Directed Graphical Models

Cédric Archambeau

Xerox Research Centre Europe
cedric.archambeau@xrce.xerox.com

Accra Bootcamp
Ghana, February 2011

Overview

1. Elements of Bayesian networks
2. Latent variable models for unsupervised learning
3. Conditional models for supervised learning (regression)
4. Elements of probability theory and statistics

Reference material

- D. Koller and N. Friedman (2009): Probabilistic graphical models: principles and techniques.
- C. Bishop (2006): Pattern recognition and machine learning. (Many figures used in these slides are taken from the book.)
- Michael Jordan (1999): Learning in graphical models.
- S. Lauritzen (1996): Graphical models.

- C. E. Rasmussen and C. K.I. Williams (2006): Gaussian processes for machine learning.
- J. Shawe-Taylor and N. Cristianin (2004)i: Kernel methods for pattern analysis.
- D. J. C. MacKay (2003): Information theory, inference, and learning algorithms.
- The Matrix Cookbook by K. B. Petersen and M. S. Pedersen.
- Many interesting tutorials and talks on videolectures!

Statistical machine learning

- A mariage between statistics and computer science
- Digital data is omnipresent (web, images, sound, sensors, ...)
- Data is inherently noisy and unreliable (recording errors, machine faults, ...)
- Modelling strategy: assume the data was generated according to some (hierarchy of) probability distributions
- Amount of data grows exponentially over time, so computational complexity is major issue!

Graphical models

- A marriage between probability theory and graph theory
- Graph theoretic aspect provides intuitive representation and is helpful to analyse, to reason on and to devise new models
- Complex sytems are built by combining simpler parts and the possible relations among them in a probabilistic way
- Probability theory is the glue, ensuring whole system is consistent and can take data into account
- Structured in terms of conditional independence assumptions

Graphical models are applied in ...

- Bioinformatics
- Natural language processing
- Document processing
- Speech processing
- Image processing
- Computer vision
- Time series analysis
- Economics
- Physics
- Social Sciences
- ...

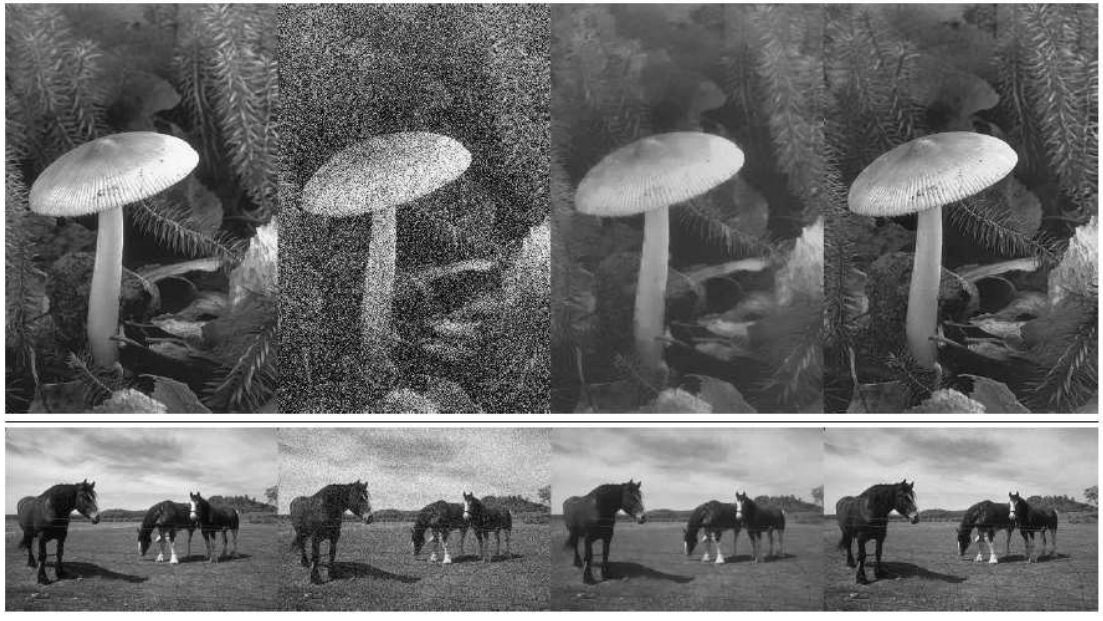
Organising document collections (Blei et al., JMLR 2003)

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

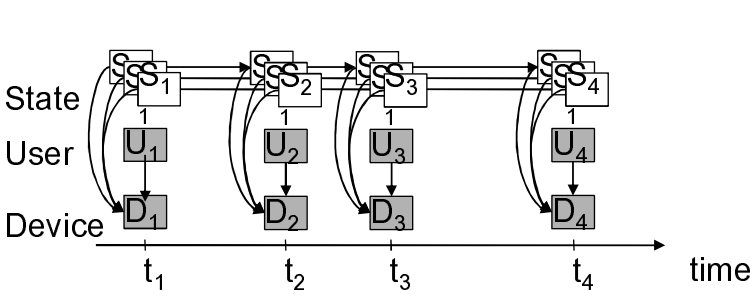
- Discovering themes/topics in large text corpora
- Simple generative model for text (bag-of-words assumption)
- Monitor trends, discover social network, etc.

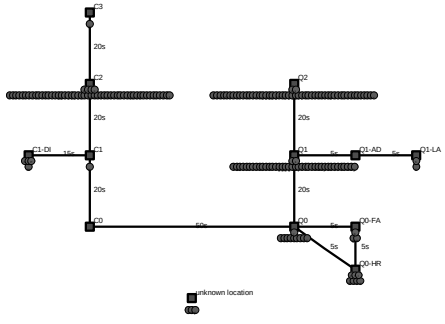
Image denoising (McAuley et al., ICML 2006)



(Markov random fields, use neighborhood information.)

Printer infrastructure management



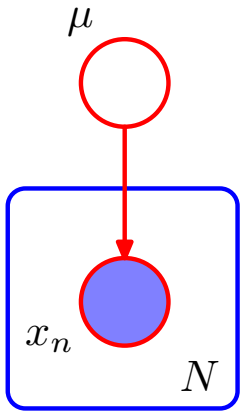


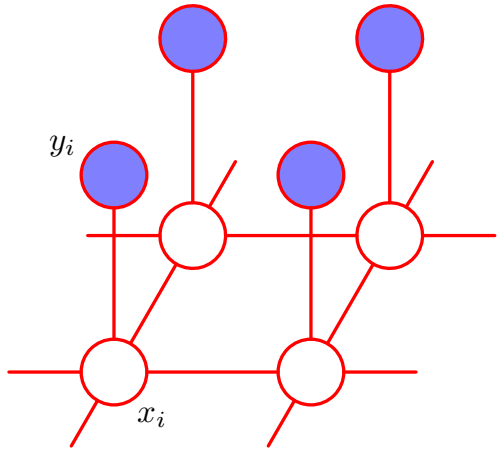
(infrastructure map, user and device locations, printing profiles, device characteristics, soft failure detection, infractructure optimisation, ...)

Part 1: Elements of Bayesian networks

- Conditional independence
- Directed graphical models
- D-separation and Markov blanket
- Learning in Bayesian networks

Basics





- Nodes denote random variables, shaded nodes are observed, unshaded are unobserved (latent, hidden) random variables
- Edges represent conditional dependencies, plates indicate replications
- Directed graphs: Bayesian networks or nets, belief networks, generative models, etc.
- Undirected graphs: Markov networks, Markov Random Fields, etc.
- Combinations are called chain graphs

Conditional independence

- Statistical independence (SI): $X \perp\!\!\!\perp Y$

$$p(x, y) = p(x)p(y).$$

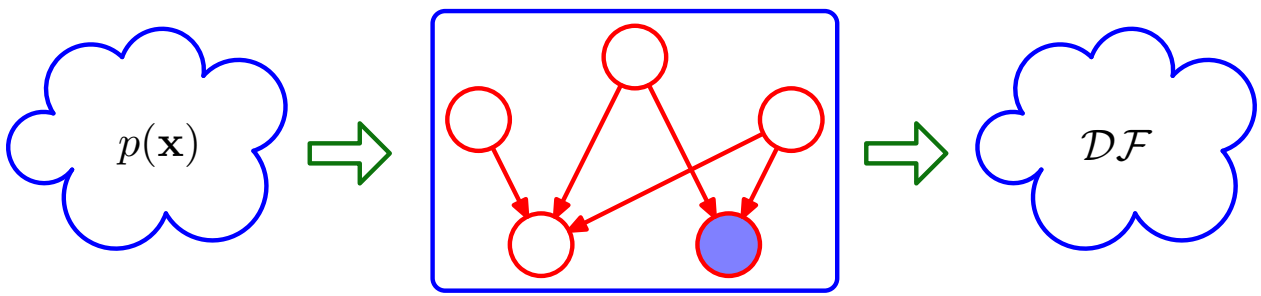
- Conditional independence (CI): $X \perp\!\!\!\perp Y|Z$

$$\begin{aligned} p(x, y|z) &= p(x|y, z)p(y|z) = p(x|z)p(y|z), \\ p(x|y, z) &= p(x|z), \\ p(y|x, z) &= p(y|z). \end{aligned}$$

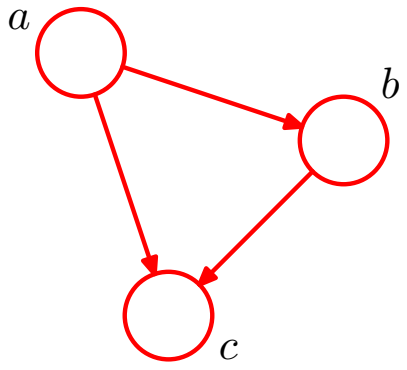
- Examples:
  - My wife's mood $\perp\!\!\!\perp$ my boss' mood | my mood
  - My genome $\perp\!\!\!\perp$ my grandmother's genome | my mother's genome
  - The color of a pixel $\perp\!\!\!\perp$ the color of faraway pixels | the color of neighboring pixels
  - ...

Probabilistic graphical models

- Let $\{X_n\}_{n=1}^N$ be a set of random variables.
- A probabilistic graphical model is a family of joint probability distributions $p(x_1, \ldots, x_N)$ for which some CI assumptions hold.
- The set of CI assumptions $\{X_i \perp\!\!\!\perp X_j | X_k\}$ induces a structure in $p(x_1, \ldots, x_N)$, which is made explicit in the graph.
- This structure allows us to make computations more tractable and storage more efficient.
- In the case of Bayesian networks this is sometimes called a directed factorisation (DF) filtering of the joint:



Bayesian networks (directed graphical models)



- A Bayesian network is a set of probability distributions associated to a directed acyclic graph (DAG).
- Node *a* is a parent of node *b* if there is a directed link from *a* to *b* (conversely we say that *b* is a child of *a*).
- A node is independent of its ancestors given its parents.
- Random variables can be discrete or continuous.

Factorisation in directed graphical models

- CIs lead to a particular factorisation of the joint!
- For a graph with N nodes, we decompose the joint in terms of conditionals on the parents:

$$p(x_1, \ldots, x_N) = \prod_{n=1}^{N} p(x_n | \mathrm{pa}_n), \qquad \mathrm{pa}_n : \text{ parents of } x_n.$$

- The factorisation is in terms of local conditional distributions.
- The joint is correctly normalised.

Is the CI-based factorisation useful?

