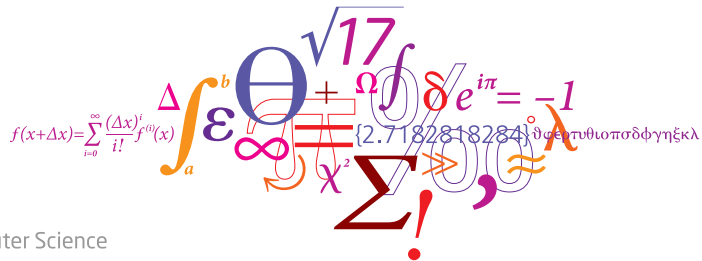
02471 Machine Learning for Signal Processing

Bayesian Inference and the EM algorithm

Tommy Sonne Alstrøm

Cognitive Systems section



DTU Compute
Department of Applied Mathematics and Computer Science

Outline

- Course admin
- Last week review
- General linear models
- Parameter estimation
- Maximum likelihood and Bayesian inference
- Bayesian modeling
- Bayesian estimation in general linear models
- EM algorithm
- Next week

Material: 3.10–3.11, 11.2, 12.1–12.2, 12.4–12.5 (12.10 useful relations)

The story so far and what the future holds

What you have learned so far:

- Parameter estimation [L2 regularization, biased estimation, mean squared error minimization]. L1 regularization, today: Bayesian parameter estimation.
- Filtering signals [Stochastic processes, correlation functions, Wiener filter, linear prediction, adaptive filtering using stochastic gradient decent (LMS, APA/NLMS), adaptive filtering using regularization (RLS)
- Signal representations [Time frequency analysis with STFT], Sparsity aware sensing (lasso, sparse priors), factor models [Independent component analysis, Non-negative matrix factorization, k-SVD],

Next three weeks: Bayesian parameter estimation and probabilistic graphical models

- Inference and EM [Related to parameter estimation] (today). We will need EM for HMM.
- Sequential models [hidden Markov models, linear dynamical systems, Kalman filter].

Learning objectives

Learning objectives

A student who has met the objectives of the course will be able to:

- Explain, apply and analyze properties of discrete time signal processing systems
- Apply the short time Fourier transform to compute the spectrogram of a signal and analyze the signal content
- Explain compressed sensing and determine the relevant parameters in specific applications
- Deduce and determine how to apply factor models such as non-negative matrix factorization (NMF), independent component analysis (ICA) and sparse coding
- Deduce and apply correlation functions for various signal classes, in particular for stochastic signals
- Analyze filtering problems and demonstrate the application of least squares filter components such as the Wiener filter
- Describe, apply and derive non-linear signal processing methods based such as kernel methods and reproducing kernel Hilbert space for applications such as denoising
- Derive maximum likelihood estimates and apply the EM algorithm to learn model parameters
- Describe, apply and derive state-space models such as Kalman filters and Hidden Markov models
- Solve and interpret the result of signal processing systems by use of a programming language
- Design simple signal processing systems based on an analysis of involved signal characteristics, the objective of the processing system, and utility of methods presented in the course
- Describe a number of signal processing applications and interpret the results

Last week review

The linear factor model

Traditionally, digital signal processing seeks to design A (e.g. DCT), machine learning seeks to learn A.

Factor model with n measurements

$$X = AZ$$
$$X := [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N], \quad (l \times n)$$

ICA, NMF, PCA and k-SVD are methods that use the above model, but impose different constraints in order to enforce the desired structure on A and Z.

- k-SVD, $A \in \mathbb{R}^{l \times m}$, $Z \in \mathbb{R}^{m \times n}$, T_0–sparse loadings.
- NMF, $A \in \mathbb{R}_+^{l \times m}$, $Z \in \mathbb{R}_+^{l \times n}$.
- ICA, $A \in \mathbb{R}^{l \times l}$, $Z \in \mathbb{R}^{l \times n}$, estimate A^{-1}, minimize probabilistic dependence in z vectors.

