functional

$$\mathcal{F}(q, \boldsymbol{\xi}) := \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l, \tag{12}$$

where $q(\mathcal{X}^l)$ is any nonnegative function that integrates to one; that is, it is a pdf defined over the latent variables. The functional $\mathcal{F}(\cdot, \cdot)$, depends on $\boldsymbol{\xi}$ and on $q(\cdot)$, and its definition bears a strong similarity with the notion of free energy, used in statistical physics. Indeed, the previous can be written as,

$$\mathcal{F}(q, \boldsymbol{\xi}) = \int q(\mathcal{X}^l) \ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi}) d\mathcal{X}^l + H,$$

where,

$$H = -\int q(\mathcal{X}^l) \ln q(\mathcal{X}^l) d\mathcal{X}^l,$$

is the entropy associated with $q(\mathcal{X}^l)$.

- If one defines $-\ln p(\mathcal{X}, \mathcal{X}^l; \xi)$ as the energy of the system, $(\mathcal{X}, \mathcal{X}^l)$, then $\mathcal{F}(q, \xi)$, represents the negative of the so-called free energy.

Looking Deeper: A Lower Bound Maximization View of the EM

- Let us consider the functional

$$\mathcal{F}(q, \boldsymbol{\xi}) := \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l, \tag{12}$$

where $q(\mathcal{X}^l)$ is any nonnegative function that integrates to one; that is, it is a pdf defined over the latent variables. The functional $\mathcal{F}(\cdot, \cdot)$, depends on $\boldsymbol{\xi}$ and on $q(\cdot)$, and its definition bears a strong similarity with the notion of free energy, used in statistical physics. Indeed, the previous can be written as,

$$\mathcal{F}(q, \boldsymbol{\xi}) = \int q(\mathcal{X}^l) \ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi}) d\mathcal{X}^l + H,$$

where,

$$H = -\int q(\mathcal{X}^l) \ln q(\mathcal{X}^l) d\mathcal{X}^l,$$

is the entropy associated with $q(\mathcal{X}^l)$.

- If one defines $-\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})$ as the energy of the system, $(\mathcal{X}, \mathcal{X}^l)$, then $\mathcal{F}(q, \boldsymbol{\xi})$, represents the negative of the so-called free energy.

Looking Deeper: A Lower Bound Maximization View of the EM

- Elaborating on (12), we get

$$\mathcal{F}(q,\boldsymbol{\xi}) = \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})p(\mathcal{X};\boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l,$$

$$= \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l + \ln p(\mathcal{X};\boldsymbol{\xi}), \qquad (13)$$

  where the latter results since $p(\mathcal{X};\boldsymbol{\xi})$ does not depend on $q(\mathcal{X}^l)$.

- The first term on the right hand side is the negative of the so-called Kullback-Leibler divergence between $q(\mathcal{X}^l)$ and $p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})$, which we will denote as $\mathrm{KL}(q \,\|\, p)$.

Looking Deeper: A Lower Bound Maximization View of the EM

- Elaborating on (12), we get

$$\mathcal{F}(q,\boldsymbol{\xi}) = \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})p(\mathcal{X};\boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l,$$

$$= \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l + \ln p(\mathcal{X};\boldsymbol{\xi}), \qquad (13)$$

  where the latter results since $p(\mathcal{X};\boldsymbol{\xi})$ does not depend on $q(\mathcal{X}^l)$.

- The first term on the right hand side is the negative of the so-called Kullback-Leibler divergence between $q(\mathcal{X}^l)$ and $p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})$, which we will denote as $\mathrm{KL}(q \,\|\, p)$.

Looking Deeper: A Lower Bound Maximization View of the EM

- Thus, finally we get

$$\ln p(\mathcal{X}; \boldsymbol{\xi}) = \mathcal{F}(q, \boldsymbol{\xi}) + \mathrm{KL}(q \parallel p). \tag{14}$$

- It is known that $\mathrm{KL}(q \parallel p) \geq 0$; thus, it turns out that

$$\ln p(\mathcal{X}; \boldsymbol{\xi}) \geq \mathcal{F}(q, \boldsymbol{\xi}). \tag{15}$$

- In other words, $\mathcal{F}(q, \boldsymbol{\xi})$ is a **lower bound** of the log-likelihood function, and the bound becomes tight if $\mathrm{KL}(q \parallel p) = 0$, which is true, **if and only if**, $q(\mathcal{X}^l) = p(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi})$.

Looking Deeper: A Lower Bound Maximization View of the EM

- Thus, finally we get

$$\ln p(\mathcal{X}; \boldsymbol{\xi}) = \mathcal{F}(q, \boldsymbol{\xi}) + \mathrm{KL}(q \parallel p). \tag{14}$$

- It is known that $\mathrm{KL}(q \parallel p) \geq 0$; thus, it turns out that

$$\ln p(\mathcal{X}; \boldsymbol{\xi}) \geq \mathcal{F}(q, \boldsymbol{\xi}). \tag{15}$$

- In other words, $\mathcal{F}(q, \boldsymbol{\xi})$ is a **lower bound** of the log-likelihood function, and the bound becomes tight if $\mathrm{KL}(q \parallel p) = 0$, which is true, **if and only if**, $q(\mathcal{X}^l) = p(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi})$.

Looking Deeper: A Lower Bound Maximization View of the EM

- Thus, finally we get

$$\ln p(\mathcal{X};\boldsymbol{\xi}) = \mathcal{F}(q,\boldsymbol{\xi}) + \mathrm{KL}(q \parallel p). \tag{14}$$

- It is known that $\mathrm{KL}(q \parallel p) \geq 0$; thus, it turns out that

$$\ln p(\mathcal{X};\boldsymbol{\xi}) \geq \mathcal{F}(q,\boldsymbol{\xi}). \tag{15}$$

- In other words, $\mathcal{F}(q,\boldsymbol{\xi})$ is a **lower bound** of the log-likelihood function, and the bound becomes tight if $\mathrm{KL}(q \parallel p) = 0$, which is true, **if and only if**, $q(\mathcal{X}^l) = p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})$.

Looking Deeper: A Lower Bound Maximization View of the EM

- The previous findings pave the way of maximizing $\ln p(\mathcal{X};\xi)$ by trying to maximize its lower bound.
- Note that maximization of $\mathcal{F}(\cdot,\cdot)$ involves two terms, namely q, ξ. We will adopt a procedure that belongs to a more general class of optimization algorithms known as alternating optimization. Such an approach naturally imposes an iterative procedure.

Looking Deeper: A Lower Bound Maximization View of the EM

- The previous findings pave the way of maximizing $\ln p(\mathcal{X}; \xi)$ by trying to maximize its lower bound.
- Note that maximization of $\mathcal{F}(\cdot, \cdot)$ involves two terms, namely q, $\boldsymbol{\xi}$. We will adopt a procedure that belongs to a more general class of optimization algorithms known as alternating optimization. Such an approach naturally imposes an iterative procedure.

Looking Deeper: A Lower Bound Maximization View of the EM

- Starting from an arbitrary $\boldsymbol{\xi}^{(0)}$, the $(j+1)$ iteration comprises the following steps:
  - Step 1: Holding $\boldsymbol{\xi}^{(j)}$ fixed, optimize w.r. to q. This step tightens the lower bound in (15). This is achieved if $\mathrm{KL}(q \,\|\, p) = 0$ and it can only happen if

$$q^{(j+1)}(\mathcal{X}^l) = p(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}),$$

    that is, if we set $q(\mathcal{X}^l)$ equal to the posterior given $\mathcal{X}$ and $\boldsymbol{\xi}^{(j)}$; as (14) suggests, this makes the bound tight, i.e.,

$$\ln p(\mathcal{X}; \boldsymbol{\xi}^{(j)}) = \mathcal{F}\left(p(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}), \boldsymbol{\xi}^{(j)}\right).$$

  - Step 2: Fixing $q^{(j+1)}(\cdot)$, insert it in the place of q in (15), and since the bound holds for any $q(\cdot)$, maximize w.r. to $\boldsymbol{\xi}$, i.e.,

$$\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{F}\left(p\left(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}\right), \boldsymbol{\xi}\right).$$

Looking Deeper: A Lower Bound Maximization View of the EM

- Starting from an arbitrary $\boldsymbol{\xi}^{(0)}$, the $(j+1)$ iteration comprises the following steps:
  - Step 1: Holding $\boldsymbol{\xi}^{(j)}$ fixed, optimize w.r. to q. This step tightens the lower bound in (15). This is achieved if $\mathrm{KL}(q \,\|\, p) = 0$ and it can only happen if

$$q^{(j+1)}(\mathcal{X}^l) = p(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}),$$

    that is, if we set $q(\mathcal{X}^l)$ equal to the posterior given $\mathcal{X}$ and $\boldsymbol{\xi}^{(j)}$; as (14) suggests, this makes the bound tight, i.e.,

$$\ln p(\mathcal{X}; \boldsymbol{\xi}^{(j)}) = \mathcal{F}\left(p(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}), \boldsymbol{\xi}^{(j)}\right).$$

  - Step 2: Fixing $q^{(j+1)}(\cdot)$, insert it in the place of q in (15), and since the bound holds for any $q(\cdot)$, maximize w.r. to ξ, i.e.,

$$\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{F}\left(p\left(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}\right), \boldsymbol{\xi}\right).$$

Looking Deeper: A Lower Bound Maximization View of the EM

- Starting from an arbitrary $\boldsymbol{\xi}^{(0)}$, the $(j+1)$ iteration comprises the following steps:
  - Step 1: Holding $\boldsymbol{\xi}^{(j)}$ fixed, optimize w.r. to q. This step tightens the lower bound in (15). This is achieved if $\mathrm{KL}(q \,\|\, p) = 0$ and it can only happen if

$$q^{(j+1)}(\mathcal{X}^l) = p(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}),$$

    that is, if we set $q(\mathcal{X}^l)$ equal to the posterior given $\mathcal{X}$ and $\boldsymbol{\xi}^{(j)}$; as (14) suggests, this makes the bound tight, i.e.,

$$\ln p(\mathcal{X}; \boldsymbol{\xi}^{(j)}) = \mathcal{F}\left(p(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}), \boldsymbol{\xi}^{(j)}\right).$$

  - Step 2: Fixing $q^{(j+1)}(\cdot)$, insert it in the place of q in (15), and since the bound holds for any $q(\cdot)$, maximize w.r. to $\boldsymbol{\xi}$, i.e.,

$$\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{F}\left(p\left(\mathcal{X}^l | \mathcal{X}; \boldsymbol{\xi}^{(j)}\right), \boldsymbol{\xi}\right).$$

Looking Deeper: A Lower Bound Maximization View of the EM

- Thus, we have re-derived the EM algorithm. Indeed, recall that

$$\mathcal{F}(q, \boldsymbol{\xi}) := \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l; \text{ hence,}$$

$$\mathcal{F}\left(p(\mathcal{X}^l|\mathcal{X}; \boldsymbol{\xi}^{(j)}), \boldsymbol{\xi}\right) = \mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) - \text{constant (w.r. to } \boldsymbol{\xi}),$$

where $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\Big[\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})\Big]$ is the same used in the EM;

- This rederivation of the EM makes it clear that the quantity, which is maximized, is the log-likelihood, $\ln p(\mathcal{X}; \boldsymbol{\xi})$, and that its value is guaranteed **not to decrease** after each combined iteration step.

Looking Deeper: A Lower Bound Maximization View of the EM

- Thus, we have re-derived the EM algorithm. Indeed, recall that

$$\mathcal{F}(q,\boldsymbol{\xi}) := \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X},\mathcal{X}^l;\boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l; \text{ hence,}$$

$$\mathcal{F}\left(p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi}^{(j)}),\boldsymbol{\xi}\right) = \mathcal{Q}(\boldsymbol{\xi},\boldsymbol{\xi}^{(j)}) - \text{constant (w.r. to } \boldsymbol{\xi}),$$

where $\mathcal{Q}(\boldsymbol{\xi},\boldsymbol{\xi}^{(j)}) = \mathbb{E}\Big[\ln p(\mathcal{X},\mathcal{X}^l;\boldsymbol{\xi})\Big]$ is the same used in the EM;

- This rederivation of the EM makes it clear that the quantity, which is maximized, is the log-likelihood, $\ln p(\mathcal{X};\boldsymbol{\xi})$, and that its value is guaranteed **not to decrease** after each combined iteration step.

Looking Deeper: A Lower Bound Maximization View of the EM

- Thus, we have re-derived the EM algorithm. Indeed, recall that

$$\mathcal{F}(q, \boldsymbol{\xi}) := \int q(\mathcal{X}^l) \ln \frac{p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})}{q(\mathcal{X}^l)} d\mathcal{X}^l; \text{ hence,}$$

$$\mathcal{F}\left(p(\mathcal{X}^l|\mathcal{X}; \boldsymbol{\xi}^{(j)}), \boldsymbol{\xi}\right) = \mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) - \text{constant (w.r. to } \boldsymbol{\xi}),$$

  where $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\Big[\ln p(\mathcal{X}, \mathcal{X}^l; \boldsymbol{\xi})\Big]$ is the same used in the EM;

- This rederivation of the EM makes it clear that the quantity, which is maximized, is the log-likelihood, $\ln p(\mathcal{X}; \boldsymbol{\xi})$, and that its value is guaranteed **not to decrease** after each combined iteration step.

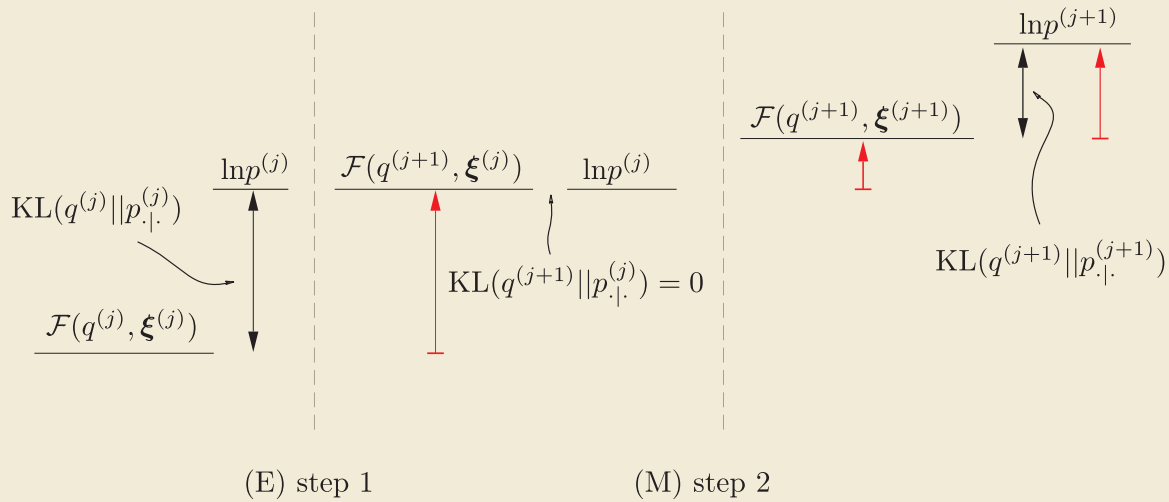
- Thus, we have re-derived the EM algorithm. Indeed, recall that

$$\mathcal{F}(q,\boldsymbol{\xi}) := \int q(\mathcal{X}^l)\ln\frac{p(\mathcal{X},\mathcal{X}^l;\boldsymbol{\xi})}{q(\mathcal{X}^l)}d\mathcal{X}^l; \text{ hence,}$$

$$\mathcal{F}\left(p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi}^{(j)}),\boldsymbol{\xi}\right) = \mathcal{Q}(\boldsymbol{\xi},\boldsymbol{\xi}^{(j)}) - \text{constant (w.r. to } \boldsymbol{\xi}),$$

where $\mathcal{Q}(\boldsymbol{\xi},\boldsymbol{\xi}^{(j)}) = \mathbb{E}\Big[\ln p(\mathcal{X},\mathcal{X}^l;\boldsymbol{\xi})\Big]$ is the same used in the EM;

- This rederivation of the EM makes it clear that the quantity, which is maximized, is the log-likelihood, $\ln p(\mathcal{X};\boldsymbol{\xi})$, and that its value is guaranteed **not to decrease** after each combined iteration step.



The E-step adjusts $q^{(j)} := q^{(j)}(\mathcal{X}^l)$ so that its KL distance from $p^{(j)} := p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi}^{(j)})$ becomes zero.
The M-step maximizes with respect to $\boldsymbol{\xi}$.

Exponential Family of Probability Distributions

- Maximizing w.r. to a function is facilitated by constraining the function to lie within a parametric family of functions.
- The exponential family of distributions is of particular importance. Let $\mathbf{x} \in \mathbb{R}^l$ be a random vector and $\theta \in \mathbb{R}^K$ a (random) parameter vector. We say that the parameterized pdf $p(\boldsymbol{x}|\boldsymbol{\theta})$ is of the exponential form if

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x})\right), \tag{16}$$

  where

$$g(\boldsymbol{\theta}) = \frac{1}{\int f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x})\right) d\boldsymbol{x}},$$

  is the normalizing constant of the pdf.
- The vector $\boldsymbol{\phi}(\boldsymbol{\theta})$ comprises the set of the so-called natural parameters. The function $\boldsymbol{u}(\boldsymbol{x})$ is a sufficient statistic for the parameter $\boldsymbol{\theta}$. If $\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{\theta}$, then the exponential family is said to be in canonical form.

Exponential Family of Probability Distributions

- Maximizing w.r. to a function is facilitated by constraining the function to lie within a parametric family of functions.
- The exponential family of distributions is of particular importance. Let $\mathbf{x} \in \mathbb{R}^l$ be a random vector and $\boldsymbol{\theta} \in \mathbb{R}^K$ a (random) parameter vector. We say that the parameterized pdf $p(\boldsymbol{x}|\boldsymbol{\theta})$ is of the exponential form if

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x})\right), \tag{16}$$

  where

$$g(\boldsymbol{\theta}) = \frac{1}{\int f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x})\right) d\boldsymbol{x}},$$

  is the normalizing constant of the pdf.
- The vector $\phi(\theta)$ comprises the set of the so-called natural parameters. The function $u(x)$ is a sufficient statistic for the parameter θ. If $\phi(\theta) = \theta$, then the exponential family is said to be in canonical form.

Exponential Family of Probability Distributions

- Maximizing w.r. to a function is facilitated by constraining the function to lie within a parametric family of functions.
- The exponential family of distributions is of particular importance. Let $\mathbf{x} \in \mathbb{R}^l$ be a random vector and $\boldsymbol{\theta} \in \mathbb{R}^K$ a (random) parameter vector. We say that the parameterized pdf $p(\boldsymbol{x}|\boldsymbol{\theta})$ is of the exponential form if

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x})\right), \tag{16}$$

  where

$$g(\boldsymbol{\theta}) = \frac{1}{\int f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x})\right) d\boldsymbol{x}},$$

  is the normalizing constant of the pdf.
- The vector $\boldsymbol{\phi}(\boldsymbol{\theta})$ comprises the set of the so-called natural parameters. The function $\boldsymbol{u}(\boldsymbol{x})$ is a sufficient statistic for the parameter $\boldsymbol{\theta}$. If $\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{\theta}$, then the exponential family is said to be in canonical form.

Exponential Family of Probability Distributions

- An advantage of the exponential family is that one can find conjugate priors for θ; that is, priors which lead to posteriors, $p(\boldsymbol{\theta}|\mathcal{X})$, of the **same** functional form as $p(\boldsymbol{\theta})$.
- If the conditional (likelihood) pdf is of the exponential form, i.e.,

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x})\right),$$

its conjugate prior is defined as,

$$p(\boldsymbol{\theta}; \lambda, \boldsymbol{v}) = h(\lambda, \boldsymbol{v}) \big(g(\boldsymbol{\theta})\big)^{\lambda} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{v}\right), \tag{17}$$

where $\lambda > 0$ and $\boldsymbol{v}$ are known as hyperparameters; that is, parameters that control other parameters. The factor $h(\lambda, \boldsymbol{v})$ is an appropriate normalizing constant.

- It is easy to see that defining the prior as in (17) and the likelihood function as above, the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ is of the same form as in (17).

Exponential Family of Probability Distributions

- An advantage of the exponential family is that one can find conjugate priors for θ; that is, priors which lead to posteriors, $p(\boldsymbol{\theta}|\mathcal{X})$, of the **same** functional form as $p(\boldsymbol{\theta})$.
- If the conditional (likelihood) pdf is of the exponential form, i.e.,

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\boldsymbol{u}(\boldsymbol{x})\right),$$

its conjugate prior is defined as,

$$p(\boldsymbol{\theta}; \lambda, \boldsymbol{v}) = h(\lambda, \boldsymbol{v})\big(g(\boldsymbol{\theta})\big)^{\lambda} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\boldsymbol{v}\right), \tag{17}$$

where $\lambda > 0$ and $\boldsymbol{v}$ are known as hyperparameters; that is, parameters that control other parameters. The factor $h(\lambda, \boldsymbol{v})$ is an appropriate normalizing constant.

- It is easy to see that defining the prior as in (17) and the likelihood function as above, the posterior $p(\theta|x)$ is of the same form as in (17).