- Consider the special case where $M = 3$:

$$p(x_1, x_2, x_3) = p(x_1)p(x_2|x_1)p(x_3|x_2), \quad \forall m : x_m \in \{1, \ldots, K\}.$$

- Factorisation allows us to exploit the distributive law to make computations more tractable:

  without: $p(x_2) = \sum_{x_1, x_3} p(x_1, x_2, x_3)$ is $\mathcal{O}(K^3)$

  with: $p(x_2) = \sum_{x_1} p(x_1, x_2) \sum_{x_3} p(x_3|x_2)$ is $\mathcal{O}(2K^2)$

- Factorisation leads to a more efficient representation:

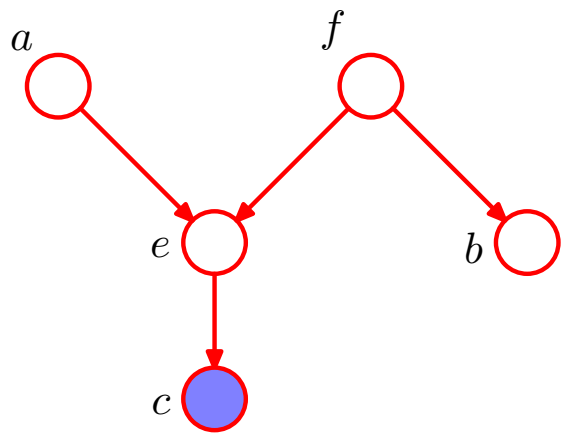
  without: requires $K^M - 1$ parameters

  with: requires $K - 1 + (M - 1)K(K - 1)$ parameters

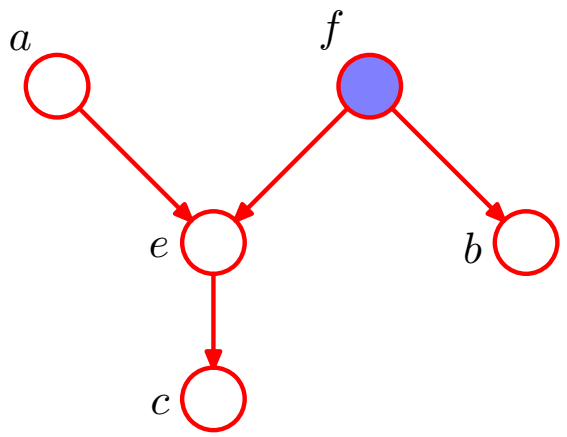
Is there a rule to deduce CIs directly from the DAG?

- CIs are usually known by the (human) expert.
- CIs are imposed by removing links.
- Do we induce other (hidden) CIs?

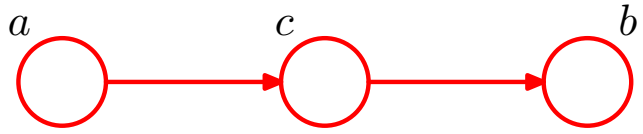
$a \perp\!\!\!\perp b$?



$a \perp\!\!\!\perp b | f$?



Head-to-tail nodes: statistical independence

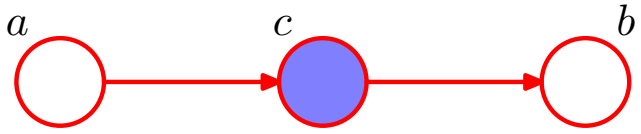


- Consider the head-to-tail node c. Are a and b independent?
- Let's check SI: $p(a,b) \stackrel{?}{=} p(a)p(b)$:

$$\begin{aligned} p(a,b) &= \sum_c p(a,b,c) = \sum_c p(a)p(c|a)p(b|c) \\ &= p(a)\sum_c p(c|a)p(b|c) = p(a)\sum_c p(b,c|a) \\ &= p(a)p(b|a) \end{aligned}$$

- No SI in general.

Head-to-tail nodes: conditional independence

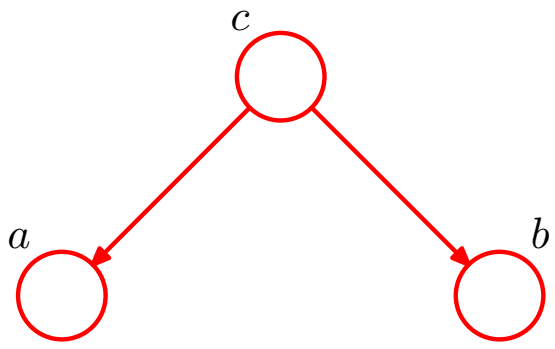


- Assume c is observed. Are a and b conditionally independent?
- Let's check CI: $p(a, b|c) \stackrel{?}{=} p(a|c)p(b|c)$:

$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a)p(c|a)p(b|c)}{p(c)} = p(a|c)p(b|c)$$

- We obtain $a \perp\!\!\!\perp b|c$.
- Applying Bayes rule reverts the link!

Tail-to-tail nodes: statistical independence

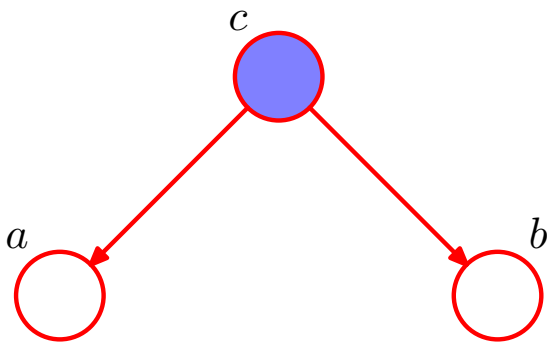


- Consider the tail-to-tail node c. Are a and b independent?
- Let's check SI: $p(a,b) \stackrel{?}{=} p(a)p(b)$:

$$\begin{aligned} p(a,b) &= \sum_c p(a,b,c) = \sum_c p(a|c)p(b|c)p(c) \\ &= \sum_c p(a)p(c|a)p(b|c) = p(a)\sum_c p(b,c|a) \\ &= p(a)p(b|a) \end{aligned}$$

- No SI in general.

Tail-to-tail nodes: conditional independence

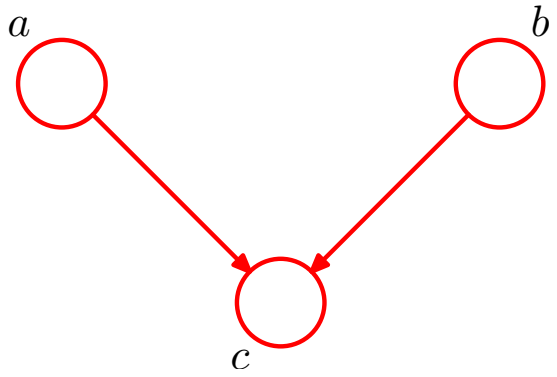


- Assume c is observed. Are a and b conditionally independent?
- Let's check CI: $p(a, b|c) \stackrel{?}{=} p(a|c)p(b|c)$:

$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a|c)p(b|c)p(c)}{p(c)} = p(a|c)p(b|c)$$

- We obtain $a \perp\!\!\!\perp b|c$.

Head-to-head nodes: statistical independence

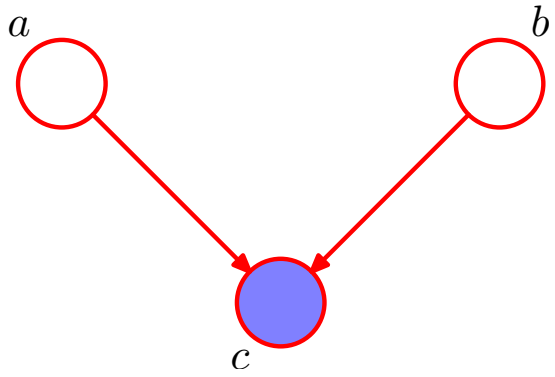


- Consider the head-to-head node c. Are a and b independent?
- Let's check SI: $p(a,b) \stackrel{?}{=} p(a)p(b)$:

$$p(a,b) = \sum_c p(a,b,c) = \sum_c p(a)p(b)p(c|a,b)$$
$$= p(a)p(b)\sum_c p(c|a,b) = p(a)p(b)$$

- We obtain $a \perp\!\!\!\perp b$.

Head-to-head nodes: conditional independence



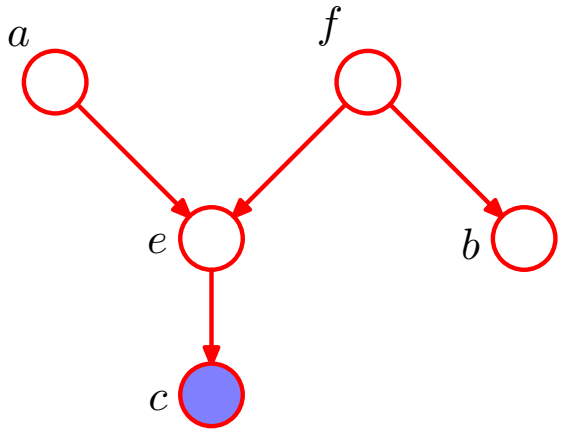
- Assume c is observed. Are a and b conditionally independent?
- Let's check CI: $p(a, b|c) \stackrel{?}{=} p(a|c)p(b|c)$:

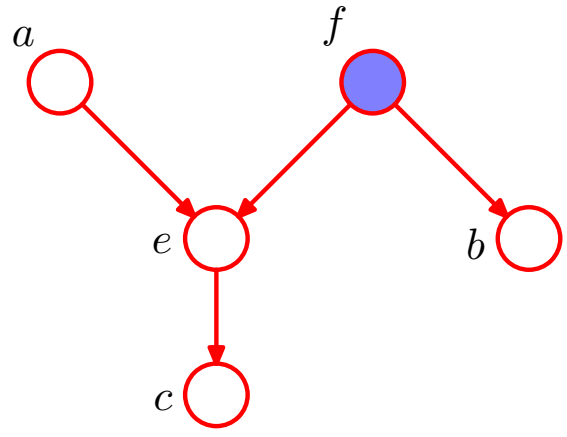
$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a)p(b)p(c|a, b)}{p(c)}$$

- No CI in general.

D-separation

- A blocked path is one containing at least one of the following types of nodes:
  - An observed head-to-tail or tail-to-tail node.
  - An unobserved head-to-head node, of which none of the descendants are observed.





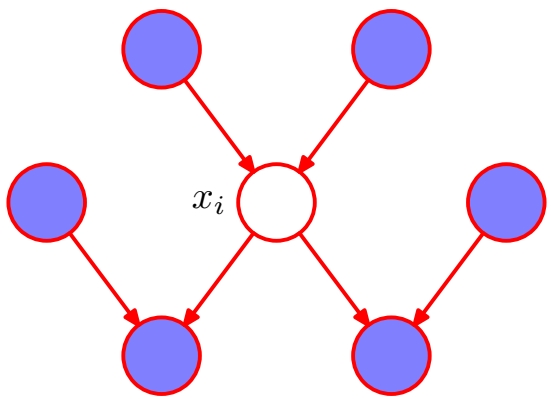
- Let A, B and C be nonintersecting sets of nodes. $A \perp\!\!\!\perp B | C$ if all possible paths from any node in A to any node in B are blocked.
- We say that A is d-separated from B by C.
- d-separation allows us to directly reason on the graph.

Markov blanket

- The Markov blanket of x_i is the minimal set of nodes that isolates x_i from the rest of the graph.
- Using CI we can express any conditional $p(x_i|\{x_j\}_{j\neq i})$:

$$p(x_i|\{x_n\}_{n\neq i}) = \frac{\prod_n p(x_n|\mathrm{pa}_n)}{\sum_{x_i}\prod_n p(x_n|\mathrm{pa}_n)} \propto p(x_i|\mathrm{pa}_i)\prod_{n_i} p(x_{n_i}|\mathrm{pa}_{n_i}),$$

  where pa_{n_i} includes node x_i.
- The Markov blanket of x_i contains the parents and children of x_i, as well as co-parents (spouses) of the children of x_i.



Learning in Bayesian networks

- We assume the data $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{M}$ are drawn i.i.d.:
  - All $\mathbf{x}^{(i)}$ are drawn from the same distribution (identical assumption).
  - $\mathbf{x}^{(i)} \perp\!\!\!\perp \mathbf{x}^{(j)}$ for $i \neq j$ (independence assumption).
- We posit a statistical model:

$$p(\mathbf{x}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{s} f_s(\mathbf{x}_s; \boldsymbol{\theta}_s).$$

- The quality of model depends on the vector of parameters $\boldsymbol{\theta}$.
- The goal of learning is to estimate $\boldsymbol{\theta}$:
  - Maximum likelihood
  - Maximum a posteriori
  - Bayesian inference
- We assume for now that there are no latent (hidden) variables.

Maximum likelihood (ML) estimation

- The likelihood is the joint probability of observing i.i.d. data:

$$\ell(\boldsymbol{\theta}; \mathbf{X}) = \ln \prod_i p(\mathbf{x}^{(i)}; \boldsymbol{\theta}) = \sum_i \sum_s \ln f_s(\mathbf{x}_s^{(i)}; \boldsymbol{\theta}_s) - M \ln Z(\boldsymbol{\theta}).$$

- The goal in ML is to find the parameters that maximise the log-likelihood function:

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{X}).$$

- A local optimum must satisfy $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{X}) = 0$ or equivalently $\nabla_{\boldsymbol{\theta}_s} \ell(\boldsymbol{\theta}; \mathbf{X}) = 0$ for all s.
- Alternatively one can minimise the negative log-likelihood.
- ML is asymptotically (i.e. when $M \to \infty$) consistent.