Information

Information of event (discrete variable)

The information measures the amount of "surprise".
The information for a particular value x for a discrete random variable x is

$$I(x) := -\log P(x), \quad P(x) \leq 1 \Rightarrow I(x) \geq 0$$

Mutual information of events (discrete variable)

The information content provided by the occurrence of event y about event x is called mutual information

$$I(x;y) := \log \frac{P(x|y)}{P(x)}, \quad P(x,y) = P(x|y)P(y), \Rightarrow I(x;y) = \log \frac{P(x,y)}{P(x)P(y)}$$

Average mutual information

Mutual information of random variables (discrete case)

$$I(\mathrm{x};\mathrm{y}) := \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}$$

Average mutual information (continous case)

$$I(\mathrm{x};\mathrm{y}) := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy$$

It can be shown that $I(\mathrm{x},\mathrm{y}) \geq 0$ and $I(\mathrm{x},\mathrm{y}) = 0$ implies that x and y are statistically independent.

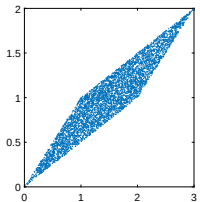
The ICA model

The ICA model

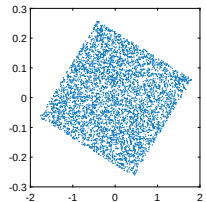
$$\mathbf{x} = A\mathbf{z}$$
$$X \in \mathbb{R}^{l \times n}$$
$$A \in \mathbb{R}^{l \times l}$$
$$Z \in \mathbb{R}^{l \times n}$$

The ICA optimization problem:
Identify A^{-1}, but with respect to maximize statistical independence on the random variables in $\mathbf{z}$

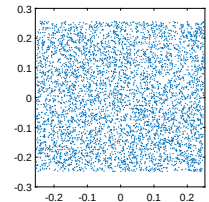
Observations



PCA



ICA



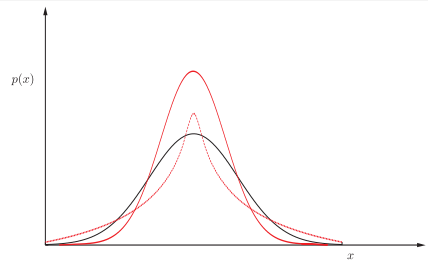
ICA update

ICA based on mutual information and natural gradient descent

1. Randomly initialize W (W is the unmixing matrix and estimates A^{-1}; model is $X = AZ$).
2. Update rule: $W^{(i)} = W^{(i-1)} + \mu_i \left(I - \mathbb{E}\left[\boldsymbol{\phi}(\mathbf{z})\mathbf{z}^T\right]\right) W^{(i-1)}$

Choose $\phi(\mathbf{z})$ as

- $\phi(\mathbf{z}) = 2\tanh(\mathbf{z})$ if $\mathbf{z}$ is super-Gaussian (more spiky, longer tails).
- $\phi(\mathbf{z}) = \mathbf{z} - \tanh(\mathbf{z})$ if $\mathbf{z}$ is sub-Gaussian (faster tail decay).



Summary

- ICA obtains maximal independence but does not reduce dimensionality.
- PCA obtains uncorrelated features, and great for feature extraction and dimensionality reduction.
- NMF is best for analysis of non-negative data, e.g. pixels, energies, count data, etc.
- k-SVD is best for sparse coding if a compact sparse dictionary is wanted with direct control on sparsity.

General linear models

The general linear model

General linear models (NB book calls this generalized linear models)

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

$\phi_k(\mathbf{x})$ is any function that maps $\boldsymbol{x} \in \mathbb{R}^l$, $\phi_k : \mathbb{R}^l \to \mathbb{R}$

Example:

A popular choice for the two-dimensional case is:

$$\phi_1(\mathbf{x}) = \mathrm{x}_1, \phi_2(\mathbf{x}) = \mathrm{x}_2, \phi_3(\mathbf{x}) = \mathrm{x}_1^2, \phi_4(\mathbf{x}) = \mathrm{x}_2^2, \phi_5(\mathbf{x}) = \mathrm{x}_1\mathrm{x}_2$$

What is this model?

The general linear model

General linear models (NB book calls this generalized linear models)

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

$\phi_k(\mathbf{x})$ is any function that maps $\boldsymbol{x} \in \mathbb{R}^l$, $\phi_k : \mathbb{R}^l \to \mathbb{R}$

Example:

A popular choice for the two-dimensional case is:

$$\phi_1(\mathbf{x}) = \mathrm{x}_1, \phi_2(\mathbf{x}) = \mathrm{x}_2, \phi_3(\mathbf{x}) = \mathrm{x}_1^2, \phi_4(\mathbf{x}) = \mathrm{x}_2^2, \phi_5(\mathbf{x}) = \mathrm{x}_1\mathrm{x}_2$$

A linear model with both quadratic effects and interaction effect. The model is used e.g. in ANOVA analysis and statistical analysis of experiments

The model is linear in parameters $\boldsymbol{\theta}$, hence we can use all tools developed so far.