Exponential Family of Probability Distributions

- An advantage of the exponential family is that one can find conjugate priors for θ; that is, priors which lead to posteriors, $p(\boldsymbol{\theta}|\mathcal{X})$, of the **same** functional form as $p(\boldsymbol{\theta})$.
- If the conditional (likelihood) pdf is of the exponential form, i.e.,

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\boldsymbol{u}(\boldsymbol{x})\right),$$

its conjugate prior is defined as,

$$p(\boldsymbol{\theta}; \lambda, \boldsymbol{v}) = h(\lambda, \boldsymbol{v})\big(g(\boldsymbol{\theta})\big)^{\lambda} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\boldsymbol{v}\right), \tag{17}$$

where $\lambda > 0$ and $\boldsymbol{v}$ are known as hyperparameters; that is, parameters that control other parameters. The factor $h(\lambda, \boldsymbol{v})$ is an appropriate normalizing constant.

- It is easy to see that defining the prior as in (17) and the likelihood function as above, the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ is of the same form as in (17).

Exponential Family of Probability Distributions

- Indeed, taking into account that $p(\boldsymbol{\theta}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, we obtain

$$p(\boldsymbol{\theta}|\boldsymbol{x}) \propto (g(\boldsymbol{\theta}))^{\lambda} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\Big(\boldsymbol{v} + \boldsymbol{u}(\boldsymbol{x})\Big)\right).$$

- Assume, now, that $\mathbf{x}$ and θ obey (16)-(17) and let $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be a set of i.i.d. observations. Then, we obtain

$$p(\mathcal{X}|\boldsymbol{\theta}) = (g(\boldsymbol{\theta}))^N \prod_{n=1}^{N} f(\boldsymbol{x}_n) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \sum_{i=1}^{N} \boldsymbol{u}(\boldsymbol{x}_i)\right), \quad (18)$$

$$p(\boldsymbol{\theta}|\mathcal{X}) \propto (g(\boldsymbol{\theta}))^{\lambda+N} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\Big(\boldsymbol{v} + \sum_{n=1}^{N} \boldsymbol{u}(\boldsymbol{x}_n)\Big)\right). \quad (19)$$

Exponential Family of Probability Distributions

- Indeed, taking into account that $p(\boldsymbol{\theta}|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, we obtain

$$p(\boldsymbol{\theta}|\boldsymbol{x}) \propto (g(\boldsymbol{\theta}))^{\lambda} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\Big(\boldsymbol{v} + \boldsymbol{u}(\boldsymbol{x})\Big)\right).$$

- Assume, now, that $\mathbf{x}$ and θ obey (16)-(17) and let $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be a set of i.i.d. observations. Then, we obtain

$$p(\mathcal{X}|\boldsymbol{\theta}) = (g(\boldsymbol{\theta}))^N \prod_{n=1}^{N} f(\boldsymbol{x}_n) \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \sum_{i=1}^{N} \boldsymbol{u}(\boldsymbol{x}_i)\right), \quad (18)$$

$$p(\boldsymbol{\theta}|\mathcal{X}) \propto (g(\boldsymbol{\theta}))^{\lambda+N} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\Big(\boldsymbol{v} + \sum_{n=1}^{N} \boldsymbol{u}(\boldsymbol{x}_n)\Big)\right). \quad (19)$$

Exponential Family of Probability Distributions

- In other words, the posterior has hyperparameters equal to

$$\tilde{\lambda} = \lambda + N, \quad \tilde{\boldsymbol{v}} = \boldsymbol{v} + \sum_{n=1}^{N} \boldsymbol{u}(\boldsymbol{x}_n).$$

- Interpreting the above, one can view λ as being the effective number of observations that, **implicitly**, the prior information contributes to the Bayesian learning process and v is the total amount of information that these (implicit) λ observations contribute to the sufficient statistic. Basically, their exact values quantify the amount of prior knowledge that the designer wants to **embed** into the learning task.

Exponential Family of Probability Distributions

- In other words, the posterior has hyperparameters equal to

$$\tilde{\lambda} = \lambda + N, \quad \tilde{\boldsymbol{v}} = \boldsymbol{v} + \sum_{n=1}^{N} \boldsymbol{u}(\boldsymbol{x}_n).$$

- Interpreting the above, one can view λ as being the effective number of observations that, **implicitly**, the prior information contributes to the Bayesian learning process and $\boldsymbol{v}$ is the total amount of information that these (implicit) λ observations contribute to the sufficient statistic. Basically, their exact values quantify the amount of prior knowledge that the designer wants to **embed** into the learning task.

- The Gaussian-gamma pair: Let our random variable, x, be a scalar and assume that,

$$p(x|\sigma^2) = \mathcal{N}(x|\mu, \sigma^2),$$

where μ is known and σ^2 is an unknown parameter. We will show that:

  - I. $p(x|\sigma^2)$ belongs to the exponential family.
    It is algebraically more convenient to work with the precision $\beta = \frac{1}{\sigma^2}$. Hence,

$$p(x|\beta) = \frac{\beta^{1/2}}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\beta(x-\mu)^2\right).$$

    Thus, $p(x|\beta)$ belongs to the exponential family with

$$f(x) = \frac{1}{\sqrt{2\pi}}, \ \phi(\beta) = -\beta, \ u(x) = \frac{1}{2}(x-\mu)^2,$$

    and

$$g(\beta) = \frac{1}{\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\beta(x-\mu)^2\right) dx} = \beta^{1/2}.$$

- The Gaussian-gamma pair: Let our random variable, x, be a scalar and assume that,

$$p(x|\sigma^2) = \mathcal{N}(x|\mu, \sigma^2),$$

where μ is known and σ^2 is an unknown parameter. We will show that:

  - I. $p(x|\sigma^2)$ belongs to the exponential family.
    It is algebraically more convenient to work with the precision $\beta = \frac{1}{\sigma^2}$. Hence,

$$p(x|\beta) = \frac{\beta^{1/2}}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\beta(x-\mu)^2\right).$$

Thus, $p(x|\beta)$ belongs to the exponential family with

$$f(x) = \frac{1}{\sqrt{2\pi}}, \ \phi(\beta) = -\beta, \ u(x) = \frac{1}{2}(x-\mu)^2,$$

and

$$g(\beta) = \frac{1}{\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\beta(x-\mu)^2\right) dx} = \beta^{1/2}.$$

- The Gaussian-gamma pair: Let our random variable, x, be a scalar and assume that,

$$p(x|\sigma^2) = \mathcal{N}(x|\mu, \sigma^2),$$

where μ is known and σ^2 is an unknown parameter. We will show that:

  - I. $p(x|\sigma^2)$ belongs to the exponential family.
    It is algebraically more convenient to work with the precision $\beta = \frac{1}{\sigma^2}$. Hence,

$$p(x|\beta) = \frac{\beta^{1/2}}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\beta(x-\mu)^2\right).$$

    Thus, $p(x|\beta)$ belongs to the exponential family with

$$f(x) = \frac{1}{\sqrt{2\pi}},\ \phi(\beta) = -\beta,\ u(x) = \frac{1}{2}(x-\mu)^2,$$

    and

$$g(\beta) = \frac{1}{\int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\beta(x-\mu)^2\right) dx} = \beta^{1/2}.$$

Exponential Family of Probability Distributions

- (Continued)
  - II. The conjugate prior of $\mathcal{N}(x|\mu, \sigma^2)$, for known μ and unknown σ^2, follows the gamma distribution.

    From the corresponding definition in (17), we have,

    $$p(\beta; \lambda, v) = h(\lambda, v)\beta^{\frac{\lambda}{2}} \exp(-\beta v).$$

    This has the form of

    $$\text{Gamma}(\beta; a, b) = \frac{1}{\Gamma(a)} b^a \beta^{a-1} \exp(-b\beta),$$

    with parameters $a = \frac{\lambda}{2} + 1$ and $b = v$. The normalizing constant, $h(\lambda, v)$, is necessarily equal to $b^a/\Gamma(a)$.

Exponential Family of Probability Distributions

- (Continued)
  - II. The conjugate prior of $\mathcal{N}(x|\mu, \sigma^2)$, for known μ and unknown σ^2, follows the gamma distribution.

    From the corresponding definition in (17), we have,

    $$p(\beta; \lambda, v) = h(\lambda, v)\beta^{\frac{\lambda}{2}} \exp(-\beta v).$$

    This has the form of

    $$\text{Gamma}(\beta; a, b) = \frac{1}{\Gamma(a)} b^a \beta^{a-1} \exp(-b\beta),$$

    with parameters $a = \frac{\lambda}{2} + 1$ and $b = v$. The normalizing constant, $h(\lambda, v)$, is necessarily equal to $b^a/\Gamma(a)$.

Exponential Family of Probability Distributions

- If we are given multiple observations $x_n,\ n = 1, 2, \ldots, N$, then the resulting posterior according to (19) will be a gamma distribution with

$$\tilde{b} = b + \frac{1}{2}\sum_{n=1}^{N}(x_n - \mu)^2 = b + \frac{N}{2}\hat{\sigma}^2_{ML},$$

  where $\hat{\sigma}^2_{ML}$ denotes the maximum likelihood estimate of the variance.

- Hence, the physical meaning of b is that it quantifies our prior guess about the unknown variance. It can easily be shown that the conjugate prior w.r. to μ, if σ^2 is known, is a Gaussian.

Exponential Family of Probability Distributions

- If we are given multiple observations $x_n,\ n = 1, 2, \ldots, N$, then the resulting posterior according to (19) will be a gamma distribution with

$$\tilde{b} = b + \frac{1}{2}\sum_{n=1}^{N}(x_n - \mu)^2 = b + \frac{N}{2}\hat{\sigma}^2_{ML},$$

  where $\hat{\sigma}^2_{ML}$ denotes the maximum likelihood estimate of the variance.

- Hence, the physical meaning of b is that it quantifies our prior guess about the unknown variance. It can easily be shown that the conjugate prior w.r. to μ, if σ^2 is known, is a Gaussian.

Exponential Family of Probability Distributions

- In case of a multivariate Gaussian of known mean $\boldsymbol{\mu}$ and unknown covariance matrix Σ (precision matrix $Q = \Sigma^{-1}$), it can also be shown that it is of the exponential form and its conjugate prior is given by the Wishart distribution (multivariate analogue of the gamma distribution),

$$\mathcal{W}(Q|W,\nu) = h|Q|^{\frac{\nu-l-1}{2}} \exp\left(-\frac{1}{2}\text{trace}\left\{W^{-1}Q\right\}\right),$$

where h is the normalizing constant and W is an $l \times l$ matrix.

The normalizing constant is given by,

$$h = |W|^{-\frac{\nu}{2}} \left(2^{\frac{\nu l}{2}} \pi^{\frac{l(l-1)}{4}} \prod_{i=1}^{l} \Gamma\left(\frac{\nu+1-i}{2}\right)\right)^{-1}.$$

.

Exponential Family of Probability Distributions

- In case of a multivariate Gaussian of known mean $\boldsymbol{\mu}$ and unknown covariance matrix Σ (precision matrix $Q = \Sigma^{-1}$), it can also be shown that it is of the exponential form and its conjugate prior is given by the Wishart distribution (multivariate analogue of the gamma distribution),

$$\mathcal{W}(Q|W,\nu) = h|Q|^{\frac{\nu-l-1}{2}} \exp\left(-\frac{1}{2}\text{trace}\left\{W^{-1}Q\right\}\right),$$

where h is the normalizing constant and W is an $l \times l$ matrix. The normalizing constant is given by,

$$h = |W|^{-\frac{\nu}{2}} \left(2^{\frac{\nu l}{2}} \pi^{\frac{l(l-1)}{4}} \prod_{i=1}^{l} \Gamma\left(\frac{\nu+1-i}{2}\right)\right)^{-1}.$$

.

Variational Approximation in Bayesian Learning

- Recall that in order to apply the EM algorithm, the functional form of the posterior of the latent variables, given the observations, $p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})$, must be known.
- Furthermore, the analytic computation of the posterior is not always tractable. In such cases, the EM algorithm, in its standard form, is not applicable.
- We are going to describe an alternative path, that builds upon the EM interpretation, based upon the lower bound interpretation.

Variational Approximation in Bayesian Learning

- Recall that in order to apply the EM algorithm, the functional form of the posterior of the latent variables, given the observations, $p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})$, must be known.
- Furthermore, the analytic computation of the posterior is not always tractable. In such cases, the EM algorithm, in its standard form, is not applicable.
- We are going to describe an alternative path, that builds upon the EM interpretation, based upon the lower bound interpretation.

Variational Approximation in Bayesian Learning

- Recall that in order to apply the EM algorithm, the functional form of the posterior of the latent variables, given the observations, $p(\mathcal{X}^l|\mathcal{X};\boldsymbol{\xi})$, must be known.
- Furthermore, the analytic computation of the posterior is not always tractable. In such cases, the EM algorithm, in its standard form, is not applicable.
- We are going to describe an alternative path, that builds upon the EM interpretation, based upon the lower bound interpretation.

Variational Approximation in Bayesian Learning

- Let $\mathcal{X}$ be the set of observed variables and $\mathcal{X}^l$ the respective set of the latent ones. Furthermore, we will explicitly bring into the game the set of parameters, $\boldsymbol{\theta} \in \mathbb{R}^K$, which are treated as random variables in the Bayesian context, accompanied by a prior pdf.
- Note that we reserve the term "latent" for hidden variables whose number depends on the number of observations, N. In contrast, a random parameter vector, $\boldsymbol{\theta}$, although a hidden random vector, it has a fixed dimension.
- The functional in (12) is now redefined as,

$$\mathcal{F}(q, \boldsymbol{\xi}) = \int q(\mathcal{X}^l, \boldsymbol{\theta}) \ln \frac{p(\mathcal{X}, \mathcal{X}^l, \boldsymbol{\theta}; \boldsymbol{\xi})}{q(\mathcal{X}^l, \boldsymbol{\theta})} d\mathcal{X}^l d\boldsymbol{\theta}, \tag{20}$$

where $\boldsymbol{\xi}$ is the set of deterministic (hyper)parameters.

Variational Approximation in Bayesian Learning

- Let $\mathcal{X}$ be the set of observed variables and $\mathcal{X}^l$ the respective set of the latent ones. Furthermore, we will explicitly bring into the game the set of parameters, $\boldsymbol{\theta} \in \mathbb{R}^K$, which are treated as random variables in the Bayesian context, accompanied by a prior pdf.
- Note that we reserve the term "latent" for hidden variables whose number depends on the number of observations, N. In contrast, a random parameter vector, $\boldsymbol{\theta}$, although a hidden random vector, it has a fixed dimension.
- The functional in (12) is now redefined as,

$$\mathcal{F}(q, \boldsymbol{\xi}) = \int q(\mathcal{X}^l, \boldsymbol{\theta}) \ln \frac{p(\mathcal{X}, \mathcal{X}^l, \boldsymbol{\theta}; \boldsymbol{\xi})}{q(\mathcal{X}^l, \boldsymbol{\theta})} d\mathcal{X}^l d\boldsymbol{\theta}, \tag{20}$$

where $\boldsymbol{\xi}$ is the set of deterministic (hyper)parameters.

Variational Approximation in Bayesian Learning

- Let $\mathcal{X}$ be the set of observed variables and $\mathcal{X}^l$ the respective set of the latent ones. Furthermore, we will explicitly bring into the game the set of parameters, $\boldsymbol{\theta} \in \mathbb{R}^K$, which are treated as random variables in the Bayesian context, accompanied by a prior pdf.
- Note that we reserve the term "latent" for hidden variables whose number depends on the number of observations, N. In contrast, a random parameter vector, $\boldsymbol{\theta}$, although a hidden random vector, it has a fixed dimension.
- The functional in (12) is now redefined as,

$$\mathcal{F}(q, \boldsymbol{\xi}) = \int q(\mathcal{X}^l, \boldsymbol{\theta}) \ln \frac{p(\mathcal{X}, \mathcal{X}^l, \boldsymbol{\theta}; \boldsymbol{\xi})}{q(\mathcal{X}^l, \boldsymbol{\theta})} d\mathcal{X}^l d\boldsymbol{\theta}, \tag{20}$$

where $\boldsymbol{\xi}$ is the set of deterministic (hyper)parameters.

Variational Approximation in Bayesian Learning

- Then the counterpart of (13) becomes (suppressing the notational dependence on $\boldsymbol{\xi}$)

$$\mathcal{F}(q) = \ln p(\mathcal{X}) + \int q(\mathcal{X}^l, \boldsymbol{\theta}) \ln \frac{p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})}{q(\mathcal{X}^l, \boldsymbol{\theta})} d\mathcal{X}^l d\boldsymbol{\theta}. \tag{21}$$

- The difference with (13) lies in the fact that $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ is **not** known; so maximizing the above w.r. to q by setting to zero the KL divergence, $\mathrm{KL}(q||p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X}))$, is no more possible.

- In order to deal with the current problem, we will constrain $q(\mathcal{X}^l, \boldsymbol{\theta})$ to lie within a family of functions. Note that in this case, if the unknown $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ does not belong to this specific family of functions, the KL divergence cannot become zero and the lower bound, $\mathcal{F}(q)$, of the marginal log likelihood cannot be made tight. This is the reason that the method is known as variational **approximation**.

Variational Approximation in Bayesian Learning

- Then the counterpart of (13) becomes (suppressing the notational dependence on $\boldsymbol{\xi}$)

$$\mathcal{F}(q) = \ln p(\mathcal{X}) + \int q(\mathcal{X}^l, \boldsymbol{\theta}) \ln \frac{p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})}{q(\mathcal{X}^l, \boldsymbol{\theta})} d\mathcal{X}^l d\boldsymbol{\theta}. \tag{21}$$

- The difference with (13) lies in the fact that $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ is **not** known; so maximizing the above w.r. to q by setting to zero the KL divergence, $\mathrm{KL}(q||p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X}))$, is no more possible.

- In order to deal with the current problem, we will constrain $q(\mathcal{X}^l, \boldsymbol{\theta})$ to lie within a family of functions. Note that in this case, if the unknown $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ does not belong to this specific family of functions, the KL divergence cannot become zero and the lower bound, $\mathcal{F}(q)$, of the marginal log likelihood cannot be made tight. This is the reason that the method is known as variational **approximation**.

Variational Approximation in Bayesian Learning

- Then the counterpart of (13) becomes (suppressing the notational dependence on $\boldsymbol{\xi}$)

$$\mathcal{F}(q) = \ln p(\mathcal{X}) + \int q(\mathcal{X}^l, \boldsymbol{\theta}) \ln \frac{p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})}{q(\mathcal{X}^l, \boldsymbol{\theta})} d\mathcal{X}^l d\boldsymbol{\theta}. \tag{21}$$

- The difference with (13) lies in the fact that $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ is **not** known; so maximizing the above w.r. to q by setting to zero the KL divergence, $\mathrm{KL}(q||p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X}))$, is no more possible.

- In order to deal with the current problem, we will constrain $q(\mathcal{X}^l, \boldsymbol{\theta})$ to lie within a family of functions. Note that in this case, if the unknown $p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X})$ does not belong to this specific family of functions, the KL divergence cannot become zero and the lower bound, $\mathcal{F}(q)$, of the marginal log likelihood cannot be made tight. This is the reason that the method is known as variational **approximation**.

The Mean Field Approximation

- This type of approximation results by constraining $q(\mathcal{X}^l, \boldsymbol{\theta})$ to be **factorized**, i.e.,

$$q(\mathcal{X}^l, \boldsymbol{\theta}) = q_{\mathcal{X}^l}(\mathcal{X}^l) q_\theta(\boldsymbol{\theta}). \tag{22}$$

This factorization can be, and usually it is, extended to

$$q(\mathcal{X}^l, \boldsymbol{\theta}) = q_{\boldsymbol{x}_1^l}(\boldsymbol{x}_1^l) \dots q_{\boldsymbol{x}_N^l}(\boldsymbol{x}_N^l) q_\theta(\boldsymbol{\theta}) \tag{23}$$

To simplify our notation, without sacrificing generality, we will work with (22). This type of approximation has been used in statistical physics and it is known as mean field approximation.

The Mean Field Approximation

- This type of approximation results by constraining $q(\mathcal{X}^l, \boldsymbol{\theta})$ to be **factorized**, i.e.,

$$q(\mathcal{X}^l, \boldsymbol{\theta}) = q_{\mathcal{X}^l}(\mathcal{X}^l) q_\theta(\boldsymbol{\theta}). \tag{22}$$

This factorization can be, and usually it is, extended to

$$q(\mathcal{X}^l, \boldsymbol{\theta}) = q_{\boldsymbol{x}_1^l}(\boldsymbol{x}_1^l) \dots q_{\boldsymbol{x}_N^l}(\boldsymbol{x}_N^l) q_\theta(\boldsymbol{\theta}) \tag{23}$$

To simplify our notation, without sacrificing generality, we will work with (22). This type of approximation has been used in statistical physics and it is known as mean field approximation.

The Mean Field Approximation

- This type of approximation results by constraining $q(\mathcal{X}^l, \boldsymbol{\theta})$ to be **factorized**, i.e.,

$$q(\mathcal{X}^l, \boldsymbol{\theta}) = q_{\mathcal{X}^l}(\mathcal{X}^l) q_\theta(\boldsymbol{\theta}). \tag{22}$$

This factorization can be, and usually it is, extended to

$$q(\mathcal{X}^l, \boldsymbol{\theta}) = q_{\boldsymbol{x}_1^l}(\boldsymbol{x}_1^l) \dots q_{\boldsymbol{x}_N^l}(\boldsymbol{x}_N^l) q_\theta(\boldsymbol{\theta}) \tag{23}$$

To simplify our notation, without sacrificing generality, we will work with (22). This type of approximation has been used in statistical physics and it is known as mean field approximation.