Maximum a posteriori (MAP) estimation

- The likelihood is unbounded, so ML can lead to overfitting (especially for small data set).
- Penalise unreasonable values ($\sim$ regularisation) by imposing a prior distribution on the parameters:

$$p(\boldsymbol{\theta}|\mathbf{X}) \propto p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

- MAP maximises the penalised log-likelihood :

$$\ell_{\mathrm{MAP}}(\boldsymbol{\theta}; \mathbf{X}) = \ell(\boldsymbol{\theta}; \mathbf{X}) + \sum_s \ln p(\boldsymbol{\theta}_s).$$

- A local optimum must satisfy $\nabla_{\boldsymbol{\theta}} \ell_{\mathrm{MAP}}(\boldsymbol{\theta}; \mathbf{X}) = 0$ or equivalently $\nabla_{\boldsymbol{\theta}_s} \ell_{\mathrm{MAP}}(\boldsymbol{\theta}; \mathbf{X}) = 0$ for all s.
- MAP leads to a point estimate of $\boldsymbol{\theta}$ (while Bayesian statistics is interested in the full posterior).
- MAP is not invariant under reparametrisation.

Bayesian inference

- Bayesian statistics views $\boldsymbol{\theta}$ as a latent variable and is interested the full posterior of $\boldsymbol{\theta}$ instead of a point estimate.
- The prior information (if any) is encoded in the prior distribution and is updated into a posterior distribution based on the data:

$$\underbrace{p(\boldsymbol{\theta}|\mathbf{X})}_{\text{posterior}} = \frac{\overbrace{p(\mathbf{X}|\boldsymbol{\theta})}^{\text{likelihood}}\overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{\underbrace{p(\mathbf{X})}_{\text{evidence}}}, \qquad p(\mathbf{X}) = \int p(\mathbf{X}, \boldsymbol{\theta})\ d\boldsymbol{\theta}.$$

- Prediction is performed by averaging over all possible models:

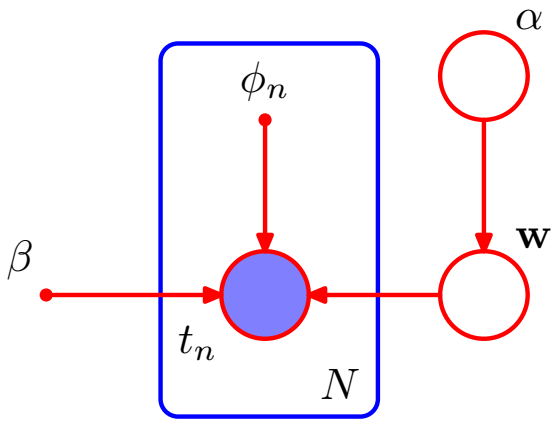
$$p(\mathbf{x}_*|\mathbf{X}) = \int p(\mathbf{x}_*|\boldsymbol{\theta})\ p(\boldsymbol{\theta}|\mathbf{X})\ d\boldsymbol{\theta}.$$

- The Bayesian approach provides confidence measures for estimates and predictions.
- Computing the marginals is in general analytically intractable...

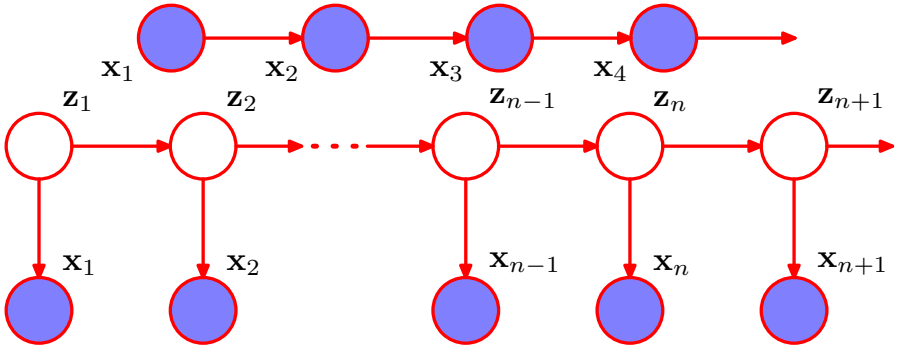
Part 2: Latent variable models for unsupervised learning

- Learning latent variable models
- Discrete latent variables:
  - Mixture of Bernoullis
  - Mixture of Gaussians
- Continuous latent variables:
  - Probabilistic PCA
  - Extensions

Latent variable models



- Nodes represent random variables or parameters.
- Random variables can be observed or unobserved (latent).
- Latent variables are a flexible way to model the data.
- Example: hidden Markov models for sequential data



$$\mathbf{x}_{n+1} \perp\!\!\!\perp \mathbf{x}_{n-1} \,|\, \mathbf{x}_n$$

$$\mathbf{z}_{n+1} \perp\!\!\!\perp \mathbf{z}_{n-1} \,|\, \mathbf{z}_n$$

$$\mathbf{x}_n \perp\!\!\!\perp \mathbf{x}_{n-1} \,|\, \mathbf{z}_n$$

Expectation-maximisation (EM)

- Assume there are observed as well as latent variables:

$$p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_s f_s(\mathbf{x}_s, \mathbf{z}_s; \boldsymbol{\theta}_s).$$

- If we knew the latent variables $\{\mathbf{z}^{(i)}\}$, the problem of learning $\boldsymbol{\theta}$ would reduce to ML (or MAP) estimation.
- Since $\{\mathbf{z}^{(i)}\}$ are unobserved, ML requires to maximise the incomplete log-likelihood:

$$\begin{aligned}\ell(\boldsymbol{\theta}; \mathbf{X}) &= \ln \prod_i \sum_{\mathbf{z}^{(i)}} p(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}; \boldsymbol{\theta}) \\ &= \sum_i \ln \sum_{\mathbf{z}^{(i)}} \prod_s f_s(\mathbf{x}_s^{(i)}; \boldsymbol{\theta}_s) - M \ln Z(\boldsymbol{\theta}).\end{aligned}$$

- The product is "blocked" inside the logarithm because of the sum, making the marginalisation often analytically intractable.

EM (lower bound)

- The key idea is to maximise the expected value of the log-complete likelihood since $\mathbf{Z} = \{\mathbf{z}^{(i)}\}$ are unobserved:

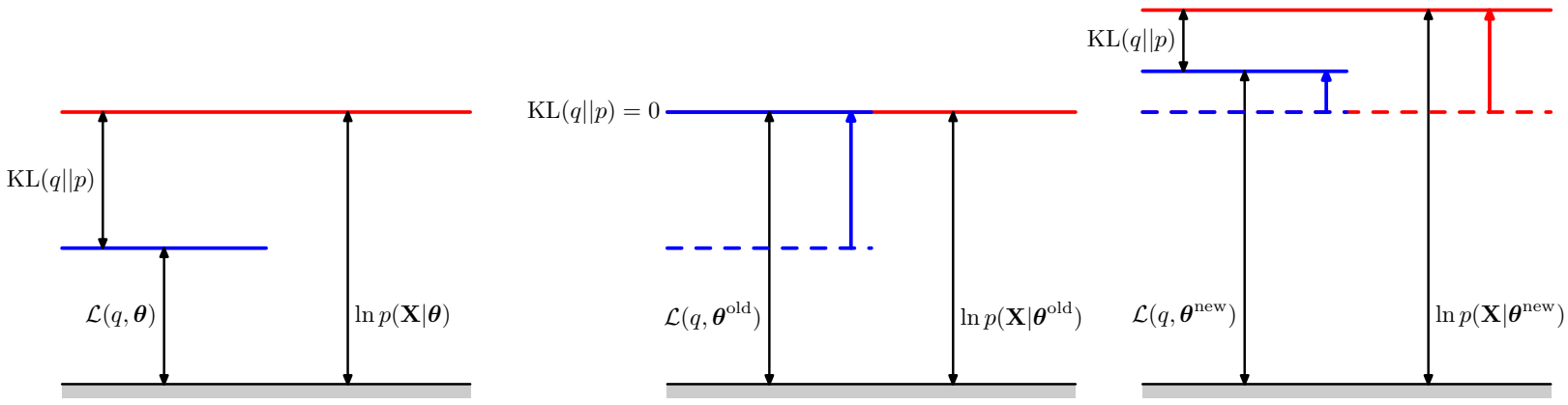
$$\begin{aligned}
\ell(\boldsymbol{\theta}; \mathbf{X}) &= \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}; \boldsymbol{\theta}) \\
&= \ln \sum_{\mathbf{Z}} q(\mathbf{Z}) \frac{p(\mathbf{X}, \mathbf{Z}; \boldsymbol{\theta})}{q(\mathbf{Z})} \\
&\geq \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z}; \boldsymbol{\theta})}{q(\mathbf{Z})} \equiv \mathcal{L}(q, \boldsymbol{\theta})
\end{aligned}$$

  where $q(\mathbf{Z})$ is called the variational distribution.
- The lower bound follows from Jensen's inequality:

$$f(x) \text{ is convex} \Rightarrow \mathbb{E}(f(x)) \geq f(\mathbb{E}(x)).$$

- The quantity $-\mathcal{L}(q, \boldsymbol{\theta})$ can be interpretted as the (variational) free energy from statistical physics.

EM (principle)

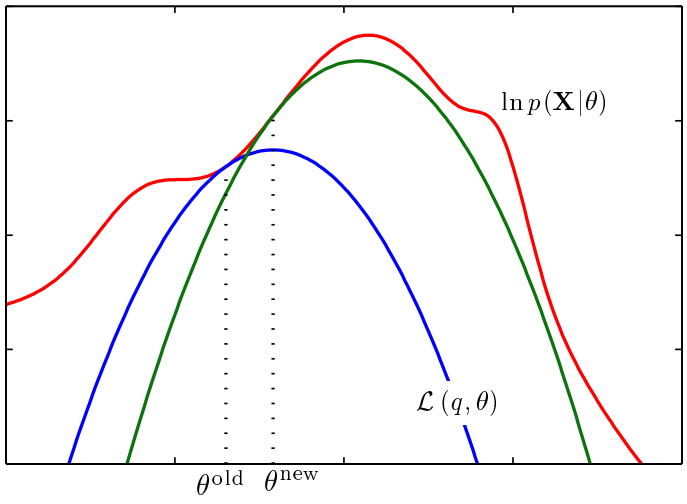


EM is based on two decompositions of the bound $\mathcal{L}(q, \boldsymbol{\theta})$:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \ln p(\mathbf{X}|\boldsymbol{\theta}) - \mathrm{KL}[q(\mathbf{Z}) \| p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})],$$
$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_q\{\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})\} + \mathrm{H}[q(\mathbf{Z})].$$

where $\mathrm{KL}[q\|p] = \mathbb{E}_q\{\ln \frac{q}{p}\}$ is the Kullback-Leibler divergence (or relative entropy) and $\mathrm{H}[q] = -\mathbb{E}\{\ln q)\}$ the entropy.

EM (algorithm)



- Maximise lower bound by alternating between 2 steps:

  E step: Minimise KL for fixed $\boldsymbol{\theta}$ by setting $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$.

  M step: Maximise $\mathbb{E}_q\{\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})\}$ for given $q(\mathbf{Z})$.

- Gradient ascent to local maxima of $\ell(\boldsymbol{\theta}; \mathbf{X})$, by construction it ensures monotonic increase of the bound.
- ML estimates of the parameters, still ok if q is a good approximation of the posterior (approximate E step).

Mixture of Bernoulli distributions

- Let $\mathbf{x} = (x_1, \ldots, x_N)$ be a set of binary variables (e.g. B&W image).
- Each component is a product of Bernoulli distributions:

$$p(\mathbf{x}|\boldsymbol{\mu}_k) = \prod_n \text{Bernoulli}(\mu_{kn}) = \prod_n \mu_{kn}^{x_n}(1-\mu_{kn})^{1-x_n}.$$

- The mixture model (likelihood) is defined as

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_k \pi_k p(\mathbf{x}|\boldsymbol{\mu}_k), \qquad \pi_k \in [0,1], \qquad \sum_k \pi_k = 1.$$

- No closed form solution for ML estimates of $\boldsymbol{\theta} = \{\boldsymbol{\mu}_k, \pi_k\}$.