The general linear model matrix form

Suppose we observe N measurements, $(y, \boldsymbol{x}) \in \mathbb{R} \times \mathbb{R}^l$, then

General linear model in matrix form

$$\mathbf{y} = \Phi\boldsymbol{\theta} + \eta$$
$$\boldsymbol{y} = [y_1\ y_2\ \cdots\ y_N]^T \in \mathbb{R}^{N\times 1}$$
$$\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x})\ \phi_2(\boldsymbol{x})\ \cdots\ \phi_{K-1}(\boldsymbol{x})\ 1)]^T \in \mathbb{R}^{K\times 1}$$
$$\Phi = [\boldsymbol{\phi}(\boldsymbol{x}_1)\ \boldsymbol{\phi}(\boldsymbol{x}_2)\ \cdots\ \boldsymbol{\phi}(\boldsymbol{x}_N)]^T \in \mathbb{R}^{N\times K}$$

Why? we had something simple, why make it more complicated ???

We will work with this model for the rest of the lecture.

Note: this model is elaborated substantively in the course 02424 Advanced Data Analysis and Statistical Modeling

Summary

- The general linear model approach is a powerful idea to capture non-linear trends in data, and still keep the parameter estimation task linear.
- Having this in mind, empowers all results we have derived so far for linear models.

Parameter estimation

Parameter estimation so far

Loss function approach

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

$$J(\boldsymbol{\theta}) := \sum_{n=1}^{N} \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))$$

Least-Squares (LS) loss function

$$\mathcal{L}(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) = (y_n - f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))^2$$

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} (y_n - f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))^2$$

$$= \|\boldsymbol{y} - f_{\boldsymbol{\theta}}(X)\|_2^2$$

Maximum likelihood estimation and Bayesian inference

Bayes theorem

$$p(\boldsymbol{\theta}|\mathcal{X}) = \frac{p(\mathcal{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{X})} \quad \mathcal{X} \text{ is our observed data } (\boldsymbol{y}, X)$$

Taking the log we get

$$\ln p(\boldsymbol{\theta}|\mathcal{X}) = \ln p(\mathcal{X}|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) - \ln p(\mathcal{X})$$

$$\underbrace{\ln p(\boldsymbol{\theta}|\mathcal{X})}_{\text{log posterior}} \propto \underbrace{\ln p(\mathcal{X}|\boldsymbol{\theta})}_{\text{log likelihood}} + \underbrace{\ln p(\boldsymbol{\theta})}_{\text{log prior}}$$

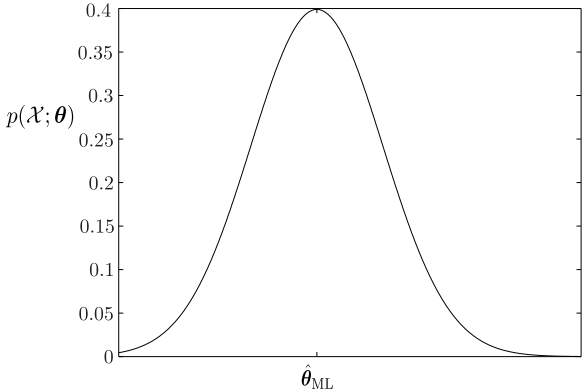
There is a direct correspondence between Bayes theorem and loss functions used so far:

- MSE: Normal likelihood with variance $\sigma^2 I$ + constant prior.
- Ridge: Normal likelihood with variance $\sigma^2 I$ + Normal prior with variance $\sigma_p^2 I$.
- LASSO: Normal likelihood with variance $\sigma^2 I$ + Laplace prior.