The Mean Field Approximation

- Maximization of $\mathcal{F}$ w.r. to $q(\mathcal{X}^l, \boldsymbol{\theta})$ (as it is required by the E-step of the EM algorithm) will take place by splitting the process so that to maximize first w.r. to $q_{\mathcal{X}^l}$ and then w.r. to q_θ.
- Bringing back into the scene the (deterministic) parameter vector, ξ, and initializing the algorithm from arbitrary values for $\xi^{(0)}$ as well as for the involved statistics related to q_θ (this will become clear while dealing with the examples), the $(j+1)$ iteration comprises the following steps:

The Mean Field Approximation

- Maximization of $\mathcal{F}$ w.r. to $q(\mathcal{X}^l, \boldsymbol{\theta})$ (as it is required by the E-step of the EM algorithm) will take place by splitting the process so that to maximize first w.r. to $q_{\mathcal{X}^l}$ and then w.r. to q_θ.
- Bringing back into the scene the (deterministic) parameter vector, $\boldsymbol{\xi}$, and initializing the algorithm from arbitrary values for $\boldsymbol{\xi}^{(0)}$ as well as for the involved statistics related to q_θ (this will become clear while dealing with the examples), the $(j+1)$ iteration comprises the following steps:

The Mean Field Approximation

- E-Step 1a: Holding $\boldsymbol{\xi}^{(j)}$ and $q_\theta^{(j)}$ fixed, maximize $\mathcal{F}(q_{\mathcal{X}^l}, q_\theta^{(j)}; \boldsymbol{\xi}^{(j)})$ w.r. to $q_{\mathcal{X}^l}$. This leads to:

  $q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l)$: To this end, perform an expectation w.r. to $q_\theta^{(j)}$.

- E-Step 1b: Freezing $\boldsymbol{\xi}^{(j)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize $\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta; \boldsymbol{\xi}^{(j)})$ with respect to q_θ. This leads to:

  $q_\theta^{(j+1)}(\boldsymbol{\theta})$: To this end, perform an expectation w.r. to $q_{\mathcal{X}^l}^{(j+1)}$.

  Steps 1a and 1b comprise the E-step of the variational Bayesian EM.

- M-Step 2: Freezing $q_\theta^{(j+1)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize $\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j+1)}; \boldsymbol{\xi})$ w.r. to $\boldsymbol{\xi}$.

The Mean Field Approximation

- E-Step 1a: Holding $\boldsymbol{\xi}^{(j)}$ and $q_\theta^{(j)}$ fixed, maximize $\mathcal{F}(q_{\mathcal{X}^l}, q_\theta^{(j)}; \boldsymbol{\xi}^{(j)})$ w.r. to $q_{\mathcal{X}^l}$. This leads to:

  $q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l)$: To this end, perform an expectation w.r. to $q_\theta^{(j)}$.

- E-Step 1b: Freezing $\boldsymbol{\xi}^{(j)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize $\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta; \boldsymbol{\xi}^{(j)})$ with respect to q_θ. This leads to:

  $q_\theta^{(j+1)}(\boldsymbol{\theta})$: To this end, perform an expectation w.r. to $q_{\mathcal{X}^l}^{(j+1)}$.

  Steps 1a and 1b comprise the E-step of the variational Bayesian EM.

- M-Step 2: Freezing $q_\theta^{(j+1)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize $\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j+1)}; \boldsymbol{\xi})$ w.r. to $\boldsymbol{\xi}$.

The Mean Field Approximation

- E-Step 1a: Holding $\boldsymbol{\xi}^{(j)}$ and $q_\theta^{(j)}$ fixed, maximize $\mathcal{F}(q_{\mathcal{X}^l}, q_\theta^{(j)}; \boldsymbol{\xi}^{(j)})$ w.r. to $q_{\mathcal{X}^l}$. This leads to:

  $q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l)$: To this end, perform an expectation w.r. to $q_\theta^{(j)}$.

- E-Step 1b: Freezing $\boldsymbol{\xi}^{(j)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize $\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta; \boldsymbol{\xi}^{(j)})$ with respect to q_θ. This leads to:

  $q_\theta^{(j+1)}(\boldsymbol{\theta})$: To this end, perform an expectation w.r. to $q_{\mathcal{X}^l}^{(j+1)}$.

  Steps 1a and 1b comprise the E-step of the variational Bayesian EM.

- M-Step 2: Freezing $q_\theta^{(j+1)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize $\mathcal{F}(q_{\mathcal{X}^l}^{(j+1)}, q_\theta^{(j+1)}; \boldsymbol{\xi})$ w.r. to $\boldsymbol{\xi}$.

The Mean Field Approximation

In a more explicit form:

- E-Step 1a: Holding $\boldsymbol{\xi}^{(j)}$ and $q_{\theta}^{(j)}$ fixed, maximizing the bound w.r. to $q_{\mathcal{X}^l}$, leads to:

$$q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l) = \frac{\exp\left(\mathbb{E}_{q_{\theta}^{(j)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]\right)}{\int \exp\left(\mathbb{E}_{q_{\theta}^{(j)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]\right) d\mathcal{X}^l}. \tag{24}$$

- E-Step 1b: Freezing $\boldsymbol{\xi}^{(j)}$ and $q_{\mathcal{X}^l}^{(j+1)}$ and maximizing with respect to q_{θ}, we obtain,

$$q_{\theta}^{(j+1)}(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \exp\left(\mathbb{E}_{q_{\mathcal{X}^l}^{(j+1)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]\right)}{\int p(\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \exp\left(\mathbb{E}_{q_{\mathcal{X}^l}^{(j+1)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]\right) d\boldsymbol{\theta}}. \tag{25}$$

Steps 1a and 1b comprise the E-step of the variational Bayesian EM.

The Mean Field Approximation

In a more explicit form:

- E-Step 1a: Holding $\boldsymbol{\xi}^{(j)}$ and $q_{\theta}^{(j)}$ fixed, maximizing the bound w.r. to $q_{\mathcal{X}^l}$, leads to:

$$q_{\mathcal{X}^l}^{(j+1)}(\mathcal{X}^l) = \frac{\exp\left(\mathbb{E}_{q_{\theta}^{(j)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]\right)}{\int \exp\left(\mathbb{E}_{q_{\theta}^{(j)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]\right) d\mathcal{X}^l}. \tag{24}$$

- E-Step 1b: Freezing $\boldsymbol{\xi}^{(j)}$ and $q_{\mathcal{X}^l}^{(j+1)}$ and maximizing with respect to q_{θ}, we obtain,

$$q_{\theta}^{(j+1)}(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \exp\left(\mathbb{E}_{q_{\mathcal{X}^l}^{(j+1)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]\right)}{\int p(\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)}) \exp\left(\mathbb{E}_{q_{\mathcal{X}^l}^{(j+1)}}\left[\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\right]\right) d\boldsymbol{\theta}}. \tag{25}$$

Steps 1a and 1b comprise the E-step of the variational Bayesian EM.

The Mean Field Approximation

- M-Step 2: Freezing $q_\theta^{(j+1)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize the lower bound w.r. to $\boldsymbol{\xi}$, i.e.,

$$\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{F}(q_\theta^{(j+1)}, q_{\mathcal{X}^l}^{(j+1)}; \boldsymbol{\xi}).$$

- The concept behind the mean field approximation in the Bayesian variational approach is illustrated in the figure below. There are two observations to be made. Step 1 is now split into two parts and more important, the KL divergence does not (in general) go to zero; hence, the bound does **not** become **tight**.

- M-Step 2: Freezing $q_\theta^{(j+1)}$ and $q_{\mathcal{X}^l}^{(j+1)}$, maximize the lower bound w.r.t. $\boldsymbol{\xi}$, i.e.,

$$\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{F}(q_\theta^{(j+1)}, q_{\mathcal{X}^l}^{(j+1)}; \boldsymbol{\xi}).$$

- The concept behind the mean field approximation in the Bayesian variational approach is illustrated in the figure below. There are two observations to be made. Step 1 is now split into two parts and more important, the KL divergence does not (in general) go to zero; hence, the bound does **not** become **tight**.

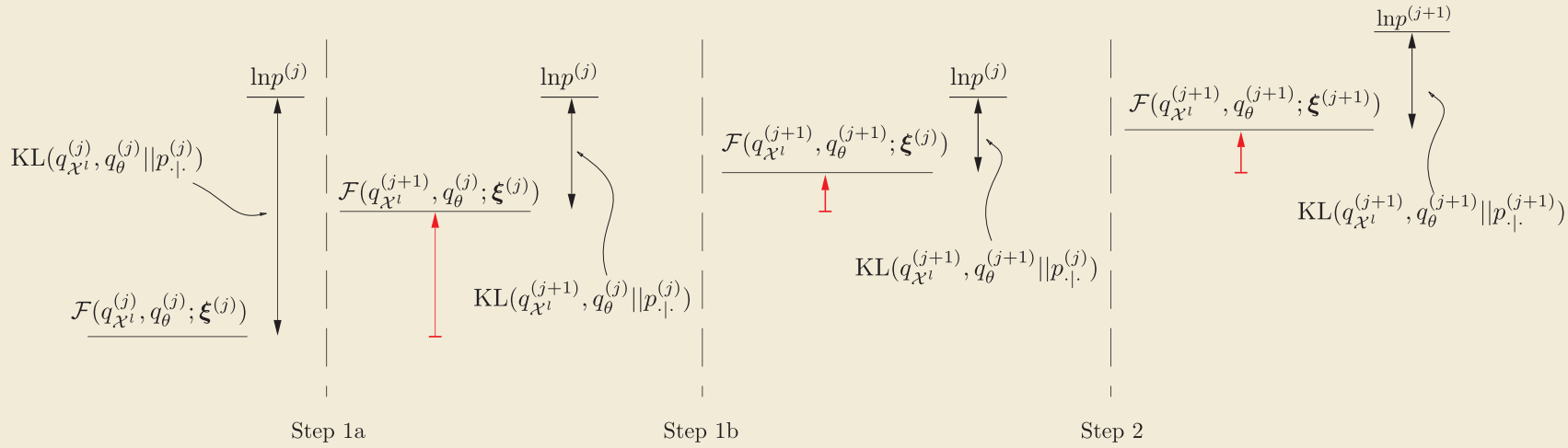


Illustration of the stepwise increase of $\ln p^{(j)}$ at the $(j+1)$ iteration of the Variational Bayesian EM algorithm. Observe that $\ln p^{(j+1)} > \ln p^{(j)}$, where we have used the notation, $p^{(j)} = p(\mathcal{X}, \boldsymbol{\xi}^{(j)})$ and $p_{\cdot|\cdot}^{(j)} := p(\mathcal{X}^l, \boldsymbol{\theta}|\mathcal{X}; \boldsymbol{\xi}^{(j)})$.

The Case of the Exponential Family of Probability Distributions

- Looking carefully at (24) and (25), it becomes clear that the practical application of the variational Bayesian EM depends on the computational tractability of the expected values of the $\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi})$.
- We will restrict the involved distributions to lie within the exponential family of probability distributions. This will simplify the computations and all the updates become updates of parameters that define such distributions!

The Case of the Exponential Family of Probability Distributions

- Looking carefully at (24) and (25), it becomes clear that the practical application of the variational Bayesian EM depends on the computational tractability of the expected values of the $\ln p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}; \boldsymbol{\xi})$.
- We will restrict the involved distributions to lie within the exponential family of probability distributions. This will simplify the computations and all the updates become updates of parameters that define such distributions!

The Case of the Exponential Family of Probability Distributions

- Let us assume that the points in the complete data set $(\boldsymbol{x}_n, \boldsymbol{x}_n^l),\ n = 1, 2, \ldots, N$, are i.i.d. Then,

$$p(\mathcal{X}, \mathcal{X}^l|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta}).$$

The Case of the Exponential Family of Probability Distributions

- We further assume $p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta})$ to lie within the exponential family, i.e.,

$$p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) \exp\Big(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)\Big).$$

- We further adopt a prior for $\boldsymbol{\theta}$ to be of the respective conjugate form, i.e.,

$$p(\boldsymbol{\theta}|\lambda, \boldsymbol{v}) = h(\lambda, \boldsymbol{v})(g(\boldsymbol{\theta}))^{\lambda} \exp\Big(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{v}\Big).$$

The parameters λ, $\boldsymbol{v}$ comprise $\boldsymbol{\xi}$, which will be considered fixed, in order to focus on the specific functional forms which $q_{\mathcal{X}^l}(\cdot)$ and $q_\theta(\cdot)$ get as iterations progress. So, we will relax the notational dependence on the parameters.

The Case of the Exponential Family of Probability Distributions

- We further assume $p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta})$ to lie within the exponential family, i.e.,

$$p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) \exp\Big(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)\Big).$$

- We further adopt a prior for $\boldsymbol{\theta}$ to be of the respective conjugate form, i.e.,

$$p(\boldsymbol{\theta}|\lambda, \boldsymbol{v}) = h(\lambda, \boldsymbol{v})(g(\boldsymbol{\theta}))^{\lambda} \exp\Big(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{v}\Big).$$

The parameters λ, v comprise ξ, which will be considered fixed, in order to focus on the specific functional forms which $q_{\mathcal{X}^l}(\cdot)$ and $q_\theta(\cdot)$ get as iterations progress. So, we will relax the notational dependence on the parameters.

The Case of the Exponential Family of Probability Distributions

- We further assume $p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta})$ to lie within the exponential family, i.e.,

$$p(\boldsymbol{x}_n, \boldsymbol{x}_n^l|\boldsymbol{\theta}) = g(\boldsymbol{\theta}) f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) \exp\Big(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)\Big).$$

- We further adopt a prior for $\boldsymbol{\theta}$ to be of the respective conjugate form, i.e.,

$$p(\boldsymbol{\theta}|\lambda, \boldsymbol{v}) = h(\lambda, \boldsymbol{v})(g(\boldsymbol{\theta}))^{\lambda} \exp\Big(\boldsymbol{\phi}^T(\boldsymbol{\theta}) \boldsymbol{v}\Big).$$

The parameters λ, $\boldsymbol{v}$ comprise $\boldsymbol{\xi}$, which will be considered fixed, in order to focus on the specific functional forms which $q_{\mathcal{X}^l}(\cdot)$ and $q_\theta(\cdot)$ get as iterations progress. So, we will relax the notational dependence on the parameters.

The Case of the Exponential Family of Probability Distributions

- E-step 1a: It turns out, after some simple algebraic manipulation on (24), that this step becomes

$$q_{\boldsymbol{x}_n^l}^{(j+1)}(\boldsymbol{x}_n^l) = \tilde{g} f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) e^{\tilde{\boldsymbol{\phi}}^T \boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)},$$

where $\tilde{g}$ is the respective normalization constant and

$$\tilde{\boldsymbol{\phi}}^T = \mathbb{E}_{q_\theta^{(j)}}[\boldsymbol{\phi}^T(\theta)].$$

This is very interesting indeed. Although no functional form was assumed for $q_{\mathcal{X}^l}$, it turns out to be a member of the exponential family!

- E-Step 1b: From (25) and some algebraic manipulations, it easily turns out that

$$q_\theta^{(j+1)}(\boldsymbol{\theta}) \propto (g(\boldsymbol{\theta}))^{\lambda+N} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\left(\boldsymbol{v} + \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{x}_n^l}^{(j+1)}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right]\right)\right).$$

Thus, the approximation $q_\theta^{(j+1)}(\boldsymbol{\theta})$ of the posterior $p(\boldsymbol{\theta}|\mathcal{X})$ is of the same form as the conjugate prior with

$$\tilde{\lambda} = \lambda + N, \quad \tilde{v} = v + \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{x}_n^l}^{(j+1)}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right].$$

The Case of the Exponential Family of Probability Distributions

- E-step 1a: It turns out, after some simple algebraic manipulation on (24), that this step becomes

$$q_{\boldsymbol{x}_n^l}^{(j+1)}(\boldsymbol{x}_n^l) = \tilde{g} f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) e^{\tilde{\boldsymbol{\phi}}^T \boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)},$$

where $\tilde{g}$ is the respective normalization constant and

$$\tilde{\boldsymbol{\phi}}^T = \mathbb{E}_{q_\theta^{(j)}}[\boldsymbol{\phi}^T(\theta)].$$

This is very interesting indeed. Although no functional form was assumed for $q_{\mathcal{X}^l}$, it turns out to be a member of the exponential family!

- E-Step 1b: From (25) and some algebraic manipulations, it easily turns out that

$$q_\theta^{(j+1)}(\boldsymbol{\theta}) \propto (g(\boldsymbol{\theta}))^{\lambda+N} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\left(\boldsymbol{v} + \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{x}_n^l}^{(j+1)}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right]\right)\right).$$

Thus, the approximation $q_\theta^{(j+1)}(\boldsymbol{\theta})$ of the posterior $p(\boldsymbol{\theta}|\mathcal{X})$ is of the same form as the conjugate prior with

$$\tilde{\lambda} = \lambda + N, \quad \tilde{v} = v + \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{x}_n^l}^{(j+1)}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right].$$

The Case of the Exponential Family of Probability Distributions

- E-step 1a: It turns out, after some simple algebraic manipulation on (24), that this step becomes

$$q_{\boldsymbol{x}_n^l}^{(j+1)}(\boldsymbol{x}_n^l) = \tilde{g} f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) e^{\tilde{\boldsymbol{\phi}}^T \boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)},$$

where $\tilde{g}$ is the respective normalization constant and

$$\tilde{\boldsymbol{\phi}}^T = \mathbb{E}_{q_\theta^{(j)}}[\boldsymbol{\phi}^T(\theta)].$$

This is very interesting indeed. Although no functional form was assumed for $q_{\mathcal{X}^l}$, it turns out to be a member of the exponential family!

- E-Step 1b: From (25) and some algebraic manipulations, it easily turns out that

$$q_\theta^{(j+1)}(\boldsymbol{\theta}) \propto (g(\boldsymbol{\theta}))^{\lambda+N} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\left(\boldsymbol{v} + \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{x}_n^l}^{(j+1)}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right]\right)\right).$$

Thus, the approximation $q_\theta^{(j+1)}(\boldsymbol{\theta})$ of the posterior $p(\boldsymbol{\theta}|\mathcal{X})$ is of the same form as the conjugate prior with

$$\tilde{\lambda} = \lambda + N, \quad \tilde{v} = v + \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{x}_n^l}^{(j+1)}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right].$$

- E-step 1a: It turns out, after some simple algebraic manipulation on (24), that this step becomes

$$q_{\boldsymbol{x}_n^l}^{(j+1)}(\boldsymbol{x}_n^l) = \tilde{g} f(\boldsymbol{x}_n, \boldsymbol{x}_n^l) e^{\tilde{\boldsymbol{\phi}}^T \boldsymbol{u}(\boldsymbol{x}_n, \boldsymbol{x}_n^l)},$$

where $\tilde{g}$ is the respective normalization constant and

$$\tilde{\boldsymbol{\phi}}^T = \mathbb{E}_{q_\theta^{(j)}}[\boldsymbol{\phi}^T(\theta)].$$

This is very interesting indeed. Although no functional form was assumed for $q_{\mathcal{X}^l}$, it turns out to be a member of the exponential family!

- E-Step 1b: From (25) and some algebraic manipulations, it easily turns out that

$$q_\theta^{(j+1)}(\boldsymbol{\theta}) \propto (g(\boldsymbol{\theta}))^{\lambda+N} \exp\left(\boldsymbol{\phi}^T(\boldsymbol{\theta})\left(\boldsymbol{v} + \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{x}_n^l}^{(j+1)}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right]\right)\right).$$

Thus, the approximation $q_\theta^{(j+1)}(\boldsymbol{\theta})$ of the posterior $p(\boldsymbol{\theta}|\mathcal{X})$ is of the same form as the conjugate prior with

$$\tilde{\lambda} = \lambda + N, \quad \tilde{v} = v + \sum_{n=1}^{N} \mathbb{E}_{q_{\boldsymbol{x}_n^l}^{(j+1)}}\left[\boldsymbol{u}(\boldsymbol{x}_n, \mathbf{x}_n^l)\right].$$

A Variational Bayesian Approach to Linear Regression

- Let us consider our familiar regression task,

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\eta},\ \mathbf{y} \in \mathbb{R}^N,\ \boldsymbol{\theta} \in \mathbb{R}^K.$$

  We have already treated the case where $\boldsymbol{\eta}$ was Gaussian and the prior $p(\boldsymbol{\theta})$ was also Gaussian. We used the EM in order to optimize the evidence $p(\boldsymbol{y})$ w.r. to the parameters, which define the two adopted Gaussian pdfs.

- In contrast, now, we will adopt assumptions that do **not allow** for tractable analytic computations of the posterior, $p(\boldsymbol{\theta}|\boldsymbol{y})$, which is a prerequisite both for the standard EM as well as for the analytic computations of the evidence $p(\boldsymbol{y})$.

A Variational Bayesian Approach to Linear Regression

- Let us consider our familiar regression task,

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \boldsymbol{\eta},\ \mathbf{y} \in \mathbb{R}^N,\ \boldsymbol{\theta} \in \mathbb{R}^K.$$

  We have already treated the case where $\boldsymbol{\eta}$ was Gaussian and the prior $p(\boldsymbol{\theta})$ was also Gaussian. We used the EM in order to optimize the evidence $p(\boldsymbol{y})$ w.r. to the parameters, which define the two adopted Gaussian pdfs.

- In contrast, now, we will adopt assumptions that do **not allow** for tractable analytic computations of the posterior, $p(\boldsymbol{\theta}|\boldsymbol{y})$, which is a prerequisite both for the standard EM as well as for the analytic computations of the evidence $p(\boldsymbol{y})$.