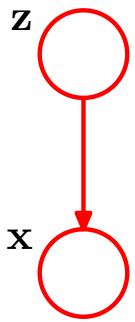
Mixture of Bernoulli distributions (continued)

- For each set of binary variables $\mathbf{x}$ we introduce a discrete latent variable z which indicates the mixture component:

$$p(z|\boldsymbol{\pi}) = \text{Discrete}(\boldsymbol{\pi}) = \prod_k \pi_k^{\delta_k(z)}.$$

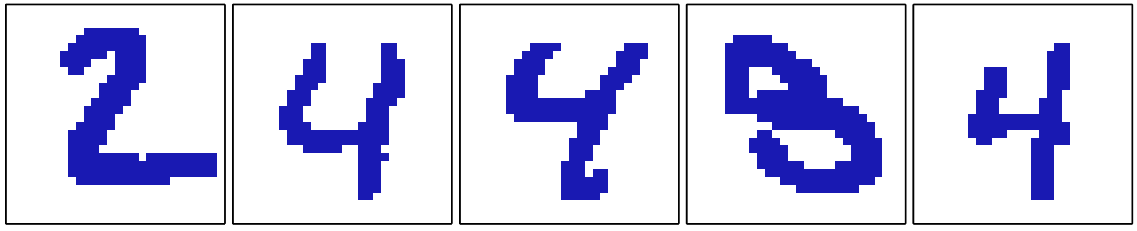
- The new graphical model is completed by

$$p(\mathbf{x}|z, \boldsymbol{\mu}) = \prod_k \left[\prod_n \text{Bernoulli}(\mu_{kn}) \right]^{\delta_k(z)}.$$

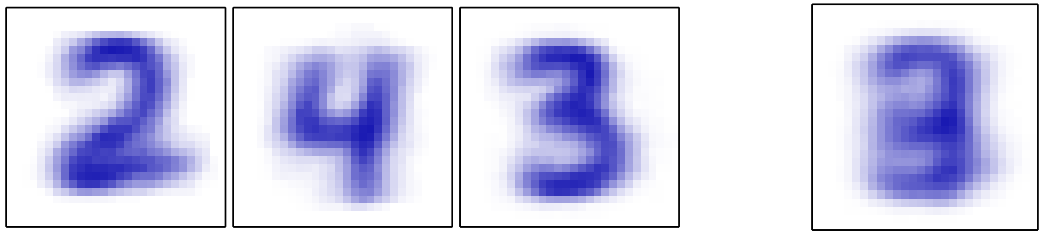


- The marginal likelihood is recovered by summing over z.

Mixture of Bernoullis (application)



- Pixelised handwritten digits, converted from grey scale to binary images by thresholding
- Goal is to cluster the images (recognise digit automatically), learning is done with EM algorithm
- The bottom figure shows the mean images for each of the 3 clusters, as well as the mean image when considering a single Bernoulli.



Mixture of Bernoullis (EM updates)

E step : responsibilities:

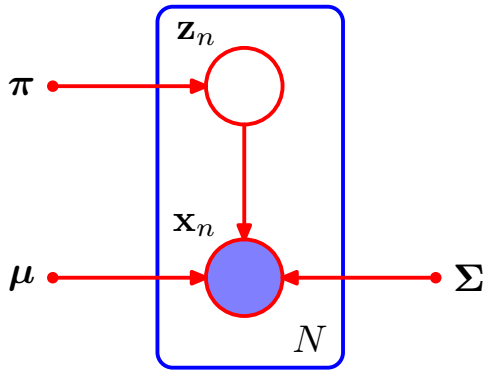
$$\rho_{ik} \equiv \mathbb{E}\{z^{(i)} = k\} = \frac{\pi_k p(\mathbf{x}^{(i)}|\boldsymbol{\mu}_k)}{\sum_{k'} \pi_{k'} p(\mathbf{x}^{(i)}|\boldsymbol{\mu}_{k'})}$$

M step : mean and mixture proportions:

$$\boldsymbol{\mu}_k = \frac{1}{M_k} \sum_{i=1}^{M} \rho_{ik} \mathbf{x}^{(i)}, \qquad M_k = \sum_{i=1}^{M} \rho_{ik},$$

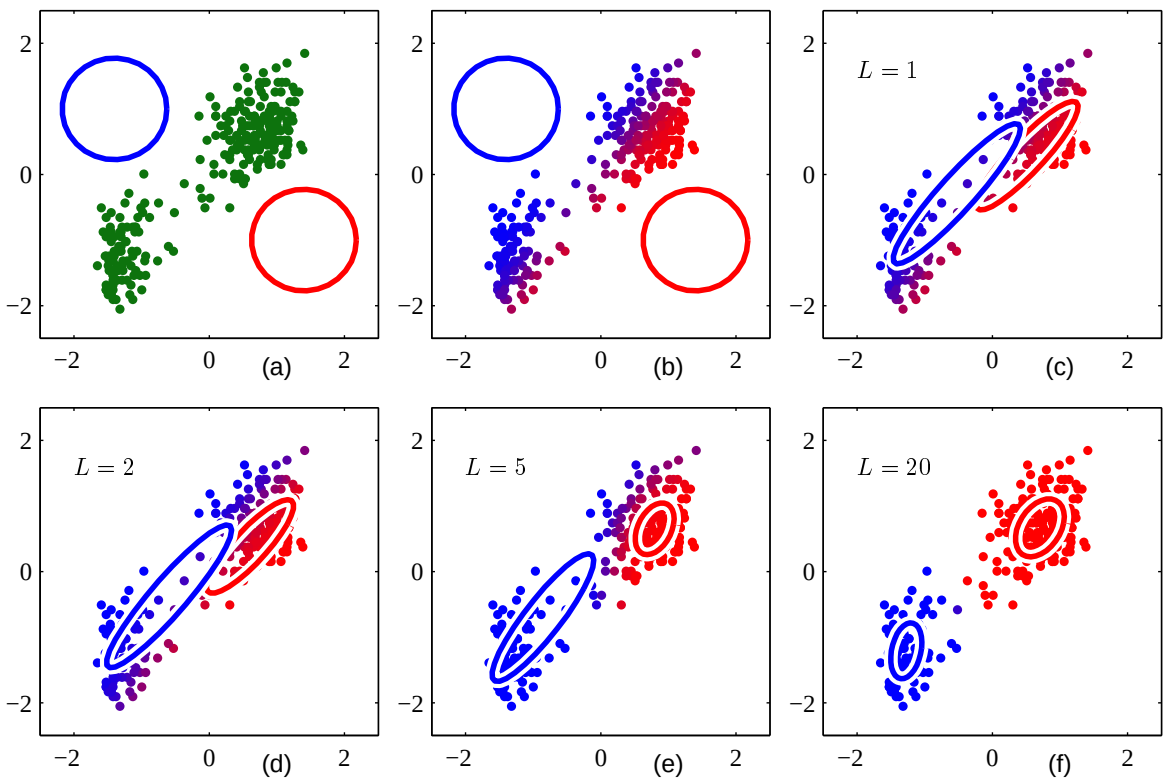
$$\pi_k = \frac{M_k}{M}.$$

Mixture of Gaussians (Old Faithful geyser data)



$$p(\mathbf{x}|\{\boldsymbol{\mu}_k\},\{\boldsymbol{\Sigma}_k\},\boldsymbol{\pi}) = \sum_k \pi_k \mathcal{N}(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k),$$

$$\pi_k \in [0,1], \quad \sum_k \pi_k = 1.$$



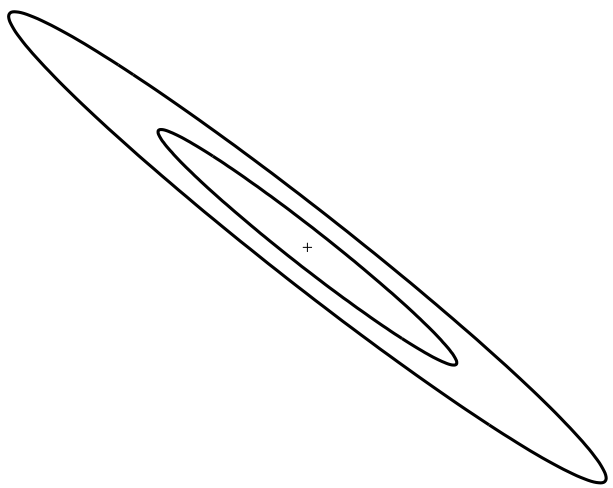
Multivariate Gaussian density

Let $\mathbf{x}$ be a D-dimensional Gaussian random vector.

The density of $\mathbf{x}$ is defined as

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-D/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\},$$

where $\boldsymbol{\mu} \in \mathbb{R}^{D \times 1}$ is the mean and $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ is the covariance matrix.



2-dimensional Gaussian

Gaussian identities

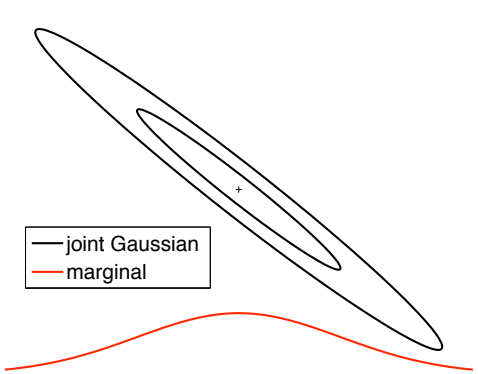
Let $\mathbf{x}$ and $\mathbf{y}$ be **jointly Gaussian**:

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_\mathbf{x} \\ \boldsymbol{\mu}_\mathbf{y} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{xx}} & \boldsymbol{\Sigma}_{\mathbf{xy}} \\ \boldsymbol{\Sigma}_{\mathbf{xy}}^\top & \boldsymbol{\Sigma}_{\mathbf{yy}} \end{bmatrix} \right).$$

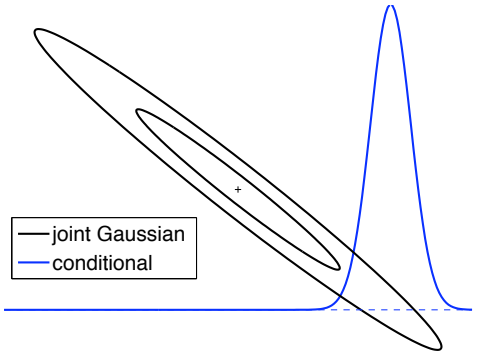
The **marginal** $p(\mathbf{x})$ is Gaussian with mean $\boldsymbol{\mu}_\mathbf{x}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{xx}}$.

The **conditional** $p(\mathbf{x}|\mathbf{y})$ is Gaussian with mean and covariance equal to

$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}} = \boldsymbol{\mu}_\mathbf{x} + \boldsymbol{\Sigma}_{\mathbf{xy}} \boldsymbol{\Sigma}_{\mathbf{yy}}^{-1} (\mathbf{y} - \boldsymbol{\mu}_\mathbf{y}),$$
$$\boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}} = \boldsymbol{\Sigma}_{\mathbf{xx}} - \boldsymbol{\Sigma}_{\mathbf{xy}} \boldsymbol{\Sigma}_{\mathbf{yy}}^{-1} \boldsymbol{\Sigma}_{\mathbf{xy}}^\top.$$



(a) Marginal.



(b) Conditional.

Gaussian identities (continued)

Consider the following two Gaussian distributions:

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_{\mathbf{xx}}),$$
$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{A}\mathbf{x} + \mathbf{b}, \boldsymbol{\Lambda}).$$

The **marginal** $p(\mathbf{y})$ is Gaussian with mean and covariance given by

$$\boldsymbol{\mu}_\mathbf{y} = \mathbf{A}\boldsymbol{\mu}_\mathbf{x} + \mathbf{b},$$
$$\boldsymbol{\Sigma}_{\mathbf{yy}} = \boldsymbol{\Lambda} + \mathbf{A}\boldsymbol{\Sigma}_{\mathbf{xx}}\mathbf{A}^\top.$$

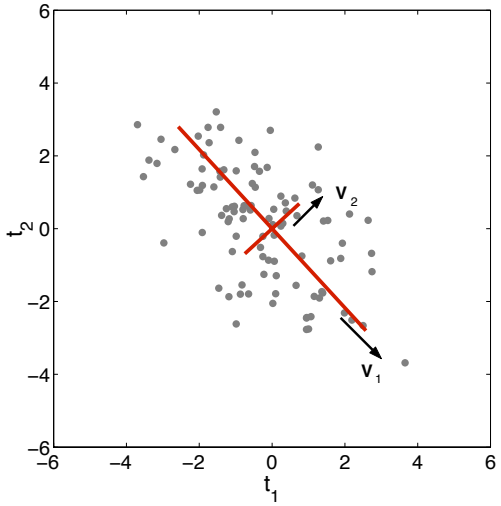
The **posterior** $p(\mathbf{x}|\mathbf{y})$ is Gaussian with mean and covariance equal to

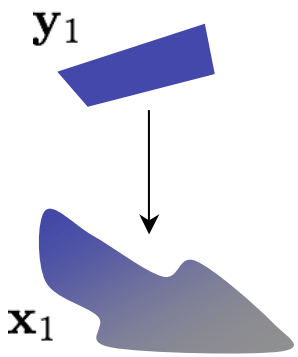
$$\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}} = \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}}\{\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}\boldsymbol{\mu}_\mathbf{x} + \mathbf{A}^\top\boldsymbol{\Lambda}^{-1}(\mathbf{y} - \mathbf{b})\},$$
$$\boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y}} = (\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} + \mathbf{A}^\top\boldsymbol{\Lambda}^{-1}\mathbf{A})^{-1}.$$

(For proofs see for example chapter 2 of Bishop, 2006.)