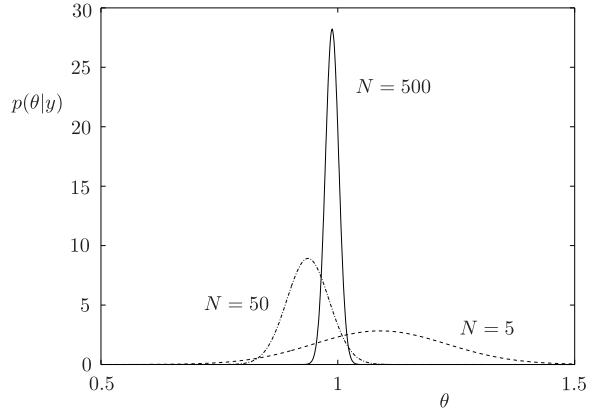
Maximum likelihood and Bayesian inference illustrated

$$\ln p(\boldsymbol{\theta}|\mathcal{X}) = \ln p(\mathcal{X}|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) - \ln p(\mathcal{X})$$

ML estimation: $\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{X}|\boldsymbol{\theta})$



MAP estimation: $\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{X})$



A principled framework

Let us consider the marginal distribution (from eq. 2.23):

Marginal distribution

$$p(y) = \int_{-\infty}^{\infty} p(y, \theta) d\theta = \int_{-\infty}^{\infty} p(y|\theta) p(\theta) d\theta$$

A more informative name is the prior predictive distribution. Let us modify this with our observed data $\mathcal{X}$, and our model parameters $\boldsymbol{\theta}$.
For a specific point $\boldsymbol{x}$ where we want to predict y, we get

$$\begin{aligned} p(y|\boldsymbol{x}, \mathcal{X}) &= \int_{-\infty}^{\infty} p(y, \boldsymbol{\theta}|\boldsymbol{x}, \mathcal{X}) d\boldsymbol{\theta} \\ &= \int_{-\infty}^{\infty} p(y|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{X}) d\boldsymbol{\theta} \end{aligned}$$

This distribution is called the posterior predictive distribution, or sometimes just the predictive distribution.

Three approaches to prediction

Maximum likelihood

$$\hat{\boldsymbol{\theta}}_{\mathsf{ML}} = \arg\max_{\boldsymbol{\theta}} \; p(\mathcal{X}|\boldsymbol{\theta})$$

Maximum a posteriori

$$\hat{\boldsymbol{\theta}}_{\mathsf{MAP}} = \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{\theta}|\mathcal{X})$$

Use the estimated weights, $\hat{\boldsymbol{\theta}}$ to perform prediction, $\hat{y} = f(\boldsymbol{x}, \hat{\boldsymbol{\theta}})$.

Posterior predictive distribution

$$p(y|\boldsymbol{x}, \mathcal{X}) = \int_{-\infty}^{\infty} p(y|\boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{X}) d\boldsymbol{\theta}$$

Use the mean of the posterior predictive distribution to perform prediction $\hat{y} = \mathbb{E}[p(y|\boldsymbol{x}, \mathcal{X})]$.

Summary

- There are a couple of ways to address the parameter estimation problem:
  - Define a cost function and optimize to get point estimates.
  - Define a likelihood function (noise model) and optimize to get point estimates.
  - Define a likelihood function (noise model) and prior probability that encodes prior beliefs, and optimize to get point estimates.
  - Define a likelihood function (noise model) and prior probability that encodes prior beliefs, and learn the complete probability distribution for our parameters.
- We can directly relate the cost function approach to a statistical approach using Bayes formula. In particular
  - LS regression corresponds to Normal likelihood.
  - Ridge regression corresponds to Normal likelihood+Normal prior.
  - LASSO regression corresponds to Normal likelihood+Laplace prior.

Bayesian modeling

Three steps in Bayesian data analysis

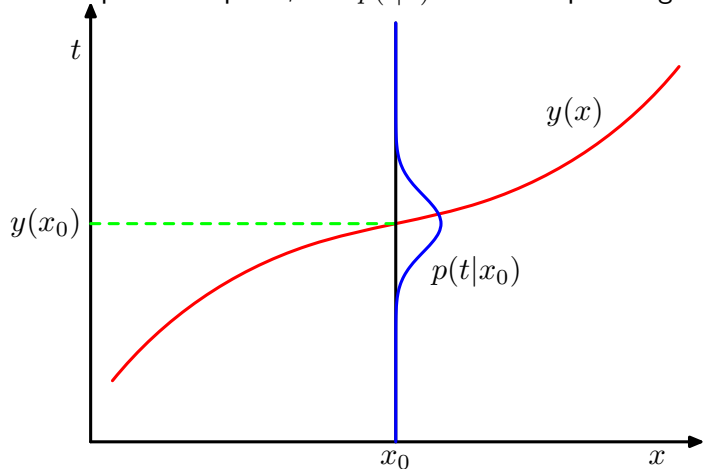
1. Setting up a full probability model – a joint probability distribution for all observable and un-observable quantities in a problem. The model should be consistent with knowledge about the underlying scientific problem and the data collection process.
2. Conditioning on observed data: calculating and interpreting the appropriate posterior distribution – the conditional probability distribution of the unobserved quantities of ultimate interest, given the observed data.
3. Evaluating the fit of the model and the implications of the resulting posterior distribution: does the model fit the data, are the substantive conclusions reasonable, and how sensitive are the results to the modeling assumptions in step I?