A Variational Bayesian Approach to Linear Regression

- Assume that,

$$p(\boldsymbol{y}|\boldsymbol{\theta}, \beta) = \mathcal{N}(\Phi\boldsymbol{\theta}, \beta^{-1}I). \tag{26}$$

That is, the noise is Gaussian and for simplicity we have considered it to be white, $\Sigma_\eta = \sigma_\eta^2 I$, and $\beta = \frac{1}{\sigma_\eta^2}$.

- Concerning the prior of θ, each one of the parameter components, θ_k, is allowed to have a **different variance**, $\sigma_k^2 := \frac{1}{\alpha_k},\ k = 0, 1, \ldots, K-1$. Moreover, the values of β and $\alpha_k,\ k = 0, \ldots, K-1$ will not be treated as deterministic variables, but they are assumed to be **random**, as well.

A Variational Bayesian Approach to Linear Regression

- Assume that,

$$p(\boldsymbol{y}|\boldsymbol{\theta}, \beta) = \mathcal{N}(\Phi\boldsymbol{\theta}, \beta^{-1}I). \tag{26}$$

  That is, the noise is Gaussian and for simplicity we have considered it to be white, $\Sigma_\eta = \sigma_\eta^2 I$, and $\beta = \frac{1}{\sigma_\eta^2}$.

- Concerning the prior of θ, each one of the parameter components, θ_k, is allowed to have a **different variance**, $\sigma_k^2 := \frac{1}{\alpha_k},\ k = 0, 1, \ldots, K-1$. Moreover, the values of β and $\alpha_k,\ k = 0, \ldots, K-1$ will not be treated as deterministic variables, but they are assumed to be **random**, as well.

A Variational Bayesian Approach to Linear Regression

- The respective priors for the unknown random variables are adopted as:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \mathcal{N}(\theta_k|0, \alpha_k^{-1}), \tag{27}$$

$$p(\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \text{Gamma}(\alpha_k|a, b), \tag{28}$$

and

$$p(\beta) = \text{Gamma}(\beta|c, d). \tag{29}$$

The priors indicate that the game will be played within the exponential family terrain. The prior $p(\alpha)$ is the conjugate pair of (27). Also, (29) would be the conjugate of (26), if we had considered θ fixed.

- The respective priors for the unknown random variables are adopted as:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \mathcal{N}(\theta_k|0, \alpha_k^{-1}), \tag{27}$$

$$p(\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \text{Gamma}(\alpha_k|a, b), \tag{28}$$

and

$$p(\beta) = \text{Gamma}(\beta|c, d). \tag{29}$$

The priors indicate that the game will be played within the exponential family terrain. The prior $p(\boldsymbol{\alpha})$ is the conjugate pair of (27). Also, (29) would be the conjugate of (26), if we had considered $\boldsymbol{\theta}$ fixed.

- The respective priors for the unknown random variables are adopted as:

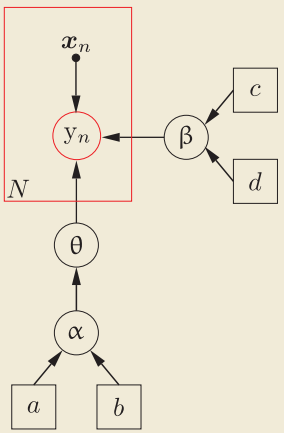
$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \mathcal{N}(\theta_k|0, \alpha_k^{-1}), \tag{27}$$

$$p(\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \text{Gamma}(\alpha_k|a, b), \tag{28}$$

and

$$p(\beta) = \text{Gamma}(\beta|c, d). \tag{29}$$

The priors indicate that the game will be played within the exponential family terrain. The prior $p(\boldsymbol{\alpha})$ is the conjugate pair of (27). Also, (29) would be the conjugate of (26), if we had considered $\boldsymbol{\theta}$ fixed.



A graphical illustration of the dependencies among the various variables involved in the model of linear regression. The red circle indicates the random variable which is observed, gray circles indicate (hidden) random variables and squares correspond to deterministic parameters. The direction of each arrow indicates the direction of the dependence between the connected variables. The red box indicates that the above dependencies hold for all, N, time instants.

A Variational Bayesian Approach to Linear Regression

- Our current task comprises hidden variables in the form of parameters grouped in θ, $\boldsymbol{\alpha}$ and β and it involves no other latent variables. The set of observations is now given by $\boldsymbol{y}$. Also, observe that the posterior $p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta | \boldsymbol{y})$ is not analytically tractable.
- We will resort to the variational Bayesian EM to obtain an estimate of the previous posterior pdfs.

A Variational Bayesian Approach to Linear Regression

- Our current task comprises hidden variables in the form of parameters grouped in θ, $\boldsymbol{\alpha}$ and β and it involves no other latent variables. The set of observations is now given by $\boldsymbol{y}$. Also, observe that the posterior $p(\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta | \boldsymbol{y})$ is not analytically tractable.
- We will resort to the variational Bayesian EM to obtain an estimate of the previous posterior pdfs.

A Variational Bayesian Approach to Linear Regression

- Using the mean field approximation, we assume that the approximation to the posterior (the dependence on $\boldsymbol{y}$ has been suppressed for notational convenience) factorizes as

$$q(\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) = q_\theta(\boldsymbol{\theta}) q_\alpha(\boldsymbol{\alpha}) q_\beta(\beta),$$

  where we have relaxed our notation, for simplicity, from the explicit dependence on a, b, c and d.

- The variational EM consists of three sub-steps, one for each factor in the previous factorized equation. Starting from some initial guesses, for $\mathbb{E}[\beta]$, $\mathbb{E}[\alpha_k]$, $k = 0, \ldots, K-1$, we get:

A Variational Bayesian Approach to Linear Regression

- Using the mean field approximation, we assume that the approximation to the posterior (the dependence on $\boldsymbol{y}$ has been suppressed for notational convenience) factorizes as

$$q(\boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) = q_\theta(\boldsymbol{\theta}) q_\alpha(\boldsymbol{\alpha}) q_\beta(\beta),$$

  where we have relaxed our notation, for simplicity, from the explicit dependence on a, b, c and d.

- The variational EM consists of three sub-steps, one for each factor in the previous factorized equation. Starting from some initial guesses, for $\mathbb{E}[\beta]$, $\mathbb{E}[\alpha_k]$, $k = 0, \ldots, K-1$, we get:

- E-Step 1a: "Rephrasing" the general update form of (25) we have,

$$\ln q_{\theta}^{(j+1)}(\boldsymbol{\theta}) = \mathbb{E}_{q_{\alpha}^{(j)} q_{\beta}^{(j)}}\Big[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)\Big] + \text{constant}.$$

  After some manipulations, the following results.

- Let $A := \text{diag}\Big\{\mathbb{E}[\alpha_0], \ldots, \mathbb{E}[\alpha_{K-1}]\Big\}$. Then,

$$q_{\theta}^{(j+1)}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta}^{(j+1)}, \Sigma_{\theta}^{(j+1)}),$$

  where

$$\Sigma_{\theta}^{(j+1)} = \left(A + \mathbb{E}[\beta]\Phi^T\Phi\right)^{-1}, \quad \boldsymbol{\mu}_{\theta}^{(j+1)} = \mathbb{E}[\beta]\Sigma_{\theta}^{(j+1)}\Phi^T\boldsymbol{y}. \quad (30)$$

  Note that the approximation to the posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$ turns out to be Gaussian, although we did not assumed it to be so. This is a consequence of the particular form of the adopted pdfs, which spring from the exponential family.

- E-Step 1a: "Rephrasing" the general update form of (25) we have,

$$\ln q_{\theta}^{(j+1)}(\boldsymbol{\theta}) = \mathbb{E}_{q_{\alpha}^{(j)} q_{\beta}^{(j)}}\Big[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)\Big] + \text{constant}.$$

After some manipulations, the following results.

- Let $A := \text{diag}\Big\{\mathbb{E}[\alpha_0], \ldots, \mathbb{E}[\alpha_{K-1}]\Big\}$. Then,

$$q_{\theta}^{(j+1)}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta}^{(j+1)}, \Sigma_{\theta}^{(j+1)}),$$

where

$$\Sigma_{\theta}^{(j+1)} = \left(A + \mathbb{E}[\beta]\Phi^T\Phi\right)^{-1}, \quad \boldsymbol{\mu}_{\theta}^{(j+1)} = \mathbb{E}[\beta]\Sigma_{\theta}^{(j+1)}\Phi^T\boldsymbol{y}. \quad (30)$$

Note that the approximation to the posterior $p(\theta|y)$ turns out to be Gaussian, although we did not assumed it to be so. This is a consequence of the particular form of the adopted pdfs, which spring from the exponential family.

- E-Step 1a: "Rephrasing" the general update form of (25) we have,

$$\ln q_{\theta}^{(j+1)}(\boldsymbol{\theta}) = \mathbb{E}_{q_{\alpha}^{(j)} q_{\beta}^{(j)}}\Big[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)\Big] + \text{constant}.$$

After some manipulations, the following results.

- Let $A := \text{diag}\Big\{\mathbb{E}[\alpha_0], \ldots, \mathbb{E}[\alpha_{K-1}]\Big\}$. Then,

$$q_{\theta}^{(j+1)}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_{\theta}^{(j+1)}, \Sigma_{\theta}^{(j+1)}),$$

where

$$\Sigma_{\theta}^{(j+1)} = \left(A + \mathbb{E}[\beta]\Phi^T\Phi\right)^{-1}, \quad \boldsymbol{\mu}_{\theta}^{(j+1)} = \mathbb{E}[\beta]\Sigma_{\theta}^{(j+1)}\Phi^T\boldsymbol{y}. \quad (30)$$

Note that the approximation to the posterior $p(\boldsymbol{\theta}|\boldsymbol{y})$ turns out to be Gaussian, although we did not assumed it to be so. This is a consequence of the particular form of the adopted pdfs, which spring from the exponential family.

- E-Step 1b: We have that:

$$\ln q_{\alpha}^{(j+1)}(\boldsymbol{\alpha}) = \mathbb{E}_{q_{\theta}^{(j+1)} q_{\beta}^{(j)}}[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)] + \text{constant}$$
$$= \mathbb{E}_{q_{\theta}^{(j+1)} q_{\beta}^{(j)}}[\ln p(\boldsymbol{\theta}|\boldsymbol{\alpha}) + \ln p(\boldsymbol{\alpha})] + \text{constant},$$

which finally leads to (for $k = 0, \ldots, K-1$)

$$q_{\alpha}^{(j+1)}(\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \text{Gamma}(\alpha_k|\tilde{a}, \tilde{b}_k), \quad \tilde{a} = a + \frac{1}{2}, \quad \tilde{b}_k = b + \frac{1}{2}\mathbb{E}_{q_{\theta}^{(j+1)}}[\theta_k^2].$$

- Note that in the previous recursions we still need to compute the following ($k = 0, 1, \ldots, K-1$):

$$\mathbb{E}[\theta_k^2] = \left[E_{q_{\theta}^{(j+1)}}[\boldsymbol{\theta}\boldsymbol{\theta}^T]\right]_{kk} = \left[\Sigma_{\theta}^{(j+1)} + \boldsymbol{\mu}_{\theta}^{(j+1)}\boldsymbol{\mu}_{\theta}^{(j+1)T}\right]_{kk},$$

where $[A]_{kk}$ denotes the (k, k) element of A. We still need to compute $\mathbb{E}[\alpha_k]$, $k = 0, 1, \ldots, K-1$, to be used in the next iteration in E-Step 1a. However, each α_k follows a gamma distribution, hence

$$\mathbb{E}_{q_{\alpha}^{(j+1)}}[\alpha_k] = \frac{\tilde{a}}{\tilde{b}_k}.$$

- E-Step 1b: We have that:

$$\ln q_{\alpha}^{(j+1)}(\boldsymbol{\alpha}) = \mathbb{E}_{q_{\theta}^{(j+1)} q_{\beta}^{(j)}}[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)] + \text{constant}$$
$$= \mathbb{E}_{q_{\theta}^{(j+1)} q_{\beta}^{(j)}}[\ln p(\boldsymbol{\theta}|\boldsymbol{\alpha}) + \ln p(\boldsymbol{\alpha})] + \text{constant},$$

which finally leads to (for $k = 0, \ldots, K-1$)

$$q_{\alpha}^{(j+1)}(\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \text{Gamma}(\alpha_k|\tilde{a}, \tilde{b}_k), \quad \tilde{a} = a + \frac{1}{2}, \quad \tilde{b}_k = b + \frac{1}{2}\mathbb{E}_{q_{\theta}^{(j+1)}}[\theta_k^2].$$

- Note that in the previous recursions we still need to compute the following ($k = 0, 1, \ldots, K-1$):

$$\mathbb{E}[\theta_k^2] = \left[E_{q_{\theta}^{(j+1)}}[\boldsymbol{\theta}\boldsymbol{\theta}^T]\right]_{kk} = \left[\Sigma_{\theta}^{(j+1)} + \boldsymbol{\mu}_{\theta}^{(j+1)}\boldsymbol{\mu}_{\theta}^{(j+1)T}\right]_{kk},$$

where $[A]_{kk}$ denotes the (k, k) element of A. We still need to compute $\mathbb{E}[\alpha_k]$, $k = 0, 1, \ldots, K-1$, to be used in the next iteration in E-Step 1a. However, each α_k follows a gamma distribution, hence

$$\mathbb{E}_{q_{\alpha}^{(j+1)}}[\alpha_k] = \frac{\tilde{a}}{\tilde{b}_k}.$$

- E-Step 1b: We have that:

$$\ln q_{\alpha}^{(j+1)}(\boldsymbol{\alpha}) = \mathbb{E}_{q_{\theta}^{(j+1)} q_{\beta}^{(j)}}[\ln p(\boldsymbol{y}, \theta, \boldsymbol{\alpha}, \beta)] + \text{constant}$$
$$= \mathbb{E}_{q_{\theta}^{(j+1)} q_{\beta}^{(j)}}[\ln p(\theta|\boldsymbol{\alpha}) + \ln p(\boldsymbol{\alpha})] + \text{constant},$$

which finally leads to (for $k = 0, \ldots, K-1$)

$$q_{\alpha}^{(j+1)}(\boldsymbol{\alpha}) = \prod_{k=0}^{K-1} \text{Gamma}(\alpha_k|\tilde{a}, \tilde{b}_k), \quad \tilde{a} = a + \frac{1}{2}, \quad \tilde{b}_k = b + \frac{1}{2}\mathbb{E}_{q_{\theta}^{(j+1)}}[\theta_k^2].$$

- Note that in the previous recursions we still need to compute the following ($k = 0, 1, \ldots, K-1$):

$$\mathbb{E}[\theta_k^2] = \left[E_{q_{\theta}^{(j+1)}}[\theta\theta^T]\right]_{kk} = \left[\Sigma_{\theta}^{(j+1)} + \boldsymbol{\mu}_{\theta}^{(j+1)}\boldsymbol{\mu}_{\theta}^{(j+1)T}\right]_{kk},$$

where $[A]_{kk}$ denotes the (k, k) element of A. We still need to compute $\mathbb{E}[\alpha_k]$, $k = 0, 1, \ldots, K-1$, to be used in the next iteration in E-Step 1a. However, each α_k follows a gamma distribution, hence

$$\mathbb{E}_{q_{\alpha}^{(j+1)}}[\alpha_k] = \frac{\tilde{a}}{\tilde{b}_k}.$$

- E-Step 1c: From the general rule we have:

$$\ln q_{\beta}^{(j+1)}(\beta) = \mathbb{E}_{q_{\theta}^{(j+1)} q_{\alpha}^{(j+1)}}\Big[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)\Big] + \text{constant}$$

$$= \mathbb{E}_{q_{\theta}^{(j+1)} q_{\alpha}^{(j+1)}}\Big[\ln p(\boldsymbol{y}|\boldsymbol{\theta}, \beta) + \ln p(\beta)\Big] + \text{constant},$$

which finally results in

$$q_{\beta}^{(j+1)}(\beta) = \text{Gamma}(\beta|\tilde{c}, \tilde{d}),$$

where $\tilde{c} = c + \frac{N}{2}$, $\tilde{d} = d + \frac{1}{2}\mathbb{E}_{q_{\theta}^{(j+1)}}[\|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2]$.

- To complete the recursions we need the expectation

$$\mathbb{E}_{q_{\theta}^{(j+1)}}[\|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2] = \|\boldsymbol{y} - \Phi\boldsymbol{\mu}_{\theta}^{(j+1)}\|^2 + \text{trace}\left\{\Phi\Sigma_{\theta}^{(j+1)}\Phi^T\right\}.$$

Also, for the E-Step 1a of the next iteration we need to compute,

$$\mathbb{E}_{q_{\beta}^{(j+1)}}[\beta] = \frac{\tilde{c}}{\tilde{d}}.$$

- E-Step 1c: From the general rule we have:

$$\ln q_{\beta}^{(j+1)}(\beta) = \mathbb{E}_{q_{\theta}^{(j+1)} q_{\alpha}^{(j+1)}} \Big[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta) \Big] + \text{constant}$$

$$= \mathbb{E}_{q_{\theta}^{(j+1)} q_{\alpha}^{(j+1)}} \Big[\ln p(\boldsymbol{y}|\boldsymbol{\theta}, \beta) + \ln p(\beta) \Big] + \text{constant},$$

which finally results in

$$q_{\beta}^{(j+1)}(\beta) = \text{Gamma}(\beta|\tilde{c}, \tilde{d}),$$

where $\tilde{c} = c + \frac{N}{2}$, $\tilde{d} = d + \frac{1}{2}\mathbb{E}_{q_{\theta}^{(j+1)}}[\|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2]$.

- To complete the recursions we need the expectation

$$\mathbb{E}_{q_{\theta}^{(j+1)}}[\|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2] = \|\boldsymbol{y} - \Phi\boldsymbol{\mu}_{\theta}^{(j+1)}\|^2 + \text{trace}\left\{\Phi\Sigma_{\theta}^{(j+1)}\Phi^T\right\}.$$

Also, for the E-Step 1a of the next iteration we need to compute,

$$\mathbb{E}_{q_{\beta}^{(j+1)}}[\beta] = \frac{\tilde{c}}{\tilde{d}}.$$

- E-Step 1c: From the general rule we have:

$$\ln q_{\beta}^{(j+1)}(\beta) = \mathbb{E}_{q_{\theta}^{(j+1)} q_{\alpha}^{(j+1)}}\Big[\ln p(\boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\alpha}, \beta)\Big] + \text{constant}$$
$$= \mathbb{E}_{q_{\theta}^{(j+1)} q_{\alpha}^{(j+1)}}\Big[\ln p(\boldsymbol{y}|\boldsymbol{\theta}, \beta) + \ln p(\beta)\Big] + \text{constant},$$

  which finally results in

$$q_{\beta}^{(j+1)}(\beta) = \text{Gamma}(\beta|\tilde{c}, \tilde{d}),$$

  where $\tilde{c} = c + \frac{N}{2}$, $\tilde{d} = d + \frac{1}{2}\mathbb{E}_{q_{\theta}^{(j+1)}}[\|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2]$.

- To complete the recursions we need the expectation

$$\mathbb{E}_{q_{\theta}^{(j+1)}}[\|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|^2] = \|\boldsymbol{y} - \Phi\boldsymbol{\mu}_{\theta}^{(j+1)}\|^2 + \text{trace}\left\{\Phi\Sigma_{\theta}^{(j+1)}\Phi^T\right\}.$$

  Also, for the E-Step 1a of the next iteration we need to compute,

$$\mathbb{E}_{q_{\beta}^{(j+1)}}[\beta] = \frac{\tilde{c}}{\tilde{d}}.$$

A Variational Bayesian Approach to Linear Regression

- In this case we have chosen $a = b = c = d = 10^{-6}$. Otherwise, an extra optimization step is required.
- Once the algorithm has converged, predictions can be made on the basis of the predictive distribution (recall (6)), i.e.,

$$p(y|\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(y|\mu_y, \sigma_y^2\right),$$

where

$$\mu_y = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\mu}_{\theta|y}, \quad \sigma_y^2 = \sigma_\eta^2 + \boldsymbol{\phi}^T(\boldsymbol{x})\Sigma_{\theta|y}\boldsymbol{\phi}(\boldsymbol{x})$$

by replacing $\Sigma_{\theta|y}$, $\boldsymbol{\mu}_{\theta|y}$ and σ_η^2 by the converged values of Σ_θ, $\boldsymbol{\mu}_\theta$ and $\mathbb{E}[\beta]$, respectively.

- Note, however, that this is only an approximation, since the Gaussian form for the posterior of the parameters is a result of the mean field approximation and also we have used the mean value, $\mathbb{E}[\beta]$, in place of the noise variance. The latter can be justified that as the number of training samples increases, the distribution of β sharply peaks around its mean value.

A Variational Bayesian Approach to Linear Regression

- In this case we have chosen $a = b = c = d = 10^{-6}$. Otherwise, an extra optimization step is required.
- Once the algorithm has converged, predictions can be made on the basis of the predictive distribution (recall (6)), i.e.,

$$p(y|\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(y|\mu_y, \sigma_y^2\right),$$

where

$$\mu_y = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\mu}_{\theta|y}, \quad \sigma_y^2 = \sigma_\eta^2 + \boldsymbol{\phi}^T(\boldsymbol{x})\Sigma_{\theta|y}\boldsymbol{\phi}(\boldsymbol{x})$$

by replacing $\Sigma_{\theta|y}$, $\boldsymbol{\mu}_{\theta|y}$ and σ_η^2 by the converged values of Σ_θ, $\boldsymbol{\mu}_\theta$ and $\mathbb{E}[\beta]$, respectively.