Probabilistic principal component analysis (PPCA)

- PCA is a standard pre-processing tool for (linear) dimensionality reduction.
- It uses a maximal variance criterion (or minimal mean squared reconstruction error).
- Standard algorithms are $\mathcal{O}(D^3)$ (e.g. Gaussian elimination).





- PPCA assumes a single Gaussian latent variable and a Gaussian likelihood.
- ML solution spans same subspace as PCA solution.
- Standard EM is $\mathcal{O}(DNd)$ per iteration.

Probabilistic principal component analysis (PPCA)

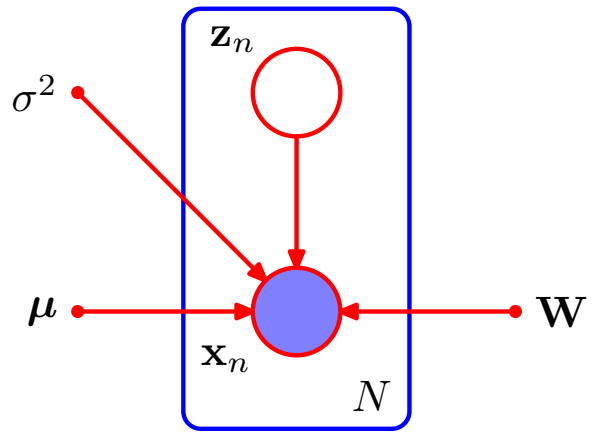
$$\mathbf{x}_n = \mathbf{W}\mathbf{z}_n + \boldsymbol{\mu} + \boldsymbol{\epsilon}_n$$

- Likelihood (noise model):

$$\mathbf{x}_n|\mathbf{z}_n \sim \mathcal{N}(\mathbf{W}\mathbf{z}_n + \boldsymbol{\mu}, \sigma^2\mathbf{I}_D).$$

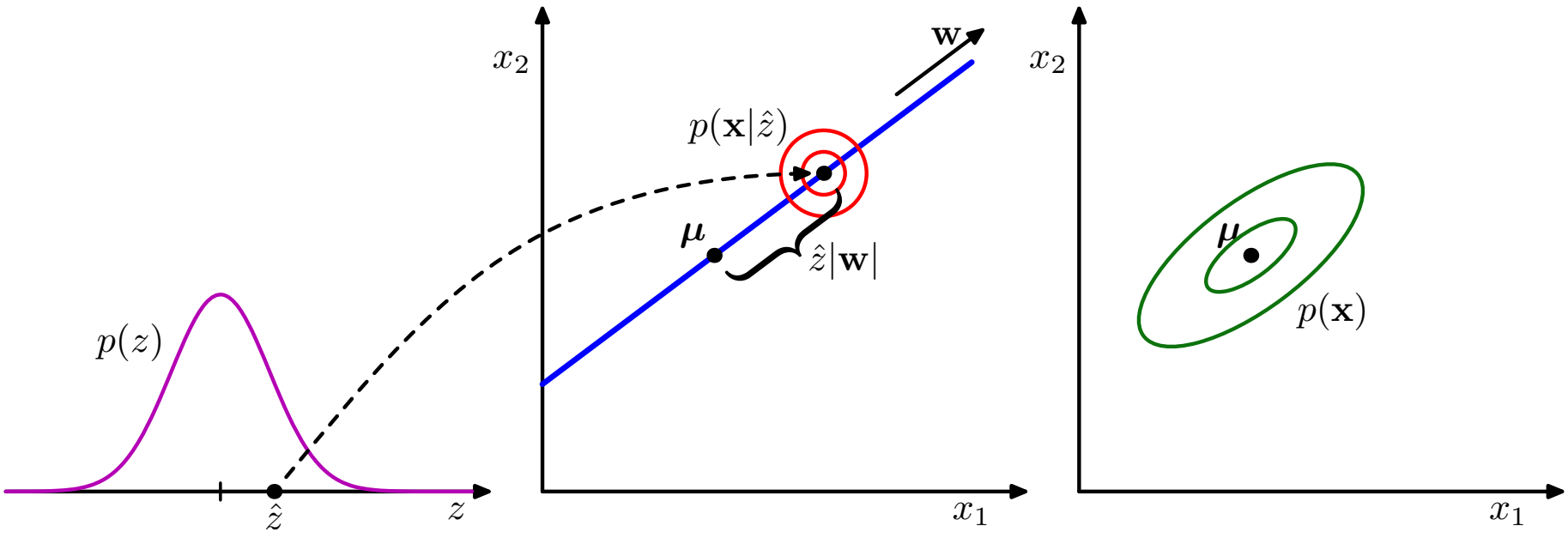
- Continuous latent variable:

$$\mathbf{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d).$$



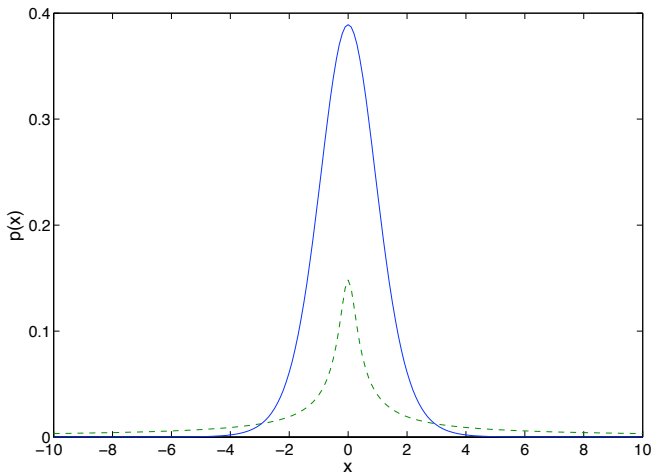
- ML estimate of the projection matrix: $\mathbf{W} = \mathbf{U}_d(\boldsymbol{\Lambda}_d - \sigma^2\mathbf{I}_d)^{1/2}\mathbf{R}$.
- ML estimate is equivalent to PCA solution up to a rotation $\mathbf{R}$.
- Residual variance σ^2 is given by $\frac{1}{D-d}\sum_{j>d}\lambda_j$.

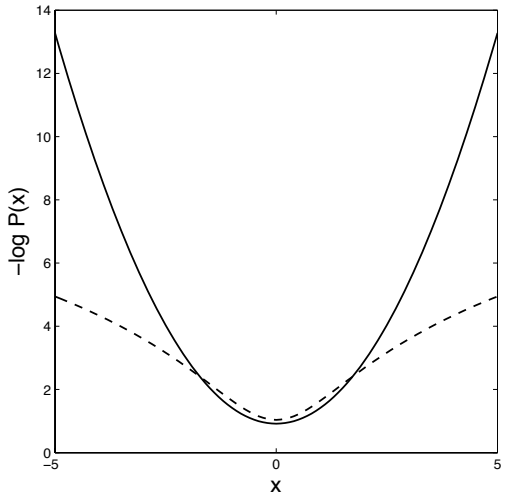
PPCA: interpretation



Robust probabilistic principal component analysis

- Many real noise sources are non-Gaussian.
- Models based on Gaussian noise are sensitive to outliers.
- A robust reformulation is based on the Student-t distribution:





- Replace Gaussian likelihood and Gaussian prior by scaled ones.
- Introduce the auxiliary (continuous) latent scale variable:

$$u_n \sim \mathcal{G}(\nu/2, \nu/2).$$

- An outlier is considered atypical in the observation and latent space.

Multivariate Student-t density

The Student-t density is defined as follows:[1]

$$\mathcal{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\Gamma(\frac{\nu+D}{2})}{\Gamma(\frac{\nu}{2})(\nu\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \left(1 + \frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)^{-\frac{\nu+D}{2}}.$$

Parameter $\nu > 0$ is the shape parameter:

- The Cauchy density is recovered for $\nu = 1$.
- The Gaussian density is recovered when $\nu \to \infty$.

The Student-t density can be reformulated as an infinite mixture of scaled Gaussians:

$$\mathcal{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \int_0^\infty \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}/u)\ \mathcal{G}(\tfrac{\nu}{2}, \tfrac{\nu}{2})\ du,$$

where u is a (latent) scale parameter.

[1] Student's t density was published in 1908 by *William S. Gosset*, while he worked at Guinness Brewery in Dublin and was not allowed to publish under his own name.

Gamma density

For $x \in \mathbb{R}^+$, the Gamma density is defined as follows:

$$\mathcal{G}(\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp\{-\beta x\}, \quad \alpha, \beta > 0,$$

where $\Gamma(u) \equiv \int_0^\infty v^{u-1} e^{-v} dv$ is the *gamma function*. We have

$$\langle x \rangle = a/b \quad \text{and} \quad \langle \ln x \rangle = \psi(a) - \ln b.$$

The function $\psi(\cdot) \equiv (\ln \Gamma)'(\cdot)$ is the *digamma* function.

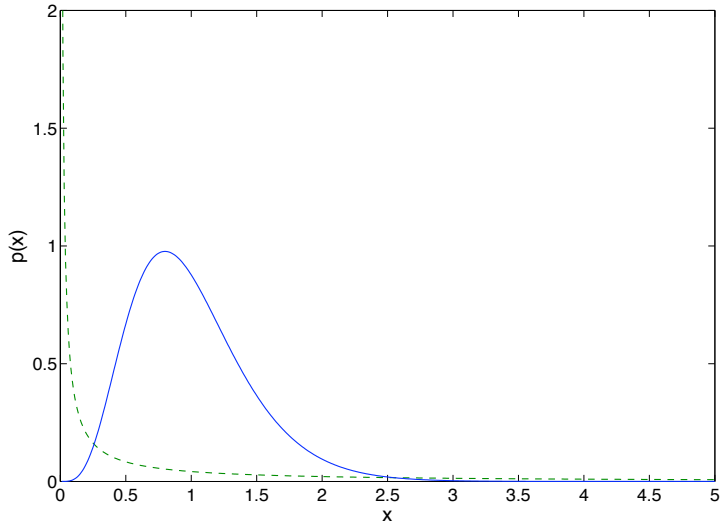
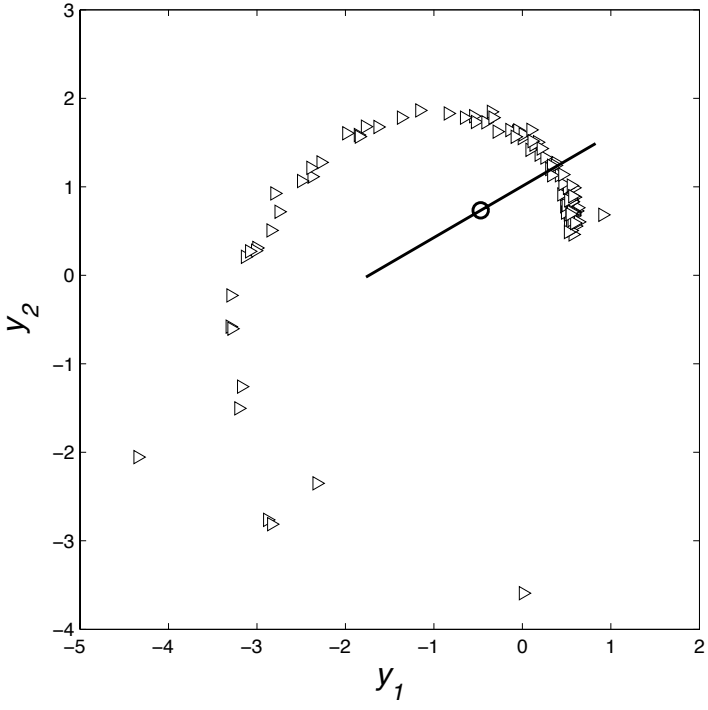
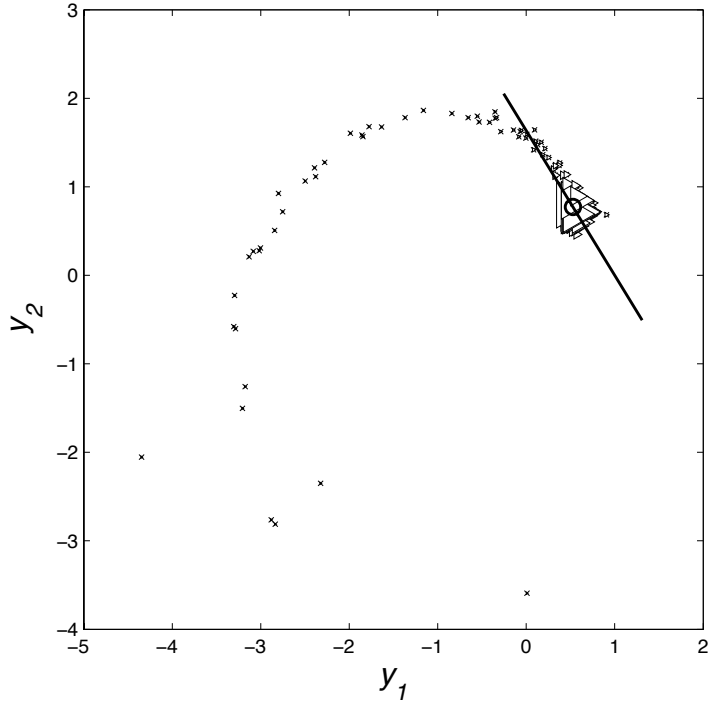


Figure: Gamma distribution for two values of a and b.