If necessary, one can alter or expand the model and repeat the three steps.

Source: Bayesian data analysis, 2014, Gelman, Carlin, Stern, Dunson, Vehtari, Rubin

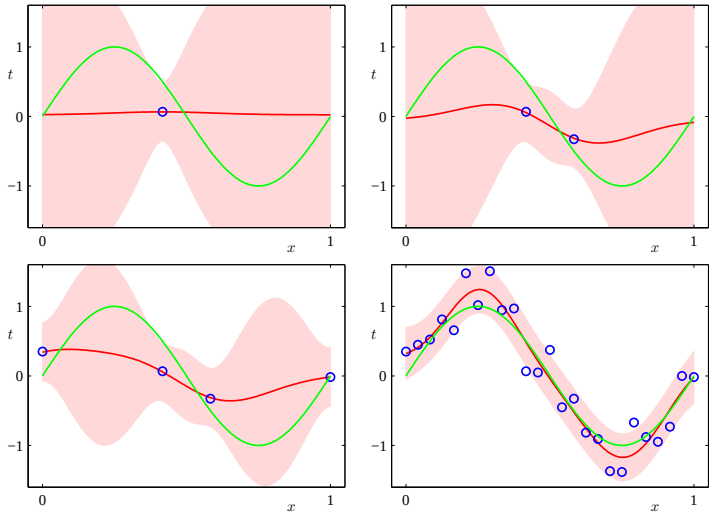
Why Bayesian? example for posterior predictive distribution

Example: $p(t|x_0)$ is the posterior predictive distribution conditioned on a specific point x_0, $y(x_0)$ is the mean at that particular point, and $p(t|x)$ is the complete regression.



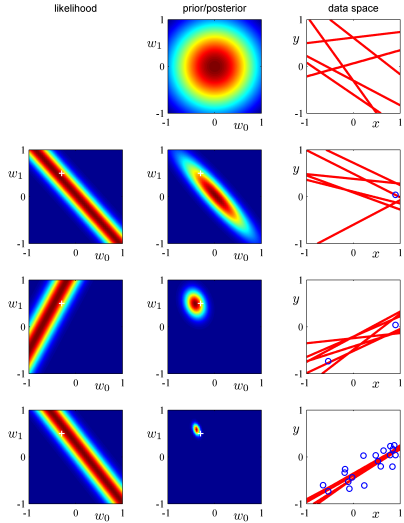
Source: Pattern Recognition and Machine Learning, 2006, C. Bishop

Why Bayesian? inherent uncertainty quantification



Source: Pattern Recognition and Machine Learning, 2006, C. Bishop

Why Bayesian? sequential learning



Source: Pattern Recognition and Machine Learning, 2006, C. Bishop

Why Bayesian? can lead to better predictions

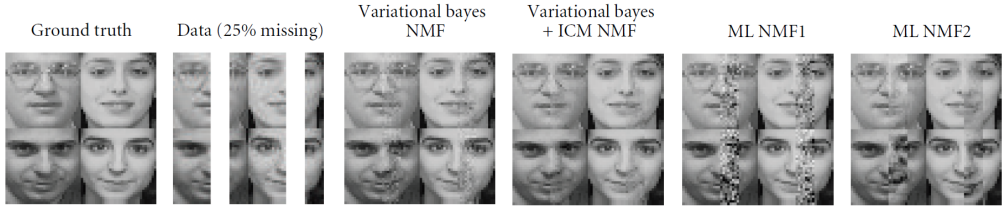


FIGURE 6: Results of a typical run. (a) Example images from the dataset. (b) Comparison of the reconstruction accuracy of different methods in terms of SNR (in dB), organised according to the sparseness of the solution. (c) (from left to right). The ground truth, data with missing pixels. The reconstructions of VB, VB + ICM, and ML-NMF with two initialisation strategies (1 = random, 2 = to image).