- Note, however, that this is only an approximation, since the Gaussian form for the posterior of the parameters is a result of the mean field approximation and also we have used the mean value, $\mathbb{E}[\beta]$, in place of the noise variance. The latter can be justified that as the number of training samples increases, the distribution of β sharply peaks around its mean value.

A Variational Bayesian Approach to Linear Regression

- In this case we have chosen $a = b = c = d = 10^{-6}$. Otherwise, an extra optimization step is required.
- Once the algorithm has converged, predictions can be made on the basis of the predictive distribution (recall (6)), i.e.,

$$p(y|\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(y|\mu_y, \sigma_y^2\right),$$

where

$$\mu_y = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\mu}_{\theta|y}, \quad \sigma_y^2 = \sigma_\eta^2 + \boldsymbol{\phi}^T(\boldsymbol{x})\Sigma_{\theta|y}\boldsymbol{\phi}(\boldsymbol{x})$$

by replacing $\Sigma_{\theta|y}$, $\boldsymbol{\mu}_{\theta|y}$ and σ_η^2 by the converged values of Σ_θ, $\boldsymbol{\mu}_\theta$ and $\mathbb{E}[\beta]$, respectively.

- Note, however, that this is only an approximation, since the Gaussian form for the posterior of the parameters is a result of the mean field approximation and also we have used the mean value, $\mathbb{E}[\beta]$, in place of the noise variance. The latter can be justified that as the number of training samples increases, the distribution of β sharply peaks around its mean value.

An Example

- The goal of this example is to demonstrate the comparative performance, via a simulation example, of a) the variational Bayesian method, b) the Maximum Likelihood/LS, and c) the EM algorithm of based on Gaussian assumptions, as discussed in the beginning of the lectures in the context of linear regression.

An Example

- To this end, we generate the training data according to the following scenario. The interval in the real axis $[-10,\ 10]$ was sampled at $N = 100$ equidistant points, $x_n,\ n = 1, 2, \ldots, 100$. The training data comprise the pairs $(y_n, x_n),\ n = 1, 2, \ldots, N$, where

$$y_n = \exp\left(-\frac{1}{2}\frac{(x_n + 5.8)^2}{0.1}\right) + \exp\left(-\frac{1}{2}\frac{(x_n - 2.6)^2}{0.1}\right) + \eta_n$$

  where η_n are i.i.d zero mean Gaussian noise samples, of variance $\sigma_\eta^2 = 0.015$. To fit the data the following model was adopted:

$$y = \sum_{k=1}^{N} \theta_k \exp\left(-\frac{1}{2}\frac{(x - x_k)^2}{0.1}\right).$$

- Thus, the matrix Φ has the following elements

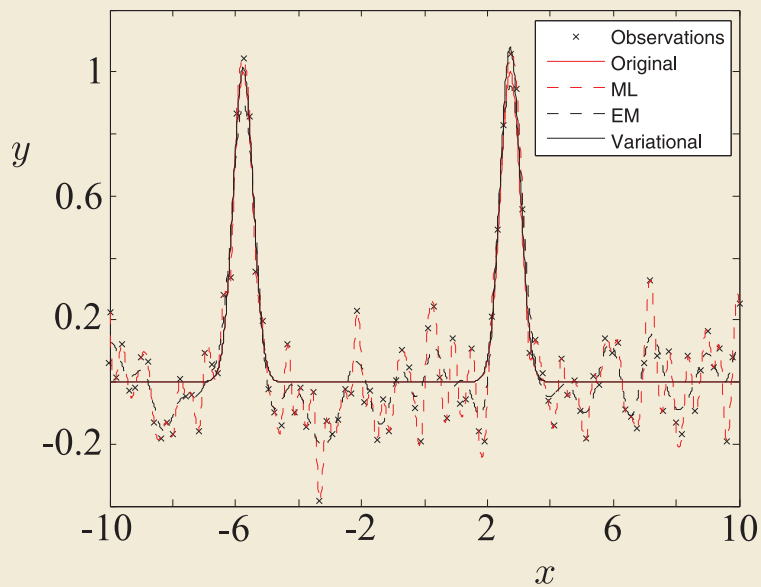
$$[\Phi]_{nk} = \exp\left(-\frac{1}{2}\frac{(x_n - x_k)^2}{0.1}\right),\ n = 1, 2, \ldots, N,\ \ k = 1, 2, \ldots, N.$$

An Example

- Note that we have used as many parameters as the number of data points. Naturally, this will lead to overfitting. The essence of the example is to demonstrate the power of the variational Bayesian method, when we use **different variances** for the different parameters. This provides a sparsifying nature to the approach; this will be justified soon.

An Example

- Note that we have used as many parameters as the number of data points. Naturally, this will lead to overfitting. The essence of the example is to demonstrate the power of the variational Bayesian method, when we use **different variances** for the different parameters. This provides a sparsifying nature to the approach; this will be justified soon.



The red full-line curve corresponds to the true function which generates the data. The gray full-curve corresponds to the model, having plugged in as estimated values $\hat{\theta}_k$ the respective posterior mean values from (30). The dotted red curve corresponds to the ML solution and the dotted gray curve to the EM, where the estimates correspond to means of the respective posteriors, ((4), using the resulting EM estimates). The performance advantages of the variational approach are obvious, which almost coincides with the true one.

When Bayesian Inference Meets Sparsity

- The close relationship between the use of a prior pdf and the regularization of a cost function has already been discussed. There, the adoption of a Gaussian prior together with a Gaussian noise for the regression task led to the equivalence of MAP with the ridge regression.
- It will not take a minute to show that the use of a Gaussian model for the noise together with a Laplacian prior for each one of the weights, i.e.,

$$p(\theta_k) = \frac{\lambda}{2} \exp\Big(- \lambda |\theta_k| \Big),$$

renders MAP equivalent to the ℓ_1 norm regularization of the LS cost.

When Bayesian Inference Meets Sparsity

- The close relationship between the use of a prior pdf and the regularization of a cost function has already been discussed. There, the adoption of a Gaussian prior together with a Gaussian noise for the regression task led to the equivalence of MAP with the ridge regression.
- It will not take a minute to show that the use of a Gaussian model for the noise together with a Laplacian prior for each one of the weights, i.e.,

$$p(\theta_k) = \frac{\lambda}{2} \exp\Big(-\lambda|\theta_k|\Big),$$

renders MAP equivalent to the ℓ_1 norm regularization of the LS cost.

When Bayesian Inference Meets Sparsity

- For a Bayesian, however, who is not interested in cost functions, the secret that lies within the Laplacian prior is hidden in the so called heavy tails of this distribution. This is in contrast to a Gaussian pdf, which has very light tails.
- In other words, the probability that an observation of a Gaussian random variable can take values far from its mean **decreases very fast**. For example, the probability of observing variables that deviate from the mean by more than 2σ, 3σ, 4σ and 5σ are 0.046, 0.003, 6×10^{-5} and 6×10^{-7}, respectively.
- That is, with a Gaussian prior, the learning process looks for values "around" the mean; values away form the mean are heavily penalized.

When Bayesian Inference Meets Sparsity

- For a Bayesian, however, who is not interested in cost functions, the secret that lies within the Laplacian prior is hidden in the so called heavy tails of this distribution. This is in contrast to a Gaussian pdf, which has very light tails.
- In other words, the probability that an observation of a Gaussian random variable can take values far from its mean **decreases very fast**. For example, the probability of observing variables that deviate from the mean by more than 2σ, 3σ, 4σ and 5σ are 0.046, 0.003, 6×10^{-5} and 6×10^{-7}, respectively.
- That is, with a Gaussian prior, the learning process looks for values "around" the mean; values away form the mean are heavily penalized.

When Bayesian Inference Meets Sparsity

- For a Bayesian, however, who is not interested in cost functions, the secret that lies within the Laplacian prior is hidden in the so called heavy tails of this distribution. This is in contrast to a Gaussian pdf, which has very light tails.
- In other words, the probability that an observation of a Gaussian random variable can take values far from its mean **decreases very fast**. For example, the probability of observing variables that deviate from the mean by more than 2σ, 3σ, 4σ and 5σ are 0.046, 0.003, 6×10^{-5} and 6×10^{-7}, respectively.
- That is, with a Gaussian prior, the learning process looks for values "around" the mean; values away form the mean are heavily penalized.

When Bayesian Inference Meets Sparsity

- Thus, in sparsity-aware learning the use of a Gaussian would be the wrong information to pass over to the learning mechanism.
- Assuming the mean of the prior to be zero, although we expect most of the components of our parameters to be zero, still we want a few of them to be large. Hence, our prior information should be selected such as to assign small (but not too small) probabilities to large values.
- To a Bayesian, sparsity-aware learning becomes synonymous with imposing **heavy-tail** priors. Let us now turn back to our current task, and see how this brief introduction is related to our model.

When Bayesian Inference Meets Sparsity

- Thus, in sparsity-aware learning the use of a Gaussian would be the wrong information to pass over to the learning mechanism.
- Assuming the mean of the prior to be zero, although we expect most of the components of our parameters to be zero, still we want a few of them to be large. Hence, our prior information should be selected such as to assign small (but not too small) probabilities to large values.
- To a Bayesian, sparsity-aware learning becomes synonymous with imposing **heavy-tail** priors. Let us now turn back to our current task, and see how this brief introduction is related to our model.

When Bayesian Inference Meets Sparsity

- Thus, in sparsity-aware learning the use of a Gaussian would be the wrong information to pass over to the learning mechanism.
- Assuming the mean of the prior to be zero, although we expect most of the components of our parameters to be zero, still we want a few of them to be large. Hence, our prior information should be selected such as to assign small (but not too small) probabilities to large values.
- To a Bayesian, sparsity-aware learning becomes synonymous with imposing **heavy-tail** priors. Let us now turn back to our current task, and see how this brief introduction is related to our model.

When Bayesian Inference Meets Sparsity

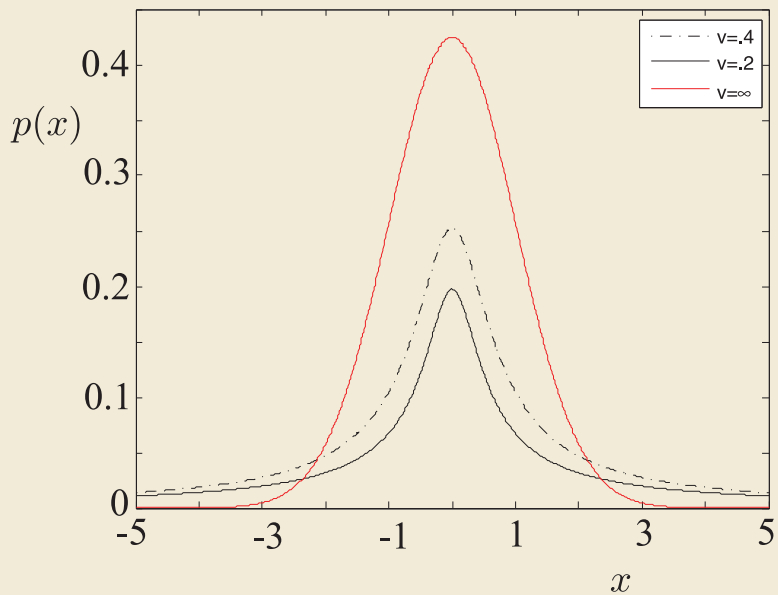
- Our prior pdf, $p(\boldsymbol{\theta})$, according to the model (27)-(28) is obtained by marginalizing out the hyperparameters $\boldsymbol{\alpha}$, i.e.,

$$
\begin{aligned}
p(\boldsymbol{\theta}; a, b) &= \int p(\boldsymbol{\theta}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}) d\boldsymbol{\alpha} \\
&= \int \prod_{k=0}^{K-1} \mathcal{N}(\theta_k|0, \alpha_k^{-1}) \mathrm{Gamma}(\alpha_k|a, b) d\boldsymbol{\alpha} \\
&= \prod_{k=0}^{K-1} \mathrm{st}(\theta_k|0, \frac{a}{b}, 2a),
\end{aligned}
$$

where student's-t pdf is defined as

$$
\mathrm{st}(x|\mu, \lambda, \nu) := \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \left(\frac{\lambda}{\pi\nu}\right)^{1/2} \frac{1}{\left(1 + \frac{\lambda(x-\mu)^2}{\nu}\right)^{\frac{\nu+1}{2}}}.
$$

- The parameter ν is known as the number of degrees of freedom. The figure below shows the graph of student's-t pdfs for different values of ν. For $\nu \to \infty$, the student's-t distribution tends to a Gaussian of the same mean and precision λ. Observe the heavy tail feature of student's-t pdf, especially for low values of ν. Recall that in our case, where we have used uninformative hyperpriors, the hyperparameter, a, was given a small value.



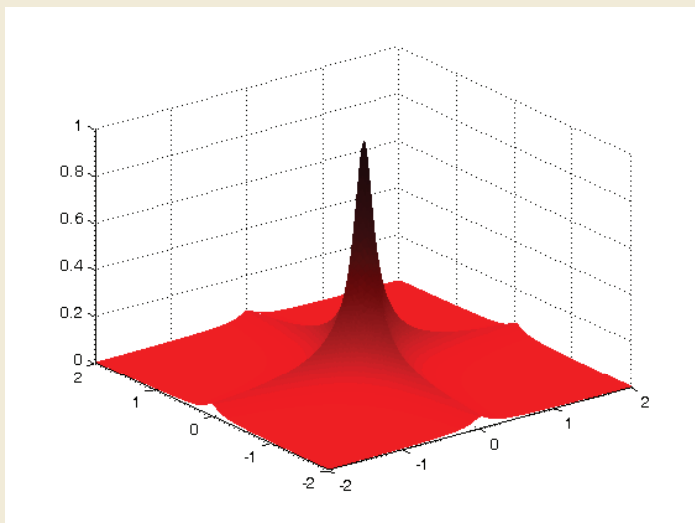
When Bayesian Inference Meets Sparsity

- Thus, our treatment of the regression task favors sparse solutions. It will push as many of the coefficients, θ_k, as possible towards zero. That is, it prunes the less relevant basis functions, $\phi_k(\boldsymbol{x})$, by setting the corresponding coefficients to zero.
- This is also the reason for using different hyperparameters, α_k, for each one of the parameters, θ_k, $k = 0, 2, \ldots, K-1$, which provide allows the learning procedure to adjust each one of the parameters individually. This approach was coined Automatic Relevance Determination (ARD).

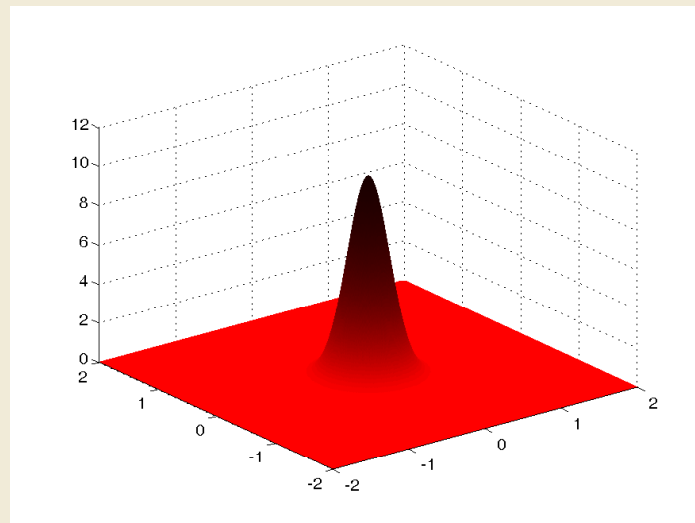
When Bayesian Inference Meets Sparsity

- Thus, our treatment of the regression task favors sparse solutions. It will push as many of the coefficients, θ_k, as possible towards zero. That is, it prunes the less relevant basis functions, $\phi_k(\boldsymbol{x})$, by setting the corresponding coefficients to zero.
- This is also the reason for using different hyperparameters, α_k, for each one of the parameters, $\theta_k,\ k = 0, 2, \ldots, K-1$, which provide allows the learning procedure to adjust each one of the parameters individually. This approach was coined Automatic Relevance Determination (ARD).

- Figure (a) provides a clear demonstration of the sparsity imposing properties of the student's-t distribution. In the two-dimensional space, and as we move away from zero, probability mass is skewed towards the coordinate axes; that is, the pdf peaks around sparse solutions and sparsity is now enforced probabilistically. In contrast, the Gaussian does not give much chance to large values, Figure (b)



(a)



(b)

A Variational Bayesian Approach to Gaussian Mixture Modeling

- One of the problems, that may be encountered in practice in the Gaussian mixture task via the standard EM algorithm, is when one of the mixture components happens to get centered at (or very close to) **one** of the data points, e.g., $\boldsymbol{\mu}_k^{(j+1)} = \boldsymbol{x}_n$, for some values of k and n.
- In such a case, the exponent term of the respective Gaussian becomes one and the contribution of this particular component in the log likelihood is equal to $(2\pi\sigma_k^2)^{-l/2}$. If, in addition, σ_k is very small, this will lead the likelihood to a large value, although this is **not indicative** that the true model has been learned.
- One way to bypass this drawback is to enforce priors on the involved parameters and resort to a variational Bayesian philosophy to estimate the quantities of interest.

A Variational Bayesian Approach to Gaussian Mixture Modeling

- One of the problems, that may be encountered in practice in the Gaussian mixture task via the standard EM algorithm, is when one of the mixture components happens to get centered at (or very close to) **one** of the data points, e.g., $\boldsymbol{\mu}_k^{(j+1)} = \boldsymbol{x}_n$, for some values of k and n.
- In such a case, the exponent term of the respective Gaussian becomes one and the contribution of this particular component in the log likelihood is equal to $(2\pi\sigma_k^2)^{-l/2}$. If, in addition, σ_k is very small, this will lead the likelihood to a large value, although this is **not indicative** that the true model has been learned.
- One way to bypass this drawback is to enforce priors on the involved parameters and resort to a variational Bayesian philosophy to estimate the quantities of interest.

A Variational Bayesian Approach to Gaussian Mixture Modeling

- One of the problems, that may be encountered in practice in the Gaussian mixture task via the standard EM algorithm, is when one of the mixture components happens to get centered at (or very close to) **one** of the data points, e.g., $\boldsymbol{\mu}_k^{(j+1)} = \boldsymbol{x}_n$, for some values of k and n.
- In such a case, the exponent term of the respective Gaussian becomes one and the contribution of this particular component in the log likelihood is equal to $(2\pi\sigma_k^2)^{-l/2}$. If, in addition, σ_k is very small, this will lead the likelihood to a large value, although this is **not indicative** that the true model has been learned.
- One way to bypass this drawback is to enforce priors on the involved parameters and resort to a variational Bayesian philosophy to estimate the quantities of interest.

A Variational Bayesian Approach to Gaussian Mixture Modeling

- The starting point is the set of observations, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. Assume that the respective pdf model is:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, Q_k^{-1}), \ \boldsymbol{x} \in \mathbb{R}^l.$$

  The unknown parameters, to be estimated are: $(P_k, \boldsymbol{\mu}_k, Q_k)\Big|_{k=1}^{K}$.

- We already know that this is a typical task with latent variables and the complete set comprises $(\boldsymbol{x}_n, k_n), \ n = 1, 2, \ldots, N$, with k_n being the index of the respective mixture, $k_n = 1, 2, \ldots, K$.

- In our previous treatment of the mixture task, via the standard EM, the information about each one of the latent variables, k_n, entered into the problem via the posterior $P(k_n|\boldsymbol{x}_n, \boldsymbol{P})$.

A Variational Bayesian Approach to Gaussian Mixture Modeling

- The starting point is the set of observations, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. Assume that the respective pdf model is:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, Q_k^{-1}), \ \boldsymbol{x} \in \mathbb{R}^l.$$

  The unknown parameters, to be estimated are: $(P_k, \boldsymbol{\mu}_k, Q_k)\Big|_{k=1}^{K}$.

- We already know that this is a typical task with latent variables and the complete set comprises $(\boldsymbol{x}_n, k_n), \ n = 1, 2, \ldots, N$, with k_n being the index of the respective mixture, $k_n = 1, 2, \ldots, K$.

- In our previous treatment of the mixture task, via the standard EM, the information about each one of the latent variables, k_n, entered into the problem via the posterior $P(k_n|\boldsymbol{x}_n, \boldsymbol{P})$.

A Variational Bayesian Approach to Gaussian Mixture Modeling

- The starting point is the set of observations, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$. Assume that the respective pdf model is:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} P_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, Q_k^{-1}), \ \boldsymbol{x} \in \mathbb{R}^l.$$

  The unknown parameters, to be estimated are: $(P_k, \boldsymbol{\mu}_k, Q_k)\Big|_{k=1}^{K}$.

- We already know that this is a typical task with latent variables and the complete set comprises $(\boldsymbol{x}_n, k_n), \ n = 1, 2, \ldots, N$, with k_n being the index of the respective mixture, $k_n = 1, 2, \ldots, K$.

- In our previous treatment of the mixture task, via the standard EM, the information about each one of the latent variables, k_n, entered into the problem via the posterior $P(k_n|\boldsymbol{x}_n, \boldsymbol{P})$.