Example



(a) Standard PPCA.



(b) Robust PPCA.

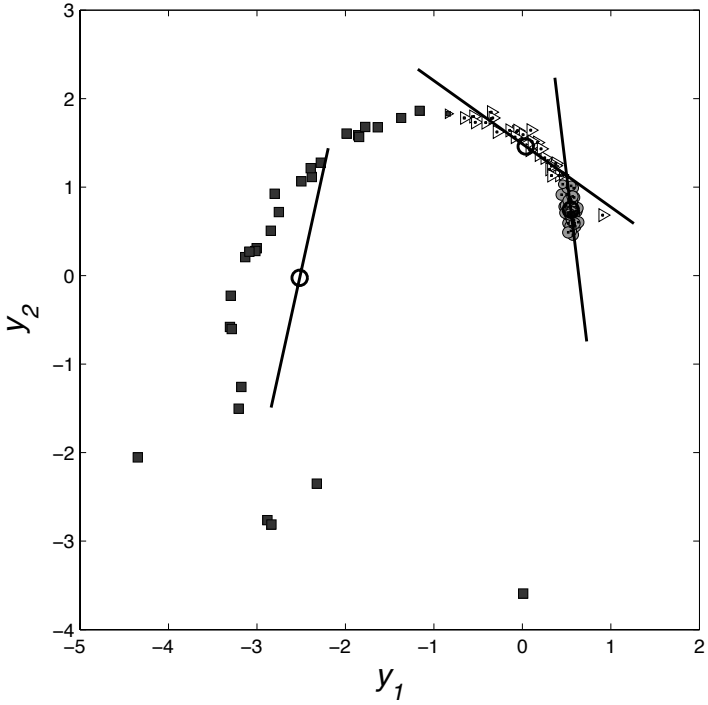
The dimension of the latent subspace is fixed in advance, but the shape parameter ν is learnt from the data (by a line search at each EM step).

Mixtures of probabilistic principal component analysers

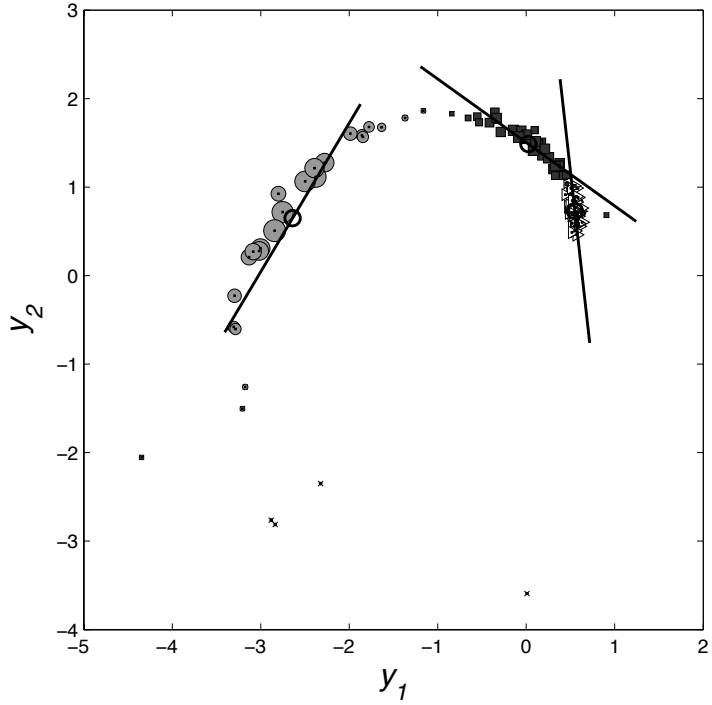
$$p(\mathbf{x}) = \sum_k \pi_k p(\mathbf{x}|z=k),$$
$$p(\mathbf{x}|z=k) = \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{W}_k\mathbf{W}_k^\top + \sigma^2\mathbf{I}_D).$$

- Clustering (very) high-dimensional data:
  - Stable due to low rank approximation of the covariance matrices.
  - Captures correlations between local leading directions.
  - Rotational ambiguity vanishes.
- The number of components and the dimension of the latent subspaces can be set by cross-validation.
- Combining local analysers to obtain nonlinear generative models.
- Possible issues are component misalignments and dimension mismatches.
- Natural extension to mixtures of robust PPCAs.

Finite mixture of (robust) PPCAs: example revisited



(c) Standard PPCA.

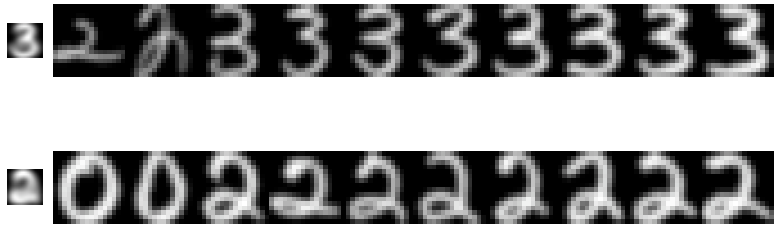


(d) Robust PPCA.

USPS handwritten digits 2 and 3

- USPS data set: 16×16 pixels images of digits (0 to 9).
- Only (respectively 731 and 658) images of digits 2 and 3 are kept.
- 100 (randomly chosen) images of digit 0.

- Mixture of PPCAs:



- Mixture of robust PPCAs:

- Standard mixture of Gaussians and diagonal mixtures collapse...

Part 3: Conditional models for regression

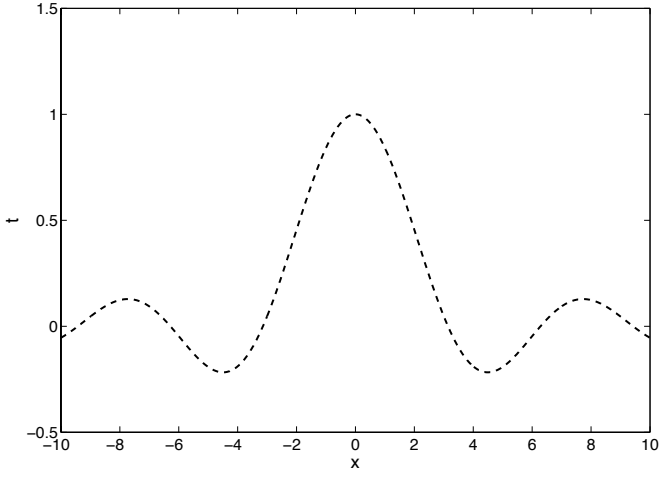
- Linear models for regression:
  - Maximum likelihood (vs. least squares)
  - Maximum a posteriori (vs. ridge regression)
  - Type II ML via EM
- Gaussian process regression

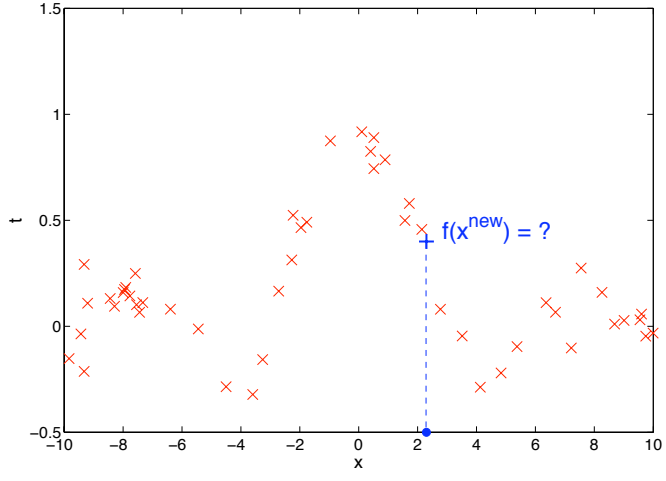
Conditional models for regression

- Consider a finite number of noisy observations $\{t_n\}_{n=1}^N$ associated to some input data $\{\mathbf{x}_n\}_{n=1}^N$.
- The conditional model $p(t|\mathbf{x})$ is not concerned with the density over the inputs.
- Conditional models for regression often assume iid noise:

$$t_n = f(\mathbf{x}_n) + \epsilon_n, \qquad \epsilon_n \sim \text{iid}.$$

- The goal is to predict the outcome $f(\mathbf{x}^*)$ of an unseen input $\mathbf{x}^*$ after having observed the training data $\{\mathbf{x}_n, t_n\}_{n=1}^N$.
- This is called generalisation.





Linear models for regression

- Let $\{\phi_m(\cdot)\}_{m=1}^M$ be a set of nonlinear basis functions centred on M learning prototypes.
- We assume $f(\mathbf{x}; \mathbf{w})$ is linear in the parameters $\mathbf{w}$:

$$f(\mathbf{x}; \mathbf{w}) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x}) + w_0 = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}).$$

- The goal is to learn $\mathbf{w}$ based on $\{\mathbf{x}_n, t_n\}_{n=1}^N$ so to predict at best on unseen data ($\sim$generalise).
- Well-known examples:
  - Least squares regression
  - Partial least squares
  - Regularization networks
  - Support vector machines
  - Splines
  - ...

Some notations

- Let $f_n = f(\mathbf{x}_n; \mathbf{w})$, $\mathbf{f} = (f_1, \ldots, f_N)^\top$ and $\mathbf{t} = (t_1, \ldots, t_N)^\top$.
- The design matrix $\mathbf{\Phi}$ is given by

$$\mathbf{\Phi} = \begin{pmatrix} 1 & \phi_1(\mathbf{x}_1) & \ldots & \phi_M(\mathbf{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_1(\mathbf{x}_N) & \ldots & \phi_M(\mathbf{x}_N) \end{pmatrix}.$$

- $\mathbf{\Phi}^\dagger = (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top$ is its Moore-Penrose pseudo-inverse.

A probabilistic view of least squares regression

- Assume observations are noisy iid samples drawn from a (univariate) Gaussian:

$$t_n = f_n + \epsilon_n, \quad \epsilon_n \sim \mathcal{N}(0, \sigma^2).$$

- The likelihood is then given by a multivariate Gaussian:

$$p(\mathbf{t}) = \prod_{n=1}^{N} \mathcal{N}(f_n, \sigma^2) = \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I}_N).$$

- Maximum likelihood leads to

$$\mathbf{w}_{\mathrm{ML}} = \mathbf{\Phi}^{\dagger}\mathbf{t}, \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\|\mathbf{t} - \mathbf{f}\|^2.$$

- The solution $\mathbf{w}_{\mathrm{ML}}$ is equal to the least squares solution:

$$\mathbf{w}_{\mathrm{ML}} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2}\sum_{n=1}^{N}(t_n - f_n)^2 = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2}\|\mathbf{t} - \mathbf{f}\|^2$$

- The solutoin σ^2_{ML} is to the residual error (or unexplained variance).