Bayesian Inference for Nonnegative Matrix Factorisation Models, 2009, Ali Taylan Cemgil

Note: there are numerous ways to carry out Bayesian inference, the book enumerates some in sec 12.2.3.

Most of those are taught in 02477 Bayesian machine learning.

Bayesian estimation in general linear models

Bayesian modeling: a model with Gaussian likelihood and Gaussian prior

To avoid notational clutter, we will now ignore $\boldsymbol{x}$ and $\mathcal{X}$, and ie. write $p(\boldsymbol{y}|\boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{X}, \boldsymbol{\theta})$.

Assume, from now on: both the likelihood and the prior is Gaussian:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_0, \Sigma_\theta)$$
$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_y, \Sigma_y)$$

For now, we assume $\boldsymbol{\theta}_0$ and Σ_θ are given, and we need to determine $\boldsymbol{\mu}_y$ and Σ_y.

Derivation of $\boldsymbol{\mu}_y$ and Σ_y

We assume the model

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

Let us evaluate the mean:

$$\begin{aligned}\boldsymbol{\mu}_y &= \mathbb{E}[\mathbf{y}] \\ &= \mathbb{E}[\Phi\boldsymbol{\theta} + \eta] \\ &= \Phi\boldsymbol{\theta} + \mathbb{E}[\eta]\end{aligned}$$

For zero-mean noise ($\mathbb{E}[\eta] = 0$) we then have $\boldsymbol{\mu}_y = \Phi\boldsymbol{\theta}$. For the covariance we get

$$\begin{aligned}\Sigma_{\boldsymbol{y}} :=& \mathbb{E}\Big[(\mathbf{y} - \mathbb{E}[\mathbf{y}])\,(\mathbf{y} - \mathbb{E}[\mathbf{y}])^T\Big] \\ =& \mathbb{E}\Big[(\Phi\boldsymbol{\theta} + \eta - \Phi\boldsymbol{\theta})\,(\Phi\boldsymbol{\theta} + \eta - \Phi\boldsymbol{\theta})^T\Big] \\ =& \mathbb{E}\Big[\eta\eta^T\Big] \\ =& \Sigma_\eta\end{aligned}$$

Our updated model I

By substitution our model now becomes

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

Our goal: determine $p(\boldsymbol{\theta}|\boldsymbol{y})$.

We can use relations from sec 12.10. (Skim them to familiarize yourself with the of relations!)

The section is not part of the regular download, you can find it on DTU learn.

The notation choice of x and y is arbitrary. In sec. 12.10, we find:

- Conditional $p(x|y)$ when $p(x, y)$ is Gaussian.
- The marginal $p(x)$ when $p(x, y)$ is Gaussian.
- The posterior $p(y|x)$ when $p(x|y)$ and $p(y)$ are Gaussian distributions (will be used in todays exercise).

Our updated model II

By substitution our model now becomes

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

From sec 12.10. IF:

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_z, \Sigma_z)$$
$$p(\boldsymbol{t}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{t}|\boldsymbol{z}; A\boldsymbol{z}, \Sigma_{t|z})$$

Then the posterior is:

$$p(\boldsymbol{z}|\boldsymbol{t}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{t}; \boldsymbol{\mu}_{z|t}, \Sigma_{z|t})$$
$$\boldsymbol{\mu}_{z|t} = \boldsymbol{\mu}_z + \Sigma_{z|t} A^T \Sigma_{t|z}^{-1} (\boldsymbol{t} - A\boldsymbol{\mu}_z)$$
$$\Sigma_{z|t} = (\Sigma_z^{-1} + A^T \Sigma_{t|z}^{-1} A)^{-1}$$

What is $\boldsymbol{z}$, $\boldsymbol{t}$, $\boldsymbol{\mu}_z$, Σ_z, A, and $\Sigma_{t|z}$?

.

Our updated model II

By substitution our model now becomes

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

From sec 12.10. IF:

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_z, \Sigma_z)$$
$$p(\boldsymbol{t}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{t}|\boldsymbol{z}; A\boldsymbol{z}, \Sigma_{t|z})$$

Then the posterior is:

$$p(\boldsymbol{z}|\boldsymbol{t}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{t}; \boldsymbol{\mu}_{z|t}, \Sigma_{z|t})$$
$$\boldsymbol{\mu}_{z|t} = \boldsymbol{\mu}_z + \Sigma_{z|t} A^T \Sigma_{t|z}^{-1} (\boldsymbol{t} - A\boldsymbol{\mu}_z)$$
$$\Sigma_{z|t} = (\Sigma_z^{-1} + A^T \Sigma_{t|z}^{-1} A)^{-1}$$

Answer: $\boldsymbol{z} = \boldsymbol{\theta}$, $\boldsymbol{t} = \boldsymbol{y}$, $\boldsymbol{\mu}_z = \boldsymbol{\theta}_0$, $\Sigma_z = \Sigma_\theta$, $A = \Phi$, and $\Sigma_{t|z} = \Sigma_\eta$.