A Variational Bayesian Approach to Gaussian Mixture Modeling

- In contrast, now, an auxiliary latent random vector is introduced, $\mathbf{z}_n \in \mathbb{R}^K$, for each observation, $n = 1, 2, \ldots, N$. Its components take binary values, such as

$$z_{n_k} \in \{0, 1\}, \text{ and } \sum_{k=1}^{K} z_{n_k} = 1, \tag{31}$$

and they are used as indicators of the respective mixture from which the observation at time n, $\boldsymbol{x}_n$, was drawn; that is, if $z_{n_k} = 1$ it indicates that $\boldsymbol{x}_n$ was drawn from the k-th distribution.

- Obviously,

$$P(z_{n_k} = 1) = P_k,$$

and for any $\boldsymbol{z}_n \in \mathbb{R}^K$ that satisfies (31)

$$P(\boldsymbol{z}_n) = \prod_{k=1}^{K} P_k^{z_{n_k}}.$$

A Variational Bayesian Approach to Gaussian Mixture Modeling

- In contrast, now, an auxiliary latent random vector is introduced, $\mathbf{z}_n \in \mathbb{R}^K$, for each observation, $n = 1, 2, \ldots, N$. Its components take binary values, such as

$$z_{n_k} \in \{0, 1\}, \text{ and } \sum_{k=1}^{K} z_{n_k} = 1, \tag{31}$$

  and they are used as indicators of the respective mixture from which the observation at time n, $\boldsymbol{x}_n$, was drawn; that is, if $z_{n_k} = 1$ it indicates that $\boldsymbol{x}_n$ was drawn from the k-th distribution.

- Obviously,

$$P(z_{n_k} = 1) = P_k,$$

  and for any $\boldsymbol{z}_n \in \mathbb{R}^K$ that satisfies (31)

$$P(\boldsymbol{z}_n) = \prod_{k=1}^{K} P_k^{z_{n_k}}.$$

- Hence, the probability of occurrence of the set $\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ is

$$P(\mathcal{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} P_k^{z_{n_k}}.$$

  Hence, the N latent variables follow a standard multinomial probability distribution.

- In the sequel, we adopt the following prior pdfs,

$$p(\boldsymbol{\mu}_k) = \mathcal{N}\left(\boldsymbol{\mu}_k | 0, \beta^{-1} I\right)$$

  and

$$p(Q_k) = \mathcal{W}(Q_k | W_0, \nu_0),$$

  for fixed ν_0, W_0 and β.

- That is, the adopted priors are Gaussian for the mean values and Wishart pdfs for the precision matrices, respectively. We will treat $\boldsymbol{P} = [P_1, \ldots, P_k]^T$ as deterministic parameters whose optimized value is obtained in the M-step.

- Hence, the probability of occurrence of the set $\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ is

$$P(\mathcal{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} P_k^{z_{n_k}}.$$

  Hence, the N latent variables follow a standard multinomial probability distribution.

- In the sequel, we adopt the following prior pdfs,

$$p(\boldsymbol{\mu}_k) = \mathcal{N}\left(\boldsymbol{\mu}_k | 0, \beta^{-1} I\right)$$

  and

$$p(Q_k) = \mathcal{W}(Q_k | W_0, \nu_0),$$

  for fixed ν_0, W_0 and β.

- That is, the adopted priors are Gaussian for the mean values and Wishart pdfs for the precision matrices, respectively. We will treat $\boldsymbol{P} = [P_1, \ldots, P_k]^T$ as deterministic parameters whose optimized value is obtained in the M-step.

- Hence, the probability of occurrence of the set $\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ is

$$P(\mathcal{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} P_k^{z_{n_k}}.$$

  Hence, the N latent variables follow a standard multinomial probability distribution.

- In the sequel, we adopt the following prior pdfs,

$$p(\boldsymbol{\mu}_k) = \mathcal{N}\left(\boldsymbol{\mu}_k | 0, \beta^{-1} I\right)$$

  and

$$p(Q_k) = \mathcal{W}(Q_k | W_0, \nu_0),$$

  for fixed ν_0, W_0 and β.

- That is, the adopted priors are Gaussian for the mean values and Wishart pdfs for the precision matrices, respectively. We will treat $\boldsymbol{P} = [P_1, \ldots, P_k]^T$ as deterministic parameters whose optimized value is obtained in the M-step.

A Variational Bayesian Approach to Gaussian Mixture Modeling

- Following the philosophy of the variational Bayesian EM, we adopt

$$q(\mathcal{Z}, \boldsymbol{\mu}_{1:K}, Q_{1:K}) = q_z(\mathcal{Z}) q_\mu(\boldsymbol{\mu}_{1:K}) q_Q(Q_{1:K}),$$

where $\boldsymbol{\mu}_{1:K}$ and $Q_{1:K}$ indicate the collections $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ and $\{Q_1, \ldots, Q_K\}$, respectively.

- Furthermore, observe that the conditional pdf of the observations can now be written as

$$p(\mathcal{X}|\mathcal{Z}, \boldsymbol{\mu}_{1:K}, Q_{1:K}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, Q_k^{-1})\right)^{z_{n_k}}.$$

A Variational Bayesian Approach to Gaussian Mixture Modeling

- Following the philosophy of the variational Bayesian EM, we adopt

$$q(\mathcal{Z}, \boldsymbol{\mu}_{1:K}, Q_{1:K}) = q_z(\mathcal{Z}) q_\mu(\boldsymbol{\mu}_{1:K}) q_Q(Q_{1:K}),$$

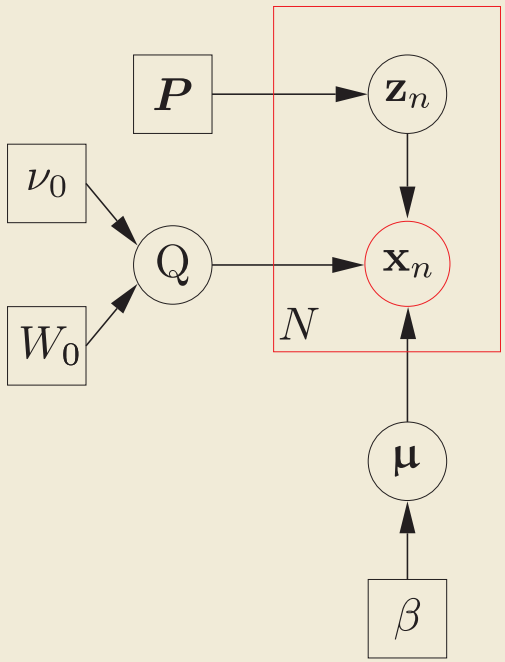
  where $\boldsymbol{\mu}_{1:K}$ and $Q_{1:K}$ indicate the collections $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ and $\{Q_1, \ldots, Q_K\}$, respectively.

- Furthermore, observe that the conditional pdf of the observations can now be written as

$$p(\mathcal{X}|\mathcal{Z}, \boldsymbol{\mu}_{1:K}, Q_{1:K}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, Q_k^{-1})\right)^{z_{n_k}}.$$

A Variational Bayesian Approach to Gaussian Mixture Modeling

- The figure below shows the graphical model that corresponds to the previous set up:



An Example

- The purpose of this example is to demonstrate the power of the variational Bayesian method for mixture modeling compared to the standard EM algorithm. Five clusters of data were generated using a corresponding number of Gaussians. The parameters used for each one of these Gaussians were:

$$\boldsymbol{\mu}_1 = [-2.5, 2.5]^T, \quad \boldsymbol{\mu}_2 = [-4.0, -2.0]^T \quad \boldsymbol{\mu}_3 = [2.0, -1.0]^T$$
$$\boldsymbol{\mu}_4 = [0.1, 0.2]^T, \quad \boldsymbol{\mu}_5 = [3.0, 3.0]^T$$

and

$$\Sigma_1 = \begin{bmatrix} 0.5 & 0.081 \\ 0.081 & 0.7 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 0.4 & 0.02 \\ 0.002 & 0.3 \end{bmatrix}$$
$$\Sigma_3 = \begin{bmatrix} 0.6 & 0.531 \\ 0.531 & 0.9 \end{bmatrix}$$
$$\Sigma_4 = \begin{bmatrix} 0.5 & 0.22 \\ 0.22 & 0.8 \end{bmatrix} \quad \Sigma_5 = \begin{bmatrix} 0.88 & 0.2 \\ 0.2 & 0.22 \end{bmatrix}$$

- Prior to running the algorithms, we assumed that we do not know the exact number of mixtures, so a number of $K = 25$ clusters was used; that is, a much larger number than the true one.

An Example

- The purpose of this example is to demonstrate the power of the variational Bayesian method for mixture modeling compared to the standard EM algorithm. Five clusters of data were generated using a corresponding number of Gaussians. The parameters used for each one of these Gaussians were:

$$\boldsymbol{\mu}_1 = [-2.5, 2.5]^T, \quad \boldsymbol{\mu}_2 = [-4.0, -2.0]^T \quad \boldsymbol{\mu}_3 = [2.0, -1.0]^T$$
$$\boldsymbol{\mu}_4 = [0.1, 0.2]^T, \quad \boldsymbol{\mu}_5 = [3.0, 3.0]^T$$

and

$$\Sigma_1 = \begin{bmatrix} 0.5 & 0.081 \\ 0.081 & 0.7 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 0.4 & 0.02 \\ 0.002 & 0.3 \end{bmatrix}$$
$$\Sigma_3 = \begin{bmatrix} 0.6 & 0.531 \\ 0.531 & 0.9 \end{bmatrix}$$
$$\Sigma_4 = \begin{bmatrix} 0.5 & 0.22 \\ 0.22 & 0.8 \end{bmatrix} \quad \Sigma_5 = \begin{bmatrix} 0.88 & 0.2 \\ 0.2 & 0.22 \end{bmatrix}$$

- Prior to running the algorithms, we assumed that we do not know the exact number of mixtures, so a number of $K = 25$ clusters was used; that is, a much larger number than the true one.

An Example

- The purpose of this example is to demonstrate the power of the variational Bayesian method for mixture modeling compared to the standard EM algorithm. Five clusters of data were generated using a corresponding number of Gaussians. The parameters used for each one of these Gaussians were:

$$\boldsymbol{\mu}_1 = [-2.5, 2.5]^T, \quad \boldsymbol{\mu}_2 = [-4.0, -2.0]^T \quad \boldsymbol{\mu}_3 = [2.0, -1.0]^T$$
$$\boldsymbol{\mu}_4 = [0.1, 0.2]^T, \quad \boldsymbol{\mu}_5 = [3.0, 3.0]^T$$

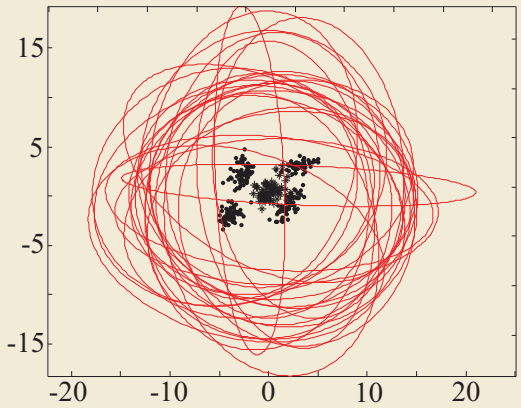
and

$$\Sigma_1 = \begin{bmatrix} 0.5 & 0.081 \\ 0.081 & 0.7 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 0.4 & 0.02 \\ 0.002 & 0.3 \end{bmatrix}$$
$$\Sigma_3 = \begin{bmatrix} 0.6 & 0.531 \\ 0.531 & 0.9 \end{bmatrix}$$
$$\Sigma_4 = \begin{bmatrix} 0.5 & 0.22 \\ 0.22 & 0.8 \end{bmatrix} \quad \Sigma_5 = \begin{bmatrix} 0.88 & 0.2 \\ 0.2 & 0.22 \end{bmatrix}$$

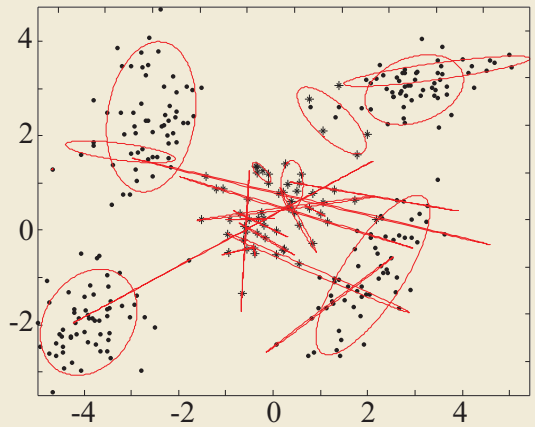
- Prior to running the algorithms, we assumed that we do not know the exact number of mixtures, so a number of $K = 25$ clusters was used; that is, a much larger number than the true one.

An Example

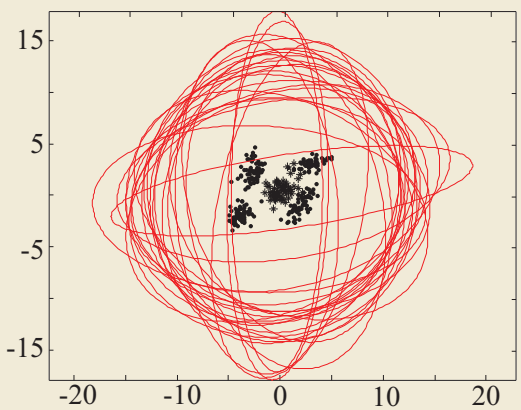
- a) The initial (25) Gaussians for the EM algorithm. b) The final clusters obtained after convergence by the EM algorithm. c) The initial (25) Gaussians for the variational EM. d) The final Gaussians obtained by the variational EM, after convergence. All the curves correspond to the 80% probability regions. Observe that the variational EM identifies the five clusters associated with the data; the rest of the mixtures correspond to zero probability weights.



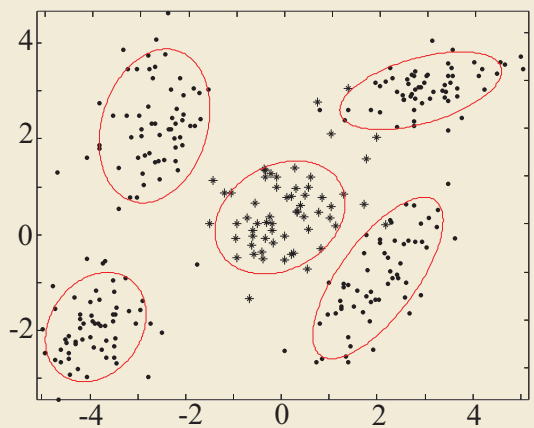
(a)



(b)



(c)



(d)

The Relevance Vector Machine Framework

- Let us now consider a specific regression model, i.e.,

$$\mathrm{y}(\mathbf{x}) = \theta_0 + \sum_{k=1}^{N} \theta_k \kappa(\mathbf{x}, \boldsymbol{x_k}) + \eta.$$

  In other words, the general regression model is considered for $K = N + 1$, where N is the number of observations and

$$\phi_k(\mathbf{x}) = \kappa(\mathbf{x}, \boldsymbol{x}_k),$$

  where $\kappa(\cdot, \cdot)$ is a kernel function, centered at the input observation points, $\boldsymbol{x}_k,\ k = 1, 2, \ldots, N$. Thus, the number of parameters becomes **equal** (plus one) to the number of training points.

- Due to the excessively large number of parameters, to be estimated, one has to resort to sparsity enforcing techniques, e.g., ARD via the variational Bayesian path. We have already done it for regression in the last example.

The Relevance Vector Machine Framework

- Let us now consider a specific regression model, i.e.,

$$\mathrm{y}(\mathbf{x}) = \theta_0 + \sum_{k=1}^{N} \theta_k \kappa(\mathbf{x}, \boldsymbol{x_k}) + \eta.$$

  In other words, the general regression model is considered for $K = N + 1$, where N is the number of observations and

$$\phi_k(\mathbf{x}) = \kappa(\mathbf{x}, \boldsymbol{x}_k),$$

  where $\kappa(\cdot, \cdot)$ is a kernel function, centered at the input observation points, $\boldsymbol{x}_k,\ k = 1, 2, \ldots, N$. Thus, the number of parameters becomes **equal** (plus one) to the number of training points.

- Due to the excessively large number of parameters, to be estimated, one has to resort to sparsity enforcing techniques, e.g., ARD via the variational Bayesian path. We have already done it for regression in the last example.

Adopting the Logistic Regression Model for Classification

- Our interest now turns on how to treat such "large" models in the context of classification. In analogy to the support vector machines (SVM), such models have become known as Relevance Vector Machines.
- The starting point is that, given the value of a measured feature vector, x, classification is performed according to the sign of the discriminant function, namely

$$f(\boldsymbol{x}) := \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) := \theta_0 + \sum_{k=1}^{N} \theta_k \phi_k(\boldsymbol{x}).$$

  The goal is to obtain an estimate of the parameters θ in the Bayesian framework.

- In this vein, the logistic regression model will be adopted.

Adopting the Logistic Regression Model for Classification

- Our interest now turns on how to treat such "large" models in the context of classification. In analogy to the support vector machines (SVM), such models have become known as Relevance Vector Machines.
- The starting point is that, given the value of a measured feature vector, $\boldsymbol{x}$, classification is performed according to the sign of the discriminant function, namely

$$f(\boldsymbol{x}) := \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) := \theta_0 + \sum_{k=1}^{N} \theta_k \phi_k(\boldsymbol{x}).$$

  The goal is to obtain an estimate of the parameters $\boldsymbol{\theta}$ in the Bayesian framework.

- In this vein, the logistic regression model will be adopted.

Adopting the Logistic Regression Model for Classification

- Our interest now turns on how to treat such "large" models in the context of classification. In analogy to the support vector machines (SVM), such models have become known as Relevance Vector Machines.
- The starting point is that, given the value of a measured feature vector, $\boldsymbol{x}$, classification is performed according to the sign of the discriminant function, namely

$$f(\boldsymbol{x}) := \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) := \theta_0 + \sum_{k=1}^{N} \theta_k \phi_k(\boldsymbol{x}).$$

  The goal is to obtain an estimate of the parameters $\boldsymbol{\theta}$ in the Bayesian framework.
- In this vein, the logistic regression model will be adopted.

Adopting the Logistic Regression Model for Classification

- According to this model and for a two-class (ω_1, ω_2) classification task, the posterior probabilities, as required by the Bayesian classifier, are modeled as

$$P(\omega_1|\boldsymbol{x}) = \frac{1}{1+\exp\Big(-\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x})\Big)}, \quad P(\omega_2|\boldsymbol{x}) = 1 - P(\omega_1|\boldsymbol{x}).$$

- The function $\sigma(t) := \frac{1}{1+\exp(-t)}$, is known as the logistic sigmoid link.
- Considering the training set $(y_n, \boldsymbol{x}_n)$, $\boldsymbol{x}_n \in \mathbb{R}^l$ and $y_n \in \{0, 1\}$, and adopting a Bernoulli distribution for $P(y|\boldsymbol{x})$, the respective likelihood function can be defined as

$$P(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \Big(\sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)^{y_n} \Big(1 - \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)^{1-y_n},$$

which is the counterpart of (26) for the regression case.

Adopting the Logistic Regression Model for Classification

- According to this model and for a two-class (ω_1, ω_2) classification task, the posterior probabilities, as required by the Bayesian classifier, are modeled as

$$P(\omega_1|\boldsymbol{x}) = \frac{1}{1+\exp\Big(-\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x})\Big)}, \quad P(\omega_2|\boldsymbol{x}) = 1 - P(\omega_1|\boldsymbol{x}).$$

- The function $\sigma(t) := \frac{1}{1+\exp(-t)}$, is known as the logistic sigmoid link.

- Considering the training set $(y_n, \boldsymbol{x}_n)$, $\boldsymbol{x}_n \in \mathbb{R}^l$ and $y_n \in \{0, 1\}$, and adopting a Bernoulli distribution for $P(y|\boldsymbol{x})$, the respective likelihood function can be defined as

$$P(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \Big(\sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)^{y_n} \Big(1 - \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)^{1-y_n},$$

which is the counterpart of (26) for the regression case.

Adopting the Logistic Regression Model for Classification

- According to this model and for a two-class (ω_1, ω_2) classification task, the posterior probabilities, as required by the Bayesian classifier, are modeled as

$$P(\omega_1|\boldsymbol{x}) = \frac{1}{1+\exp\Big(-\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x})\Big)}, \quad P(\omega_2|\boldsymbol{x}) = 1 - P(\omega_1|\boldsymbol{x}).$$

- The function $\sigma(t) := \frac{1}{1+\exp(-t)}$, is known as the logistic sigmoid link.

- Considering the training set $(y_n, \boldsymbol{x}_n)$, $\boldsymbol{x}_n \in \mathbb{R}^l$ and $y_n \in \{0, 1\}$, and adopting a Bernoulli distribution for $P(y|\boldsymbol{x})$, the respective likelihood function can be defined as

$$P(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \Big(\sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)^{y_n} \Big(1 - \sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)^{1-y_n},$$

which is the counterpart of (26) for the regression case.

Adopting the Logistic Regression Model for Classification

- In line with the ARD philosophy, we adopt the following Gaussian prior

$$p(\boldsymbol{\theta};\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, A^{-1}), \quad A := \text{diag}\{\alpha_0, \ldots, \alpha_N\}$$

- The goal now is to maximize the Type II log-likelihood with respect to the unknown parameters, $\boldsymbol{\alpha}$. However, $p(\boldsymbol{y}|\boldsymbol{\theta})$ is no more Gaussian and marginalizing out θ cannot be carried out analytically.

- To this end, the Laplacian approximation will be employed.