The log-likelihood is given by

$$\ln p(\mathbf{t}) = -\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln \sigma^2 - \frac{1}{2\sigma^2}\underbrace{(\mathbf{t}-\mathbf{f})^\top(\mathbf{t}-\mathbf{f})}_{=\|\mathbf{t}-\mathbf{f}\|^2}.$$

Hence, this leads to

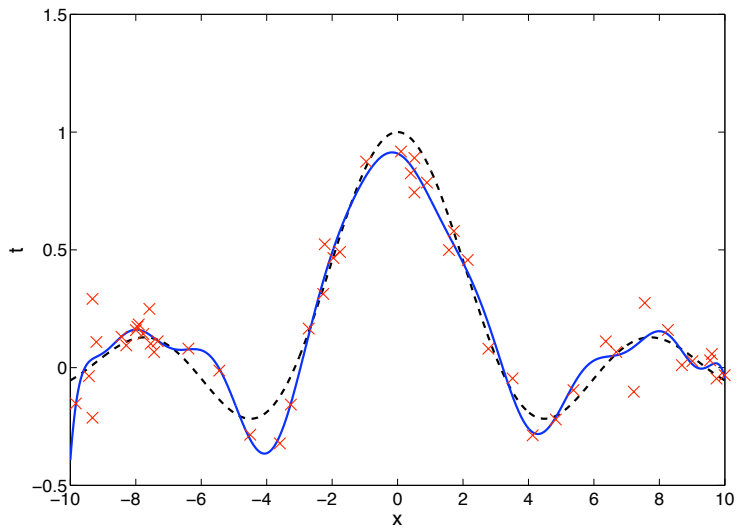
$$\frac{d\ln p(\mathbf{t})}{d\mathbf{w}} = 0 \quad \Rightarrow \quad \frac{1}{\sigma^2}\boldsymbol{\Phi}^\top(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w}) = 0,$$

$$\frac{d\ln p(\mathbf{t})}{d\sigma^2} = 0 \quad \Rightarrow \quad -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}(\mathbf{t}-\mathbf{f})^\top(\mathbf{t}-\mathbf{f}) = 0.$$

ML (or least squares) leads to overfitting

- Consider the target function $y(x) = \frac{\sin x}{x}$, $x \in [-10, 10]$.
- We choose the squared exponential basis function:

$$\phi_m(x) = \exp\left\{-\frac{\lambda_m}{2}(x - x_m)^2\right\}, \quad \lambda_m > 0.$$



- Solid blue curve is ML (least squares) solution for $\lambda_m = 1/36$.
- $\mathbf{\Phi}$ is often ill-conditioned and solving the linear system leads to overfitting (low bias, but high variance; too much flexibility!).

How to avoid overfitting?

- We model the uncertainty on the value of the parameters by imposing some prior distribution on them:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \mathbf{A}^{-1}),$$

  where $\mathbf{A} \equiv \mathrm{diag}\{\alpha_0, \ldots, \alpha_M\}$.
- In practice we will consider $\alpha_m = \alpha_0$ for all m.
- The goal is to favour regularised ($\sim$smooth) solutions by penalising large values of $\mathbf{w}$.

A probabilistic view of ridge regression (or weight decay)

- Maximum a posteriori (MAP) maximises the posterior distribution of the parameters:

$$p(\mathbf{w}|\mathbf{t}) \propto p(\mathbf{t}|\mathbf{w})p(\mathbf{w}).$$

- The MAP solution is given by

$$\mathbf{w}_{\mathrm{MAP}} = \sigma^{-2}(\sigma^{-2}\mathbf{\Phi}^\top\mathbf{\Phi} + \mathbf{A})^{-1}\mathbf{\Phi}^\top\mathbf{t},$$

  where the noise variance σ^2 is assumed to be known.
- Numerically unstable inversion of $\mathbf{\Phi}^\top\mathbf{\Phi}$ becomes stable thanks to $\mathbf{A}$.
- To learn the amount of noise, one has to use the EM (see later).
- MAP (for fixed σ) leads to the same solution as ridge regression:

$$\mathbf{w}_{\mathrm{MAP}} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2}\|\mathbf{t} - \mathbf{f}\|^2 + \frac{\alpha}{2}\|\mathbf{w}\|^2$$

  where $\alpha = \alpha_0\sigma^2$.

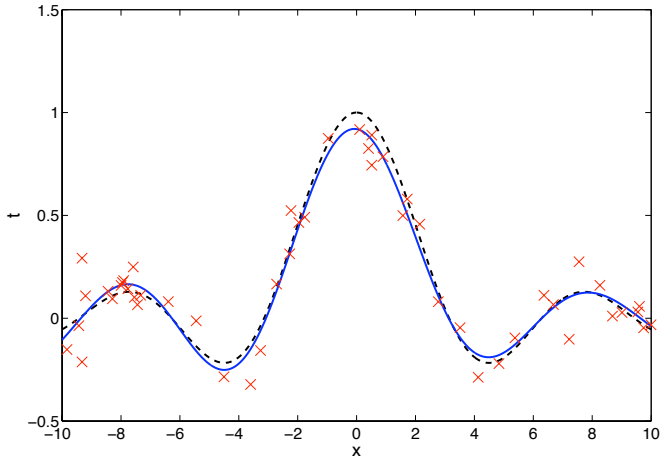
The log-posterior is given by

$$\begin{aligned}\ln p(\mathbf{w}|\mathbf{t}) = &-\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln \sigma^2 - \frac{1}{2\sigma^2}(\mathbf{t}-\mathbf{f})^\top(\mathbf{t}-\mathbf{f}) \\ &- \frac{M+1}{2}\ln 2\pi + \frac{1}{2}\ln|\mathbf{A}| - \frac{1}{2}\mathbf{w}^\top\mathbf{A}\mathbf{w} - \ln Z.\end{aligned}$$

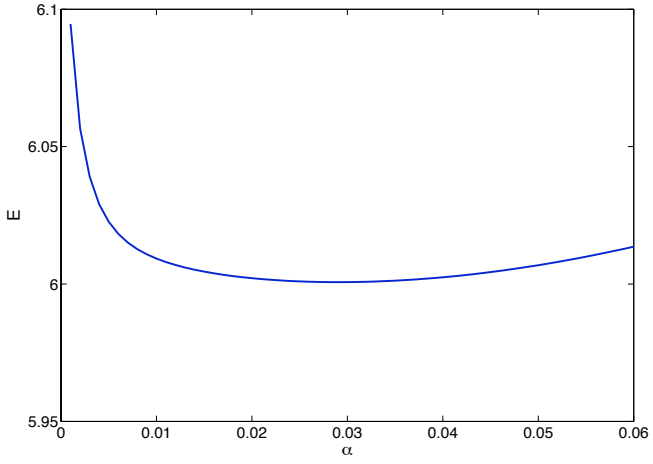
Hence, this leads to

$$\begin{aligned}\frac{d\ln p(\mathbf{w}|\mathbf{t})}{d\mathbf{w}} = 0 \quad &\Rightarrow \quad \frac{1}{\sigma^2}\boldsymbol{\Phi}^\top(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w}) - \mathbf{A}\mathbf{w} = 0 \\ &\Leftrightarrow \quad \frac{1}{\sigma^2}\boldsymbol{\Phi}^\top\mathbf{t} = \frac{1}{\sigma^2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\mathbf{w} + \mathbf{A}\mathbf{w}.\end{aligned}$$

Example revisited



(e)



(f)

Figure: (a) The target sinc function (dashed line) and the least squares regression solution (solid line) for $\lambda_m = 1/36$ for all m. The noisy observations are denoted by crosses. (b) Penalised error as a function of α.

Is the MAP solution a good solution?

- Overfitting is solved by limiting the effective model complexity.
- Model selection, i.e. choising the number of prototypes, is also solved, but it might be difficult to deal with (very) large data sets.
- The better ($\sim$smooth) solution is at the cost of an additional hyperparameter α, which can only be set by cross-validation.
- The residual noise σ^2 needs also to be set by cross-validation.
- The uncertainty on the parameters is not taken into account when making predictions (point estimate):

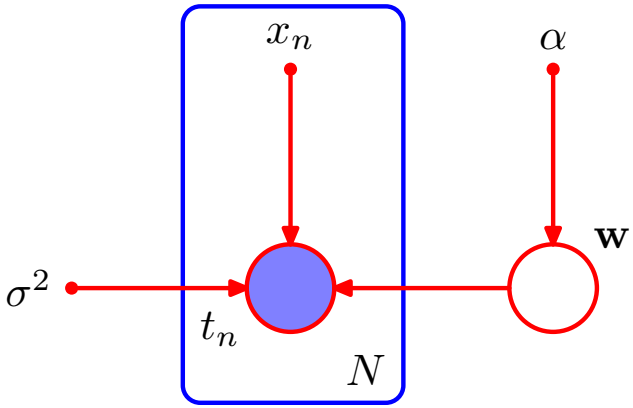
$$p(t|\mathbf{t}) \approx p(t|\mathbf{w}_{\mathrm{MAP}}) = \mathcal{N}(t|f(\mathbf{x};\mathbf{w}_{\mathrm{MAP}}), \sigma^2).$$

- The MAP solution depends on the parametrisation of the prior.

EM for linear regressors

- We view $\mathbf{w}$ as a latent variable on which an isotropic Gaussian prior is imposed:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \alpha_0^{-1}\mathbf{I}_{M+1}).$$



- The goal is to learn the noise variance σ^2 and the scale parameter α_0 via EM.
- EM applied to linear regressors is as follows

$$\mathcal{L}(q, \boldsymbol{\theta}) = \ln p(\mathbf{t}|\boldsymbol{\theta}) - \mathrm{KL}[q(\mathbf{w}) \| p(\mathbf{w}|\mathbf{t}, \boldsymbol{\theta})],$$
$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_q\{\ln p(\mathbf{t}, \mathbf{w}|\boldsymbol{\theta})\} + \mathrm{H}[q(\mathbf{w})].$$

  where $\boldsymbol{\theta} = \{\sigma, \alpha_0\}$.
- EM iteratively maximises the log-marginal likelihood:

$$\boldsymbol{\theta}_{\mathrm{ML2}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \ln p(\mathbf{t}|\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \ln \int p(\mathbf{t}, \mathbf{w}|\boldsymbol{\theta}) d\mathbf{w}.$$

- This procedure is known as type II ML (or evidence maximisation).

Type II ML for linear regressors (EM updates)

E step : compute posterior $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{w}, \boldsymbol{\Sigma}_\mathbf{w})$:[2]

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}_\mathbf{w}\boldsymbol{\Phi}^\top\mathbf{t},$$
$$\boldsymbol{\Sigma}_\mathbf{w} = (\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \alpha_0\mathbf{I}_{M+1})^{-1}.$$

M step : estimate residual noise and scale parameter:

$$\sigma^2_{\mathrm{ML2}} \leftarrow \frac{1}{N}\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}_\mathbf{w}\|^2 + \mathrm{tr}\{\boldsymbol{\Phi}\boldsymbol{\Sigma}_\mathbf{w}\boldsymbol{\Phi}^\top\},$$
$$\alpha_{\mathrm{ML2}} \leftarrow \frac{M+1}{\boldsymbol{\mu}_\mathbf{w}^\top\boldsymbol{\mu}_\mathbf{w} + \mathrm{tr}\{\boldsymbol{\Sigma}_\mathbf{w}\}}.$$

[2] The posterior mean $\boldsymbol{\mu}_\mathbf{w}$ is equal to the MAP estimate of $\mathbf{w}$ (why?).