Our complete model

The complete Bayesian general linear regression model, with Gaussian likelihood and Gaussian prior

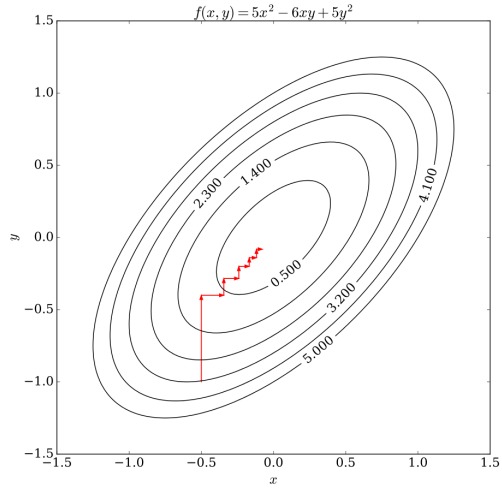
$$
\begin{aligned}
\mathbf{y} &= \Phi\boldsymbol{\theta} + \eta \\
p(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_0, \Sigma_\theta) \\
p(\boldsymbol{y}|\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{y}; \Phi\boldsymbol{\theta}, \Sigma_\eta) \\
p(\boldsymbol{\theta}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y}) \\
\boldsymbol{\mu}_{\theta|y} &= \boldsymbol{\theta}_0 + \Sigma_{\theta|y}\Phi^T\Sigma_\eta^{-1}(\boldsymbol{y} - \Phi\boldsymbol{\theta}_0) \\
\Sigma_{\theta|y} &= (\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi)^{-1}
\end{aligned}
$$

What is required to be estimated to use this model?

EM algorithm

Coordinate descent

The algorithm have a similar feel to coordinate descent (Source: wikipedia):



EM algorithm

The EM algorithm is an iterative algorithm, similar to coordinate descent

The EM algorithm

Steps to carry out before optimization:

- Specification of the complete log-likelihood, $\ln p(\boldsymbol{y}, \boldsymbol{\theta})$ (choice of model).
- Derive $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\big[\ln p(\boldsymbol{y}, \boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\big]$ to create update formulas.

Randomly initialize $\boldsymbol{\xi}^{(0)}$ and run until convergence (e.g until $\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}^{(j)}\| < \epsilon$)

1. Compute $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)})$.
2. Maximize $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)})$ in order to get $\boldsymbol{\xi}^{(j+1)}$ i.e. $\boldsymbol{\xi}^{(j+1)} = \arg\max_{\boldsymbol{\xi}} \mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)})$.

Algorithm analysis and derivation outside the scope of this course.

An example EM update

Let us simplify our linear model an let $\boldsymbol{\theta}_0 = \mathbf{0}$, $\Sigma_\theta = \alpha^{-1} I$ and $\Sigma_\eta = \beta^{-1} I$.

The parameters we need to learn using EM is then $\boldsymbol{\xi} = [\alpha, \beta]^T$.

The original model:

$$
\begin{aligned}
p(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_0, \Sigma_\theta) \\
p(\boldsymbol{y}|\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{y}; \Phi\boldsymbol{\theta}, \Sigma_\eta) \\
p(\boldsymbol{\theta}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y}) \\
\boldsymbol{\mu}_{\theta|y} &= \boldsymbol{\theta}_0 + \Sigma_{\theta|y}\Phi^T\Sigma_\eta^{-1}(\boldsymbol{y} - \Phi\boldsymbol{\theta}_0) \\
\Sigma_{\theta|y} &= (\Sigma_\theta^{-1} + \Phi^T\Sigma_\eta^{-1}\Phi)^{-1}
\end{aligned}
$$