Adopting the Logistic Regression Model for Classification

- In line with the ARD philosophy, we adopt the following Gaussian prior

$$p(\boldsymbol{\theta};\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, A^{-1}), \quad A := \text{diag}\{\alpha_0, \ldots, \alpha_N\}$$

- The goal now is to maximize the Type II log-likelihood with respect to the unknown parameters, $\boldsymbol{\alpha}$. However, $p(\boldsymbol{y}|\boldsymbol{\theta})$ is no more Gaussian and marginalizing out θ cannot be carried out analytically.

- To this end, the Laplacian approximation will be employed.

Adopting the Logistic Regression Model for Classification

- The Laplacian approximation, around the MAP estimate, is employed and the following stepwise procedure is adopted:
  - 1) Assuming $\boldsymbol{\alpha}$ to be currently available, maximize with respect to $\boldsymbol{\theta}$ the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\alpha}) = \frac{P(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{P(\boldsymbol{y}|\boldsymbol{\alpha})}.$$

Defining $\boldsymbol{s} = [s_1, s_2, \ldots, s_N]^T$, $s_n := \sigma(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n))$, we finally obtain

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = A^{-1}\Phi^T(\boldsymbol{y}-\boldsymbol{s}) \quad A := \text{diag}\{\alpha_0, \alpha_2, \ldots, \alpha_N\}.$$

  - 2) Use $\hat{\theta}_{\text{MAP}}$ and the Laplace approximation method to approximate $p(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\alpha})$ by a Gaussian centered at $\hat{\boldsymbol{\theta}}_{\text{MAP}}$, whose covariance matrix turns out to be

$$\Sigma^{-1} = (\Phi^T T\Phi + A),$$

where $T := \text{diag}\{t_1, t_2, \ldots, t_N\}$ and

$$t_n = \sigma\Big(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\Big)\Big(1-\sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}}$$

Adopting the Logistic Regression Model for Classification

- The Laplacian approximation, around the MAP estimate, is employed and the following stepwise procedure is adopted:
  - 1) Assuming $\boldsymbol{\alpha}$ to be currently available, maximize with respect to $\boldsymbol{\theta}$ the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\alpha}) = \frac{P(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{P(\boldsymbol{y}|\boldsymbol{\alpha})}.$$

    Defining $\boldsymbol{s} = [s_1, s_2, \ldots, s_N]^T$, $s_n := \sigma(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n))$, we finally obtain

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = A^{-1}\Phi^T(\boldsymbol{y}-\boldsymbol{s}) \quad A := \text{diag}\{\alpha_0, \alpha_2, \ldots, \alpha_N\}.$$

  - 2) Use $\hat{\theta}_{\text{MAP}}$ and the Laplace approximation method to approximate $p(\theta|\boldsymbol{y},\boldsymbol{\alpha})$ by a Gaussian centered at $\hat{\theta}_{\text{MAP}}$, whose covariance matrix turns out to be

$$\Sigma^{-1} = (\Phi^T T\Phi + A),$$

    where $T := \text{diag}\{t_1, t_2, \ldots, t_N\}$ and

$$t_n = \sigma\Big(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\Big)\Big(1-\sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}}$$

Adopting the Logistic Regression Model for Classification

- The Laplacian approximation, around the MAP estimate, is employed and the following stepwise procedure is adopted:
  - 1) Assuming $\boldsymbol{\alpha}$ to be currently available, maximize with respect to $\boldsymbol{\theta}$ the posterior

$$p(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\alpha}) = \frac{P(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})}{P(\boldsymbol{y}|\boldsymbol{\alpha})}.$$

    Defining $\boldsymbol{s} = [s_1, s_2, \ldots, s_N]^T$, $s_n := \sigma(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n))$, we finally obtain

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = A^{-1}\Phi^T(\boldsymbol{y}-\boldsymbol{s}) \quad A := \text{diag}\{\alpha_0, \alpha_2, \ldots, \alpha_N\}.$$

  - 2) Use $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ and the Laplace approximation method to approximate $p(\boldsymbol{\theta}|\boldsymbol{y},\boldsymbol{\alpha})$ by a Gaussian centered at $\hat{\boldsymbol{\theta}}_{\text{MAP}}$, whose covariance matrix turns out to be

$$\Sigma^{-1} = (\Phi^T T\Phi + A),$$

    where $T := \text{diag}\{t_1, t_2, \ldots, t_N\}$ and

$$t_n = \sigma\Big(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\Big)\Big(1-\sigma\left(\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x}_n)\right)\Big)\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\text{MAP}}}$$

Adopting the Logistic Regression Model for Classification

- (continued)
  - 3) Having obtained $\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}$ and computed Σ, we obtain the following approximation for the Type-II likelihood,

$$P(\boldsymbol{y}|\boldsymbol{\alpha}) = P(\boldsymbol{y}|\hat{\boldsymbol{\theta}}_{\mathrm{MAP}})p(\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}|\boldsymbol{\alpha})(2\pi)^{\frac{N}{2}}|\Sigma|^{1/2},$$

    whose maximization w.r. to $\boldsymbol{\alpha}$ finally leads to the following iterative solution (starting from some initial values),

$$\alpha_k^{(\mathrm{new})} = \frac{1 - \alpha_k^{(\mathrm{old})}\Sigma_{kk}^{(\mathrm{old})}}{(\theta_{\mathrm{MAP},k}^{(\mathrm{old})})^2}.$$

    The procedure continues till a convergence criterion is met.

RVM: A Simulation Example

- In this example, the performance of the RVM is tested in the context of a two-class two-dimensional classification task. The data set comprises $N = 150$ points uniformly distributed in the region $[-5, 5] \times [-5, 5]$. For each point, $\boldsymbol{x}_n = [x_{n,1},\ x_{n,2}]^T$, $n = 1, 2, \ldots, N$, we compute

$$y_n = 0.5x_{n,1}^3 + 0.5x_{n,1}^2 + 0.5x_{n,1} + 1 + \eta,$$

where η stands for zero-mean Gaussian noise of variance $\sigma_\eta^2 = 4$. The point is assigned to either of the two classes, depending on which side of the graph of the function

$$f(x) = 0.5x^3 + 0.5x^2 + 0.5x + 1,$$

in the two-dimensional space, y_n lies. That is, if $y_n > f(x_{n1})$ the point is assigned to class ω_1 otherwise is assigned to class ω_2.

- The Gaussian kernel was used with $\sigma^2 = 3$, which we found to give the best results.

RVM: A Simulation Example

- In this example, the performance of the RVM is tested in the context of a two-class two-dimensional classification task. The data set comprises $N = 150$ points uniformly distributed in the region $[-5, 5] \times [-5, 5]$. For each point, $\boldsymbol{x}_n = [x_{n,1},\ x_{n,2}]^T$, $n = 1, 2, \ldots, N$, we compute

$$y_n = 0.5x_{n,1}^3 + 0.5x_{n,1}^2 + 0.5x_{n,1} + 1 + \eta,$$

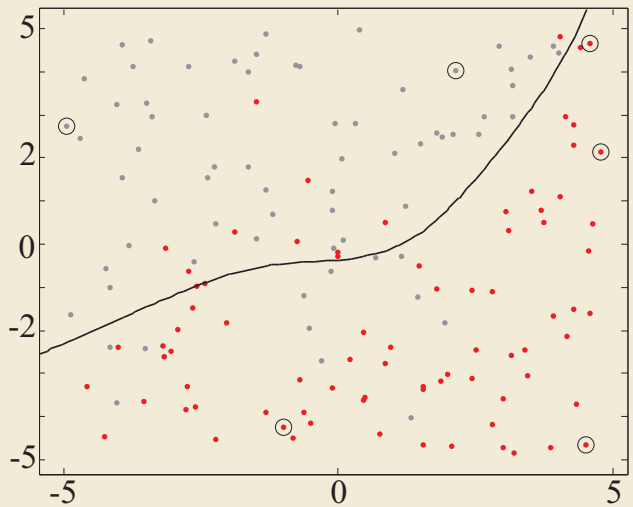
  where η stands for zero-mean Gaussian noise of variance $\sigma_\eta^2 = 4$. The point is assigned to either of the two classes, depending on which side of the graph of the function

$$f(x) = 0.5x^3 + 0.5x^2 + 0.5x + 1,$$

  in the two-dimensional space, y_n lies. That is, if $y_n > f(x_{n1})$ the point is assigned to class ω_1 otherwise is assigned to class ω_2.
- The Gaussian kernel was used with $\sigma^2 = 3$, which we found to give the best results.

RVM: A Simulation Example

- The figure below shows the resulting decision curve that results from the RVM method and classifies the points of the the red/gray classes.
- Six points, which have been encircled, are the surviving relevance vectors. The rest of the parameters come out to almost zero values, due to the sparsifying power associated with the underlying ARD philosophy.
- Note that, the number of support vectors surviving is significantly less compared to the case of SVM, treated in Chapter 11.



RVM vs SVM

- Compared to SVM (SVR), the RVM machinery presents advantages and disadvantages.
- The SVM approach results in a single solution, due to the convexity of the associated cost functions. In contrast, RVM builds upon non-convex cost. Thus, one may have to run the optimization algorithm a number of times, starting each time from different initial conditions, since a non-convex problem can be trapped in a local minimum.

RVM vs SVM

- Compared to SVM (SVR), the RVM machinery presents advantages and disadvantages.
- The SVM approach results in a single solution, due to the convexity of the associated cost functions. In contrast, RVM builds upon non-convex cost. Thus, one may have to run the optimization algorithm a number of times, starting each time from different initial conditions, since a non-convex problem can be trapped in a local minimum.

RVM vs SVM

- Concerning complexity, the RVM amounts to $O(N^3)$ operations per iteration. In contrast, the complexity for solving the SVM scales from linear to (approximately) quadratic. Also the memory for the RVM exhibits a $O(N^2)$ dependence as opposed to a linear dependence to the SVM case. Finally, RVMs need, in general, longer training times to converge, compared to SVMs, for similar error rates.
- A fast RVM algorithm has also been developed, that operates in a constructive manner, until all relevant basis functions (for which the associated weights are nonzero) have been included. If M denotes the number of relevant terms, the complexity amounts to $O(M^3)$, which for $M << N$ is more efficient than the original RVM.
- The main advantage of the RVMs is that, in general, they result in sparser solutions compared to the SVMs, for similar levels of generalization errors.

RVM vs SVM

- Concerning complexity, the RVM amounts to $O(N^3)$ operations per iteration. In contrast, the complexity for solving the SVM scales from linear to (approximately) quadratic. Also the memory for the RVM exhibits a $O(N^2)$ dependence as opposed to a linear dependence to the SVM case. Finally, RVMs need, in general, longer training times to converge, compared to SVMs, for similar error rates.
- A fast RVM algorithm has also been developed, that operates in a constructive manner, until all relevant basis functions (for which the associated weights are nonzero) have been included. If M denotes the number of relevant terms, the complexity amounts to $O(M^3)$, which for $M << N$ is more efficient than the original RVM.
- The main advantage of the RVMs is that, in general, they result in sparser solutions compared to the SVMs, for similar levels of generalization errors.

RVM vs SVM

- Concerning complexity, the RVM amounts to $O(N^3)$ operations per iteration. In contrast, the complexity for solving the SVM scales from linear to (approximately) quadratic. Also the memory for the RVM exhibits a $O(N^2)$ dependence as opposed to a linear dependence to the SVM case. Finally, RVMs need, in general, longer training times to converge, compared to SVMs, for similar error rates.
- A fast RVM algorithm has also been developed, that operates in a constructive manner, until all relevant basis functions (for which the associated weights are nonzero) have been included. If M denotes the number of relevant terms, the complexity amounts to $O(M^3)$, which for $M << N$ is more efficient than the original RVM.
- The main advantage of the RVMs is that, in general, they result in sparser solutions compared to the SVMs, for similar levels of generalization errors.

Sparsity: The Spike and Slab Method

- Let us consider our familiar regression model,

$$\mathrm{y} = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) + \eta = \sum_{k=0}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \eta.$$

- A new set of auxiliary binary indicator variables are introduced, $\mathrm{s}_k \in \{0, 1\},\ k = 0, 1, \ldots, K-1$. Let, also, the prior imposed on $\boldsymbol{\theta}$, be a Gaussian, $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \sigma^2 I)$.

- As the name suggests, the indicator variables control the presence or not of a parameter in the above summation. For example, if $s_k = 1$ the corresponding parameter, θ_k, is present and if $s_k = 0$ then θ_k is removed; this is the way that sparsity is imposed onto the model.

Sparsity: The Spike and Slab Method

- Let us consider our familiar regression model,

$$\mathrm{y} = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) + \eta = \sum_{k=0}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \eta.$$

- A new set of auxiliary binary indicator variables are introduced, $\mathrm{s}_k \in \{0, 1\},\ k = 0, 1, \ldots, K-1$. Let, also, the prior imposed on $\boldsymbol{\theta}$, be a Gaussian, $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \sigma^2 I)$.

- As the name suggests, the indicator variables control the presence or not of a parameter in the above summation. For example, if $s_k = 1$ the corresponding parameter, θ_k, is present and if $s_k = 0$ then θ_k is removed; this is the way that sparsity is imposed onto the model.

Sparsity: The Spike and Slab Method

- Let us consider our familiar regression model,

$$\mathrm{y} = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) + \eta = \sum_{k=0}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \eta.$$

- A new set of auxiliary binary indicator variables are introduced, $\mathrm{s}_k \in \{0, 1\},\ k = 0, 1, \ldots, K-1$. Let, also, the prior imposed on $\boldsymbol{\theta}$, be a Gaussian, $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \sigma^2 I)$.

- As the name suggests, the indicator variables control the presence or not of a parameter in the above summation. For example, if $s_k = 1$ the corresponding parameter, θ_k, is present and if $s_k = 0$ then θ_k is removed; this is the way that sparsity is imposed onto the model.

Sparsity: The Spike and Slab Method

- To this end, a joint Bernoulli prior distribution is adopted for the indicator variables, i.e.,

$$P(\boldsymbol{s}) = \prod_{k=0}^{K-1} p^{s_k}(1-p)^{1-s_k},$$

  where the parameter $0 \leq p \leq 1$ specifies a prior level of sparsity.

- This turns out to be equivalent with adopting the following prior on the parameters,

$$p(\boldsymbol{\theta}) = \prod_{k=0}^{K-1} \Big(s_k \mathcal{N}(\theta_k | 0, \sigma^2) + (1 - s_k)\delta(\theta_k) \Big)$$

- The corresponding posterior is not Gaussian and its computation can be done by approximate inference techniques, such as variational or Monte Carlo.

Sparsity: The Spike and Slab Method

- To this end, a joint Bernoulli prior distribution is adopted for the indicator variables, i.e.,

$$P(\boldsymbol{s}) = \prod_{k=0}^{K-1} p^{s_k}(1-p)^{1-s_k},$$

  where the parameter $0 \leq p \leq 1$ specifies a prior level of sparsity.

- This turns out to be equivalent with adopting the following prior on the parameters,

$$p(\boldsymbol{\theta}) = \prod_{k=0}^{K-1} \Big(s_k \mathcal{N}(\theta_k | 0, \sigma^2) + (1 - s_k)\delta(\theta_k) \Big)$$

- The corresponding posterior is not Gaussian and its computation can be done by approximate inference techniques, such as variational or Monte Carlo.

Sparsity: The Spike and Slab Method

- To this end, a joint Bernoulli prior distribution is adopted for the indicator variables, i.e.,

$$P(\boldsymbol{s}) = \prod_{k=0}^{K-1} p^{s_k}(1-p)^{1-s_k},$$

  where the parameter $0 \leq p \leq 1$ specifies a prior level of sparsity.

- This turns out to be equivalent with adopting the following prior on the parameters,

$$p(\boldsymbol{\theta}) = \prod_{k=0}^{K-1} \Big(s_k \mathcal{N}(\theta_k | 0, \sigma^2) + (1 - s_k)\delta(\theta_k) \Big)$$

- The corresponding posterior is not Gaussian and its computation can be done by approximate inference techniques, such as variational or Monte Carlo.

Gaussian Processes

- The emphasis now turns in nonparametric models. The main assumption is that the underlying functions, which express the input-output dependence, lie in RKH spaces. Here, we are approaching such models via Bayesian arguments.
- Let us recall the nonlinear regression task, i.e.,

$$\mathrm{y} = \theta_0 + \sum_{k=1}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \eta = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) + \eta,$$

where the parameters, $\boldsymbol{\theta}$, are treated as a random vector. Let us define,

$$\mathrm{f}(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}),$$

where $\mathrm{f}(\boldsymbol{x})$ is a random process.

Gaussian Processes

- The emphasis now turns in nonparametric models. The main assumption is that the underlying functions, which express the input-output dependence, lie in RKH spaces. Here, we are approaching such models via Bayesian arguments.
- Let us recall the nonlinear regression task, i.e.,

$$\mathrm{y} = \theta_0 + \sum_{k=1}^{K-1} \theta_k \phi_k(\boldsymbol{x}) + \eta = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}) + \eta,$$

where the parameters, $\boldsymbol{\theta}$, are treated as a random vector. Let us define,

$$\mathrm{f}(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}),$$

where $\mathrm{f}(\boldsymbol{x})$ is a random process.

Gaussian Processes

- The idea, which spans this section, is to work directly on $\mathrm{f}(\boldsymbol{x})$ instead on the indirect approach of modeling it via a set of parameters, $\boldsymbol{\theta}$. That is, we will treat the more general nonlinear regression task, expressed as

$$\mathrm{y} = \mathrm{f}(\boldsymbol{x}) + \eta.$$

- We will focus on a specific class of random processes, known as Gaussian processes.
- Definition: A random process, $\mathrm{f}(\boldsymbol{x})$, is called a Gaussian process (GP) iff for any finite number of points, $\boldsymbol{x}_{(1)}, ..., \boldsymbol{x}_{(N)}$, the respective joint pdf, $p\big(f(\boldsymbol{x}_{(1)}), ...f(\boldsymbol{x}_{(N)})\big)$, is Gaussian.

Gaussian Processes

- The idea, which spans this section, is to work directly on $\mathrm{f}(\boldsymbol{x})$ instead on the indirect approach of modeling it via a set of parameters, $\boldsymbol{\theta}$. That is, we will treat the more general nonlinear regression task, expressed as

$$\mathrm{y} = \mathrm{f}(\boldsymbol{x}) + \eta.$$

- We will focus on a specific class of random processes, known as Gaussian processes.
- Definition: A random process, $\mathrm{f}(\boldsymbol{x})$, is called a Gaussian process (GP) iff for any finite number of points, $\boldsymbol{x}_{(1)}, ..., \boldsymbol{x}_{(N)}$, the respective joint pdf, $p\big(f(\boldsymbol{x}_{(1)}), ...f(\boldsymbol{x}_{(N)})\big)$, is Gaussian.

Gaussian Processes

- The idea, which spans this section, is to work directly on $\mathrm{f}(\boldsymbol{x})$ instead on the indirect approach of modeling it via a set of parameters, $\boldsymbol{\theta}$. That is, we will treat the more general nonlinear regression task, expressed as

$$\mathrm{y} = \mathrm{f}(\boldsymbol{x}) + \eta.$$

- We will focus on a specific class of random processes, known as Gaussian processes.
- Definition: A random process, $\mathrm{f}(\boldsymbol{x})$, is called a Gaussian process (GP) iff for any finite number of points, $\boldsymbol{x}_{(1)}, ..., \boldsymbol{x}_{(N)}$, the respective joint pdf, $p\big(f(\boldsymbol{x}_{(1)}), ...f(\boldsymbol{x}_{(N)})\big)$, is Gaussian.

Gaussian Processes

- We know that a set of jointly Gaussian distributed random variables are fully described by the respective mean value and the covariance matrix. In a similar spirit, a Gaussian process is fully determined by its **mean value** and its **covariance function**, i.e.,

$$\mu_x = \mathbb{E}\left[\mathrm{f}(\boldsymbol{x})\right], \quad \mathrm{cov}_f(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[(\mathrm{f}(\boldsymbol{x}) - \mu_x)(\mathrm{f}(\boldsymbol{x}') - \mu_{x'})\right].$$

- A Gaussian process is said to be stationary if $\mu_x = \mu$ and its covariance function is of the form,

$$\mathrm{cov}_f(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{cov}_f(\boldsymbol{x} - \boldsymbol{x}').$$

- In addition, if $\mathrm{cov}_f(\cdot, \cdot)$ depends on the magnitude of the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$, i.e., $(\|\boldsymbol{x} - \boldsymbol{x}'\|)$, the Gaussian process is called homogeneous. From now on, we will assume $\mu_x = 0$.

Gaussian Processes

- We know that a set of jointly Gaussian distributed random variables are fully described by the respective mean value and the covariance matrix. In a similar spirit, a Gaussian process is fully determined by its **mean value** and its **covariance function**, i.e.,

$$\mu_x = \mathbb{E}\left[\mathrm{f}(\boldsymbol{x})\right], \quad \mathrm{cov}_f(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[(\mathrm{f}(\boldsymbol{x}) - \mu_x)(\mathrm{f}(\boldsymbol{x}') - \mu_{x'})\right].$$

- A Gaussian process is said to be stationary if $\mu_x = \mu$ and its covariance function is of the form,

$$\mathrm{cov}_f(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{cov}_f(\boldsymbol{x} - \boldsymbol{x}').$$

- In addition, if $\mathrm{cov}_f(\cdot, \cdot)$ depends on the magnitude of the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$, i.e., $(\|\boldsymbol{x} - \boldsymbol{x}'\|)$, the Gaussian process is called homogeneous. From now on, we will assume $\mu_x = 0$.

Gaussian Processes

- We know that a set of jointly Gaussian distributed random variables are fully described by the respective mean value and the covariance matrix. In a similar spirit, a Gaussian process is fully determined by its **mean value** and its **covariance function**, i.e.,

$$\mu_x = \mathbb{E}\left[\mathrm{f}(\boldsymbol{x})\right], \quad \mathrm{cov}_f(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\left[(\mathrm{f}(\boldsymbol{x}) - \mu_x)(\mathrm{f}(\boldsymbol{x}') - \mu_{x'})\right].$$

- A Gaussian process is said to be stationary if $\mu_x = \mu$ and its covariance function is of the form,

$$\mathrm{cov}_f(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{cov}_f(\boldsymbol{x} - \boldsymbol{x}').$$

- In addition, if $\mathrm{cov}_f(\cdot, \cdot)$ depends on the magnitude of the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$, i.e., $(\|\boldsymbol{x} - \boldsymbol{x}'\|)$, the Gaussian process is called homogeneous. From now on, we will assume $\mu_x = 0$.

Covariance Functions and Kernels

- For any N and any collection of N points, $\boldsymbol{x}_{(1)}, ..., \boldsymbol{x}_{(N)}$, the respective covariance matrix is defined by,

$$\Sigma = \mathbb{E}[\mathbf{f}\mathbf{f}^T], \quad \text{where} \quad \mathbf{f} := [\mathrm{f}(\boldsymbol{x}_{(1)}), ..., \mathrm{f}(\boldsymbol{x}_{(N)})]^T,$$

with elements given by

$$[\Sigma]_{ij} = \mathrm{cov}_f(\boldsymbol{x}_{(i)}, \boldsymbol{x}_{(j)}), \quad i, j = 1, 2, ...N.$$

- Since Σ is a positive semidefinite matrix, this guarantees that the covariance function is a kernel function. To stress this out, from now on, we will use the notation

$$\mathrm{cov}_f(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}, \boldsymbol{x}'),$$

and the covariance matrix becomes the corresponding kernel matrix denoted as $\mathcal{K}$.

Covariance Functions and Kernels

- For any N and any collection of N points, $\boldsymbol{x}_{(1)}, ..., \boldsymbol{x}_{(N)}$, the respective covariance matrix is defined by,

$$\Sigma = \mathbb{E}[\mathbf{f}\mathbf{f}^T], \quad \text{where} \quad \mathbf{f} := [\mathrm{f}(\boldsymbol{x}_{(1)}), ..., \mathrm{f}(\boldsymbol{x}_{(N)})]^T,$$

  with elements given by

$$[\Sigma]_{ij} = \mathrm{cov}_f(\boldsymbol{x}_{(i)}, \boldsymbol{x}_{(j)}), \quad i, j = 1, 2, ...N.$$

- Since Σ is a positive semidefinite matrix, this guarantees that the covariance function is a kernel function. To stress this out, from now on, we will use the notation

$$\mathrm{cov}_f(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x}, \boldsymbol{x}'),$$

  and the covariance matrix becomes the corresponding kernel matrix denoted as $\mathcal{K}$.

Covariance Functions and Kernels

- A popular kernel, commonly used in practice, is the squared exponential or Gaussian kernel,

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2h^2}\right),$$

  where h determines the so-called length scale of the process.

- The smaller the value of h is, the larger the "statistical" similarity (stronger correlation) of two points having a distance $d = \|x - x'\|$ apart.

Covariance Functions and Kernels

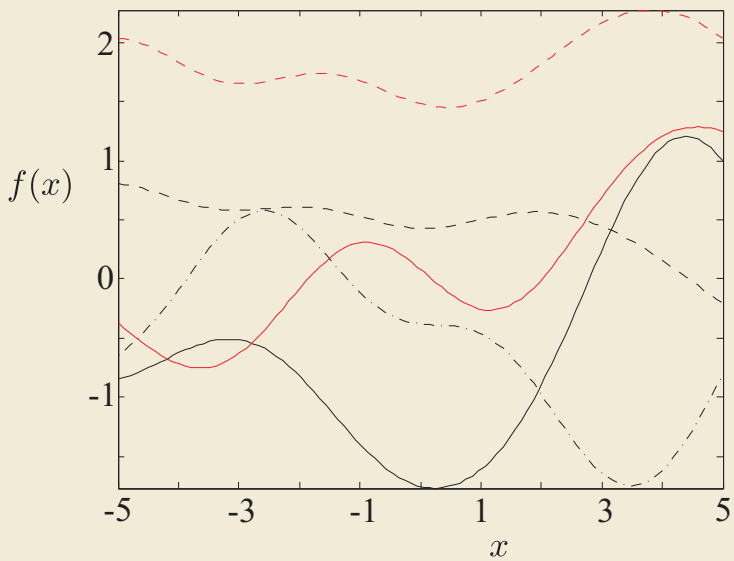
- A popular kernel, commonly used in practice, is the squared exponential or Gaussian kernel,

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2h^2}\right),$$

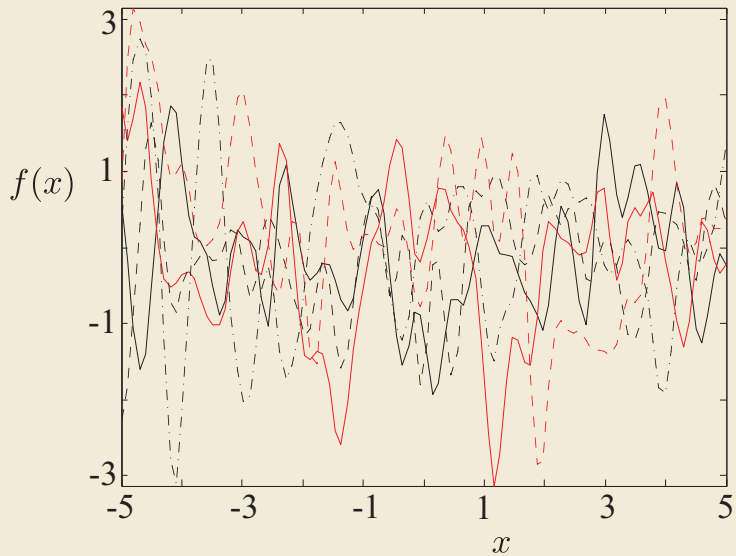
  where h determines the so-called length scale of the process.

- The smaller the value of h is, the larger the "statistical" similarity (stronger correlation) of two points having a distance $d = \|\boldsymbol{x} - \boldsymbol{x}'\|$ apart.

- Figure (a) shows examples of different realizations of a stationary Gaussian processes, using the Gaussian covariance kernel with $h = 2$ and Figure (b) for $h = 0.2$.



(a)



(b)

Covariance Functions and Kernels

- Let us assume that we are given a set $\mathcal{X}$ of input observations, $\mathcal{X} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_N\}$. Recall that the main goal in a Bayesian regression task is to obtain the two pdfs,

$$p(\boldsymbol{y}|\mathcal{X}) \text{ and } p(y|\boldsymbol{x}, \boldsymbol{y}, \mathcal{X}),$$

where,

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\eta}, \quad \mathbf{y} := [\mathrm{y}_1, ...\mathrm{y}_N]^T,$$

and

$$\mathrm{y} = \mathrm{f}(\boldsymbol{x}) + \eta.$$

- The first of the two pdfs is the joint probability density of the output variables, which are generated by input points in $\mathcal{X}$; the associated randomness is due to f **as well as** to the noise η.
- The second pdf refers to the prediction of the value of the output y, given the value of the input $\boldsymbol{x}$ and the training data $(y_n, \boldsymbol{x}_n), n = 1, 2, ..., N$. We will drop out $\mathcal{X}$ to unclutter notation.

Covariance Functions and Kernels

- Let us assume that we are given a set $\mathcal{X}$ of input observations, $\mathcal{X} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_N\}$. Recall that the main goal in a Bayesian regression task is to obtain the two pdfs,

$$p(\boldsymbol{y}|\mathcal{X}) \text{ and } p(y|\boldsymbol{x}, \boldsymbol{y}, \mathcal{X}),$$

where,

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\eta}, \quad \mathbf{y} := [\mathrm{y}_1, ...\mathrm{y}_N]^T,$$

and

$$\mathrm{y} = \mathrm{f}(\boldsymbol{x}) + \eta.$$

- The first of the two pdfs is the joint probability density of the output variables, which are generated by input points in $\mathcal{X}$; the associated randomness is due to f **as well as** to the noise η.
- The second pdf refers to the prediction of the value of the output y, given the value of the input $\boldsymbol{x}$ and the training data $(y_n, \boldsymbol{x}_n), n = 1, 2, ..., N$. We will drop out $\mathcal{X}$ to unclutter notation.

Covariance Functions and Kernels

- Let us assume that we are given a set $\mathcal{X}$ of input observations, $\mathcal{X} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_N\}$. Recall that the main goal in a Bayesian regression task is to obtain the two pdfs,

$$p(\boldsymbol{y}|\mathcal{X}) \text{ and } p(y|\boldsymbol{x}, \boldsymbol{y}, \mathcal{X}),$$

where,

$$\mathbf{y} = \mathbf{f} + \boldsymbol{\eta}, \quad \mathbf{y} := [\mathrm{y}_1, ...\mathrm{y}_N]^T,$$

and

$$\mathrm{y} = \mathrm{f}(\boldsymbol{x}) + \eta.$$

- The first of the two pdfs is the joint probability density of the output variables, which are generated by input points in $\mathcal{X}$; the associated randomness is due to f **as well as** to the noise η.
- The second pdf refers to the prediction of the value of the output y, given the value of the input $\boldsymbol{x}$ and the training data $(y_n, \boldsymbol{x}_n), n = 1, 2, ..., N$. We will drop out $\mathcal{X}$ to unclutter notation.

Covariance Functions and Kernels

- Assuming $\mathrm{f}(\cdot)$ to be a zero-mean Gaussian process, then $\mathbf{f}$ is jointly Gaussian with zero mean and covariance matrix $\mathcal{K}$, dictated by the covariance function/kernel $\kappa(\cdot,\cdot)$, i.e.,

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathcal{K}).$$

- Also, let $\boldsymbol{\eta}$ be of zero mean with covariance matrix Σ_η and independent of $\mathrm{f}(\cdot)$; without harming generality, let $\Sigma_\eta = \sigma_\eta^2 I$. Thus,

$$p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_\eta^2 I).$$

Then, following standard, by now, arguments, we obtain

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|0, \mathcal{K} + \sigma_\eta^2 I). \tag{32}$$

Covariance Functions and Kernels

- Assuming $\mathrm{f}(\cdot)$ to be a zero-mean Gaussian process, then $\mathbf{f}$ is jointly Gaussian with zero mean and covariance matrix $\mathcal{K}$, dictated by the covariance function/kernel $\kappa(\cdot,\cdot)$, i.e.,

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathcal{K}).$$

- Also, let $\boldsymbol{\eta}$ be of zero mean with covariance matrix Σ_η and independent of $\mathrm{f}(\cdot)$; without harming generality, let $\Sigma_\eta = \sigma_\eta^2 I$. Thus,

$$p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_\eta^2 I).$$

Then, following standard, by now, arguments, we obtain

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|0, \mathcal{K} + \sigma_\eta^2 I). \tag{32}$$

Covariance Functions and Kernels

- Assuming $\mathrm{f}(\cdot)$ to be a zero-mean Gaussian process, then $\mathbf{f}$ is jointly Gaussian with zero mean and covariance matrix $\mathcal{K}$, dictated by the covariance function/kernel $\kappa(\cdot,\cdot)$, i.e.,

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \mathcal{K}).$$

- Also, let $\boldsymbol{\eta}$ be of zero mean with covariance matrix Σ_η and independent of $\mathrm{f}(\cdot)$; without harming generality, let $\Sigma_\eta = \sigma_\eta^2 I$. Thus,

$$p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_\eta^2 I).$$

  Then, following standard, by now, arguments, we obtain

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|0, \mathcal{K} + \sigma_\eta^2 I). \tag{32}$$

Covariance Functions and Kernels

- To obtain $p(y|\boldsymbol{x}, \boldsymbol{y})$, we can use (32) and apply it recursively. To this end, it will also be useful to bring into the notation the number of available observations, N, explicitly and write

$$\mathbf{y}_{N+1} = \begin{bmatrix} \mathrm{y} \\ \mathbf{y}_N \end{bmatrix}, \ \mathbf{y}_N := [\mathrm{y}_1, ..., \mathrm{y}_N]^T.$$

- From (32), $\mathbf{y}_{N+1}$ follows a Gaussian distribution

$$p(\boldsymbol{y}_{N+1}|\mathbf{0}, \Sigma_{N+1}), \quad \text{where} \quad \Sigma_{N+1} := \mathcal{K}_{N+1} + \sigma_\eta^2 I_{N+1}.$$

  Then, from Bayes' theorem, we have

$$p(y|\boldsymbol{y}_N) = \frac{p(\boldsymbol{y}_{N+1})}{p(\boldsymbol{y}_N)}. \tag{33}$$

- However, since the **joint** pdf is Gaussian, the **conditional** in (33) is also Gaussian.

Covariance Functions and Kernels

- To obtain $p(y|\boldsymbol{x}, \boldsymbol{y})$, we can use (32) and apply it recursively. To this end, it will also be useful to bring into the notation the number of available observations, N, explicitly and write

$$\mathbf{y}_{N+1} = \begin{bmatrix} \mathrm{y} \\ \mathbf{y}_N \end{bmatrix}, \ \mathbf{y}_N := [\mathrm{y}_1, ..., \mathrm{y}_N]^T.$$

- From (32), $\mathbf{y}_{N+1}$ follows a Gaussian distribution

$$p(\boldsymbol{y}_{N+1}|\mathbf{0}, \Sigma_{N+1}), \quad \text{where} \quad \Sigma_{N+1} := \mathcal{K}_{N+1} + \sigma_\eta^2 I_{N+1}.$$

  Then, from Bayes' theorem, we have

$$p(y|\boldsymbol{y}_N) = \frac{p(\boldsymbol{y}_{N+1})}{p(\boldsymbol{y}_N)}. \tag{33}$$

- However, since the **joint** pdf is Gaussian, the **conditional** in (33) is also Gaussian.

Covariance Functions and Kernels

- To obtain $p(y|\boldsymbol{x}, \boldsymbol{y})$, we can use (32) and apply it recursively. To this end, it will also be useful to bring into the notation the number of available observations, N, explicitly and write

$$\mathbf{y}_{N+1} = \begin{bmatrix} \mathrm{y} \\ \mathbf{y}_N \end{bmatrix}, \ \mathbf{y}_N := [\mathrm{y}_1, ..., \mathrm{y}_N]^T.$$

- From (32), $\mathbf{y}_{N+1}$ follows a Gaussian distribution

$$p(\boldsymbol{y}_{N+1}|\mathbf{0}, \Sigma_{N+1}), \quad \text{where} \quad \Sigma_{N+1} := \mathcal{K}_{N+1} + \sigma_\eta^2 I_{N+1}.$$

  Then, from Bayes' theorem, we have

$$p(y|\boldsymbol{y}_N) = \frac{p(\boldsymbol{y}_{N+1})}{p(\boldsymbol{y}_N)}. \tag{33}$$

- However, since the **joint** pdf is Gaussian, the **conditional** in (33) is also Gaussian.

Covariance Functions and Kernels

- The respective mean and variance are computed by partitioning the matrix Σ_{N+1}, i.e.,

$$\Sigma_{N+1} = \begin{bmatrix} \kappa(\boldsymbol{x},\boldsymbol{x}) + \sigma_\eta^2, & \boldsymbol{\kappa}^T(\boldsymbol{x}) \\ \boldsymbol{\kappa}(\boldsymbol{x}), & \Sigma_N \end{bmatrix}, \ \boldsymbol{\kappa}(\boldsymbol{x}) := [\kappa(\boldsymbol{x},\boldsymbol{x}_1), ..., \kappa(\boldsymbol{x},\boldsymbol{x}_N)]^T,$$

and finally it turns out that,

$$\mu_y(\boldsymbol{x}) = \boldsymbol{\kappa}^T(\boldsymbol{x})\Sigma_N^{-1}\boldsymbol{y},$$
$$\sigma_y^2(\boldsymbol{x}) = \sigma_\eta^2 + \kappa(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{\kappa}^T(\boldsymbol{x})\Sigma_N^{-1}\boldsymbol{\kappa}(\boldsymbol{x}).$$

- Taking into account that $\Sigma_N = \mathcal{K}_N + \sigma_\eta^2 I$, note that $\mu_y(\boldsymbol{x})$ is identical to $\hat{y}$ obtained by the kernel ridge regression, for appropriate choices of the involved parameters (C and σ_η^2).
- The above formulas can be obtained from the linear case equations of the Bayesian learning, via the kernel trick.

Covariance Functions and Kernels

- The respective mean and variance are computed by partitioning the matrix Σ_{N+1}, i.e.,

$$\Sigma_{N+1} = \begin{bmatrix} \kappa(\boldsymbol{x},\boldsymbol{x}) + \sigma_\eta^2, & \boldsymbol{\kappa}^T(\boldsymbol{x}) \\ \boldsymbol{\kappa}(\boldsymbol{x}), & \Sigma_N \end{bmatrix}, \ \boldsymbol{\kappa}(\boldsymbol{x}) := [\kappa(\boldsymbol{x},\boldsymbol{x}_1), ..., \kappa(\boldsymbol{x},\boldsymbol{x}_N)]^T,$$

and finally it turns out that,

$$\mu_y(\boldsymbol{x}) = \boldsymbol{\kappa}^T(\boldsymbol{x})\Sigma_N^{-1}\boldsymbol{y},$$
$$\sigma_y^2(\boldsymbol{x}) = \sigma_\eta^2 + \kappa(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{\kappa}^T(\boldsymbol{x})\Sigma_N^{-1}\boldsymbol{\kappa}(\boldsymbol{x}).$$

- Taking into account that $\Sigma_N = \mathcal{K}_N + \sigma_\eta^2 I$, note that $\mu_y(\boldsymbol{x})$ is identical to $\hat{y}$ obtained by the kernel ridge regression, for appropriate choices of the involved parameters (C and σ_η^2).
- The above formulas can be obtained from the linear case equations of the Bayesian learning, via the kernel trick.

Covariance Functions and Kernels

- The respective mean and variance are computed by partitioning the matrix Σ_{N+1}, i.e.,

$$\Sigma_{N+1} = \begin{bmatrix} \kappa(\boldsymbol{x},\boldsymbol{x}) + \sigma_\eta^2, & \boldsymbol{\kappa}^T(\boldsymbol{x}) \\ \boldsymbol{\kappa}(\boldsymbol{x}), & \Sigma_N \end{bmatrix}, \ \boldsymbol{\kappa}(\boldsymbol{x}) := [\kappa(\boldsymbol{x},\boldsymbol{x}_1), ..., \kappa(\boldsymbol{x},\boldsymbol{x}_N)]^T,$$

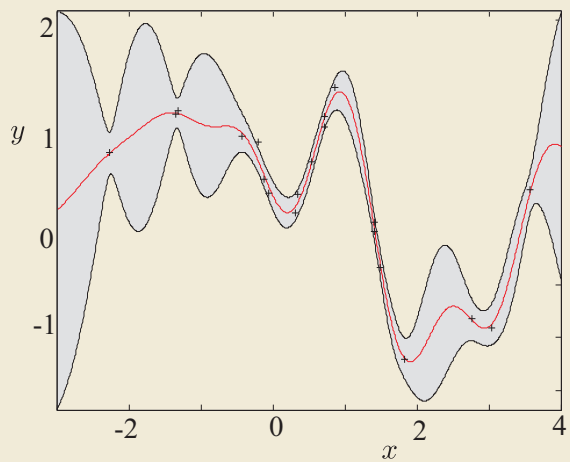
  and finally it turns out that,

$$\mu_y(\boldsymbol{x}) = \boldsymbol{\kappa}^T(\boldsymbol{x})\Sigma_N^{-1}\boldsymbol{y},$$
$$\sigma_y^2(\boldsymbol{x}) = \sigma_\eta^2 + \kappa(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{\kappa}^T(\boldsymbol{x})\Sigma_N^{-1}\boldsymbol{\kappa}(\boldsymbol{x}).$$

- Taking into account that $\Sigma_N = \mathcal{K}_N + \sigma_\eta^2 I$, note that $\mu_y(\boldsymbol{x})$ is identical to $\hat{y}$ obtained by the kernel ridge regression, for appropriate choices of the involved parameters (C and σ_η^2).
- The above formulas can be obtained from the linear case equations of the Bayesian learning, via the kernel trick.

Gaussian Processes for Regression: An Example

- A number of $N = 20$ points are randomly sampled from a realization of a Gaussian process, with zero mean and covariance function based on the Gaussian kernel with length scale $h = 0.5$. In the sequel, Gaussian noise was added to these GP points, with variance 0.01, to form the set of observed data (shown as '$+$' in the figure below).
- Then, we perform predictions of the output variables corresponding to $D = 1000$ equidistant input points in the interval $[-3, 4]$; for the prediction, the expressions for the posterior GP mean (solid line) and variance, derived before, were used. The shaded area, surrounding the curve of the posterior mean, corresponds to the error bars $\mu_y \pm 2\sigma_y$ of the posterior prediction. Notice the increase of the variance in regions where observed data points are scarce.



Gaussian Processes for Regression: An Example

Popular related package: EDWARD (http://edwardlib.org)