Changes to:

$$
\begin{aligned}
p(\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}, \alpha^{-1} I) \\
p(\boldsymbol{y}|\boldsymbol{\theta}) &= \mathcal{N}(\boldsymbol{y}; \Phi\boldsymbol{\theta}, \beta^{-1} I) \\
p(\boldsymbol{\theta}|\boldsymbol{y}) &= \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_{\theta|y}, \Sigma_{\theta|y}) \\
\boldsymbol{\mu}_{\theta|y} &= \beta\Sigma_{\theta|y}\Phi^T\boldsymbol{y} \\
\Sigma_{\theta|y} &= (\alpha I + \beta\Phi^T\Phi)^{-1}
\end{aligned}
$$

An example EM update

EM for Bayesian linear regression

Expectation step:

$$\boldsymbol{\mu}_{\theta|y}^{(j)} = \beta^{(j)} \Sigma_{\theta|y}^{(j)} \Phi^T \boldsymbol{y}$$

$$\Sigma_{\theta|y}^{(j)} = (\alpha^{(j)} I + \beta^{(j)} \Phi^T \Phi)^{-1}$$

$$A^{(j)} = \text{trace}\left(\Sigma_{\theta|y}^{(j)}\right) + \|\boldsymbol{\mu}_{\theta|y}^{(j)}\|^2$$

$$B^{(j)} = \|\boldsymbol{y} - \Phi \boldsymbol{\mu}_{\theta|y}^{(j)}\|^2 + \text{trace}\left(\Phi \Sigma_{\theta|y}^{(j)} \Phi^T\right)$$

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

Maximization step:

$$\arg\max_{\alpha,\beta} \mathcal{Q}\left(\alpha, \beta; \alpha^{(j)}, \beta^{(j)}\right) \quad \Rightarrow$$

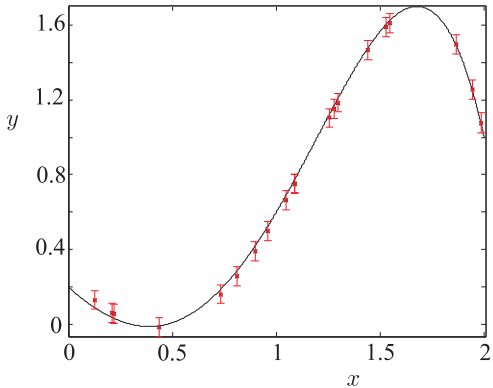
$$\alpha^{(j+1)} = K / A^{(j)}$$

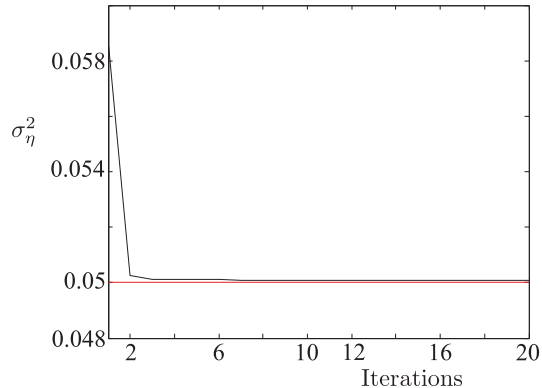
$$\beta^{(j+1)} = N / B^{(j)}$$

EM results

Example 12.1 and 12.2: generate data from $y_n = \sum_{i=0}^{5} \theta_i x_n^i + \eta_n$

Fit and Convergence curve (noise and error bars estimated directly):





You will derive and reproduce this in the exercise.

Lecture summary

- General linear models allows for modeling non-linear trends but still has linear parameter estimation.
- In the Bayesian approach, we learn a distribution over the parameters $\boldsymbol{\theta}$, and not only a parameter estimate.
- Bayesian data modeling provide inherent uncertainty quantification since we learn the distributions, and not point estimates.
- Regularization is inherent using the prior. Example for Ridge regression: the regularization parameter λ is "learned" without performing cross-validation.
- The EM algorithm is a scheme that can be applied, similar to how we have used gradient descent so far.
- Instead of deriving gradients as input to gradient descent, we derive $\mathcal{Q}(\boldsymbol{\xi}, \boldsymbol{\xi}^{(j)}) = \mathbb{E}\big[\ln p(\boldsymbol{y}, \boldsymbol{\theta}; \boldsymbol{\xi}^{(j)})\big]$ to create update formulas.
- EM can be used to learn distribution parameters. We will need EM again for Hidden Markov Model (next week).

Next week

Week 46 material; 15.1–15.3.1, 15.7, 16.4–16.5.

- Probabilistic graphical models.
- Hidden Markov models.