The posterior is given by (completing the square)

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{t}) &\propto e^{-\frac{1}{2\sigma^2}(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})^\top(\mathbf{t}-\boldsymbol{\Phi}\mathbf{w})}\; e^{-\frac{\alpha_0}{2}\mathbf{w}^\top\mathbf{w}} \\
&\propto e^{-\frac{1}{2}(\mathbf{w}^\top(\sigma^{-2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}+\alpha_0\mathbf{I}_{M+1})\mathbf{w}-2\sigma^{-2}\mathbf{t}^\top\boldsymbol{\Phi}\mathbf{w})} \\
&\propto e^{-\frac{1}{2}(\mathbf{w}^\top\boldsymbol{\Sigma}_\mathbf{w}^{-1}\mathbf{w}-2\boldsymbol{\mu}_\mathbf{w}^\top\boldsymbol{\Sigma}_\mathbf{w}^{-1}\mathbf{w})} \\
&\propto e^{-\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu}_\mathbf{w})^\top\boldsymbol{\Sigma}_\mathbf{w}^{-1}(\mathbf{w}-\boldsymbol{\mu}_\mathbf{w})}.
\end{aligned}
$$

The expected complete log-likelihood is given by

$$
\begin{aligned}
\langle \ln p(\mathbf{t},\mathbf{w})\rangle &= -\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\left\langle(\mathbf{t}-\mathbf{f})^\top(\mathbf{t}-\mathbf{f})\right\rangle \\
&\quad - \frac{M+1}{2}\ln 2\pi + \frac{M+1}{2}\ln\alpha_0 - \frac{\alpha_0}{2}\langle\mathbf{w}^\top\mathbf{w}\rangle \\
&= -\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(\mathbf{t}-\boldsymbol{\Phi}\langle\mathbf{w}\rangle)^\top(\mathbf{t}-\boldsymbol{\Phi}\langle\mathbf{w}\rangle) \\
&\quad - \frac{1}{2\sigma^2}\mathrm{tr}\{\boldsymbol{\Phi}\boldsymbol{\Sigma}_\mathbf{w}\boldsymbol{\Phi}^\top\} - \frac{M+1}{2}\ln 2\pi + \frac{M+1}{2}\ln\alpha - \frac{\alpha}{2}\langle\mathbf{w}^\top\mathbf{w}\rangle.
\end{aligned}
$$

Taking the derivative wrt σ^2 and α_0, and equating to zero leads to the desired updates.

Predictive distributions

- We are not only interested in the optimal predictions, but also in the best approximation of the full predictive distribution.
- The predictive distributions for the ML and the type II ML solutions are given by

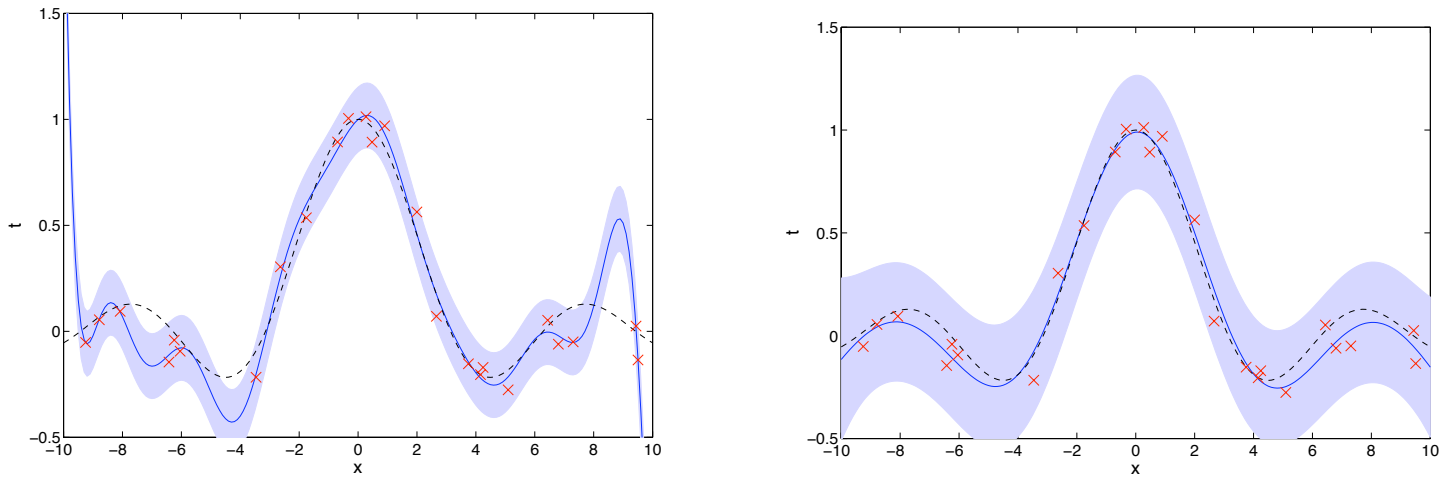
$$p(t|\mathbf{t}) \approx p(t|\mathbf{w}_{\mathrm{ML}}, \sigma_{\mathrm{ML}}) = \mathcal{N}(\mathbf{w}_{\mathrm{ML}}^{\top}\boldsymbol{\phi}(\mathbf{x}), \sigma_{\mathrm{ML}}^2),$$

$$p(t|\mathbf{t}) \approx p(t|\mathbf{t}, \sigma_{\mathrm{ML2}}, \alpha_{\mathrm{ML2}}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}}^{\top}\boldsymbol{\phi}(\mathbf{x}), \sigma_{\mathrm{ML2}}^2 + \boldsymbol{\phi}^{\top}(\mathbf{x})\boldsymbol{\Sigma}_{\mathbf{w}}\boldsymbol{\phi}(\mathbf{x})).$$

- In the case of type II ML, the predictive variance has two components:
  - One due to the noise on the data.
  - One due to the uncertainty on the parameters.

Example revisited

We compare the solutions on the sinc example with $N = 25$, $\sigma = 0.1$ and $\lambda_m = 1/9$ for all m. We show the mean and the error bars (± 3 std):

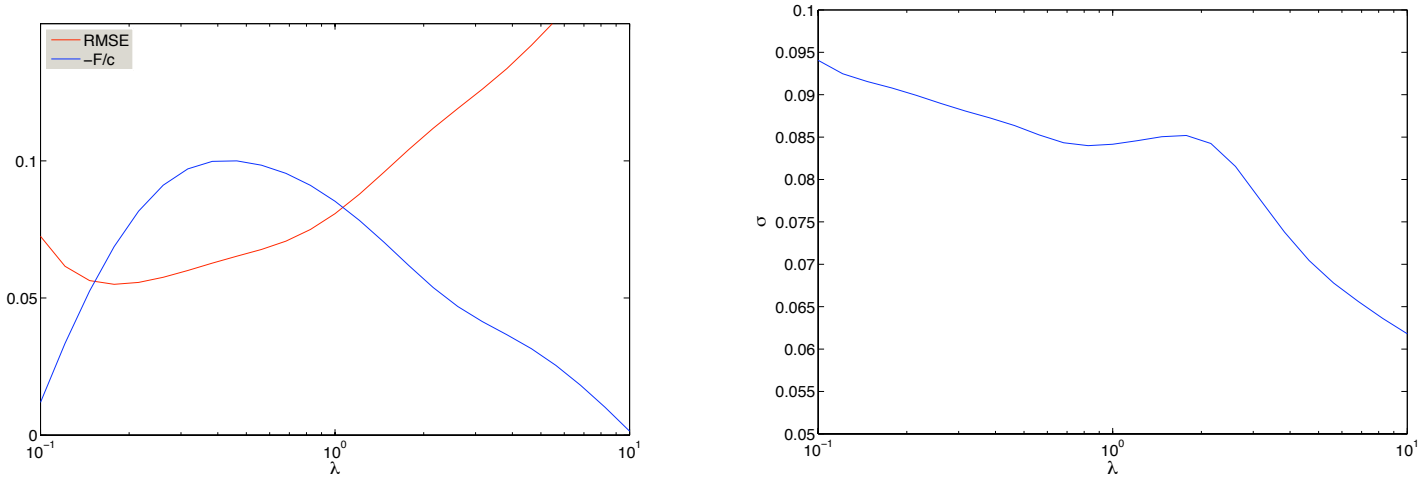


(a) ML. (b) ML2.

Figure: (a) ML solution: $\sigma_{\mathrm{ML}} = 0.05$. (b) Type II ML solution: $\sigma_{\mathrm{ML2}} = 0.08$ and $\alpha_{\mathrm{ML2}} = 1.15$. (Target function: dashed; observations: crosses.)

Is the type II ML solution a good solution?

- Overfitting is avoided by taking parameter uncertainty into account.
- Integrating out **w** leads to confidence measures for predictions.
- The variational bound is not suitable for selecting the kernel width:



(a) (b)

Figure: (a) Root mean square error (RMSE) and normalised lower bound ($-\mathcal{F}/c$) versus the kernel width λ. (b) Noise standard deviation versus λ.

From Bayesian linear regression to Gaussian processes

- Bayesian linear model for regression:

$$t_n = f_n + \epsilon_n, \qquad \epsilon_n \sim \mathcal{N}(0, \sigma^2),$$
$$f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}), \qquad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I}).$$

- Integrating out the parameter vector $\mathbf{w}$ leads to

$$\mathbb{E}\{f(\mathbf{x}; \mathbf{w})\} = \mathbb{E}\{\mathbf{w}^\top\}\boldsymbol{\phi}(\mathbf{x}) = 0,$$
$$\mathbb{E}\{f(\mathbf{x}; \mathbf{w})f(\mathbf{x}'; \mathbf{w})\} = \boldsymbol{\phi}(\mathbf{x})^\top \mathbb{E}\{\mathbf{w}\mathbf{w}^\top\}\boldsymbol{\phi}(\mathbf{x}') = \alpha^{-1}\sum_k \phi_k(\mathbf{x})\phi_k(\mathbf{x}'),$$

- The prior over functions is entirely determined by the mean function $m(\mathbf{x}) = 0$ and the covariance function $k(\mathbf{x}, \mathbf{x}') = \sum_k \phi_k(\mathbf{x})\phi_k(\mathbf{x}')$.
- This GP has a finite number of basis functions (implicit kernel).
- The idea is to define the covariance function (or kernel) directly.

Example of a covariance function

The **squared exponential kernel** is defined as

$$k(\mathbf{x}, \mathbf{x}') = c^2 \exp\left\{-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right\},$$

where $c \geqslant 0$ and $l > 0$ are hyperparameters.

- Valid kernel function as for any set $\{\mathbf{x}_n\}_{n=1}^N$, the kernel matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ is **positive semidefinite**
- Depends only on the difference $\mathbf{x} - \mathbf{x}'$, i.e. it is a **stationary kernel**
- Corresponds to projecting the input data into an infinite dimensional feature space (see e.g. *Shawe-Taylor and Cristianini*, 2004)
- Corresponds to using an **infinite number** of basis functions (not just on the training points)

Gaussian process

A **multivariate Gaussian** distribution:

- Defines a probability density over D random variables (based on correlations).
- Characterized by **mean vector** and **covariance matrix**:

$$\mathbf{f} \equiv (f_1, \ldots, f_D)^\top \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

A **Gaussian process** (GP) is a generalization of a multivariate Gaussian distribution to **infinitely many** variables.

- Defines a probability measure over **random functions** (Informally a function can be viewed as an infinitely long vector.)
- Characterized by **mean function** and **covariance function**:

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

- The (joint) marginal distribution over any **finite subset** of variables is a consistent **finite dimensional Gaussian**!

Sampling random functions from GPs

Batch sampling: $\mathbf{f} \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$

- Generate a set of inputs $\{\mathbf{x}_n\}_{n=1}^N$.
- Draw N samples from $\mathcal{N}(0, 1)$.
- Compute the function values using $\mathbf{f} = \mathbf{L}^\top \mathbf{z} + \mathbf{m}$, where $\mathbf{L}$ is the upper triangular Cholesky factor of the kernel matrix $\mathbf{K}$.

Sequential sampling: $\mathbf{f} \sim \prod_{n>0} p(f_n|\mathbf{f}_{\setminus n}) = \prod_{n>0} \mathcal{N}(\widetilde{m}_n, \widetilde{\sigma}_n^2)$

Repeat for $n > 0$:

- Generate $\mathbf{x}_n$.
- Draw a sample z_n from $\mathcal{N}(0, 1)$.
- Compute the function value associated to $\mathbf{x}_n$ using $f_n = \tilde{\sigma}_n z_n + \widetilde{m}_n$.

The function values f_n and $\mathbf{y}_{\setminus n} = (f_{n-1}, \ldots, f_1)^\top$ are jointly Gaussian:

$$p(f_n, \mathbf{f}_{\setminus n}) = \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_n) \\ \mathbf{m}_{\setminus n} \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_n, \mathbf{x}_n) & \mathbf{k}_n \\ \mathbf{k}_n^\top & \mathbf{K}_{\setminus n} \end{bmatrix}\right) = \mathcal{N}(\mathbf{m}, \mathbf{K}).$$

The conditional $p(f_n|\mathbf{f}_{\setminus n})$ is then also Gaussian with the **conditional mean** and the **conditional variance** respectively given by

$$\widetilde{m}_n = m(\mathbf{x}_n) + \mathbf{k}_n \mathbf{K}_{\setminus n}^{-1}(\mathbf{f}_{\setminus n} - \mathbf{m}_{\setminus n}),$$
$$\widetilde{\sigma}_n^2 = k(\mathbf{x}_n, \mathbf{x}_n) - \mathbf{k}_n \mathbf{K}_{\setminus n}^{-1} \mathbf{k}_n^\top.$$

Example

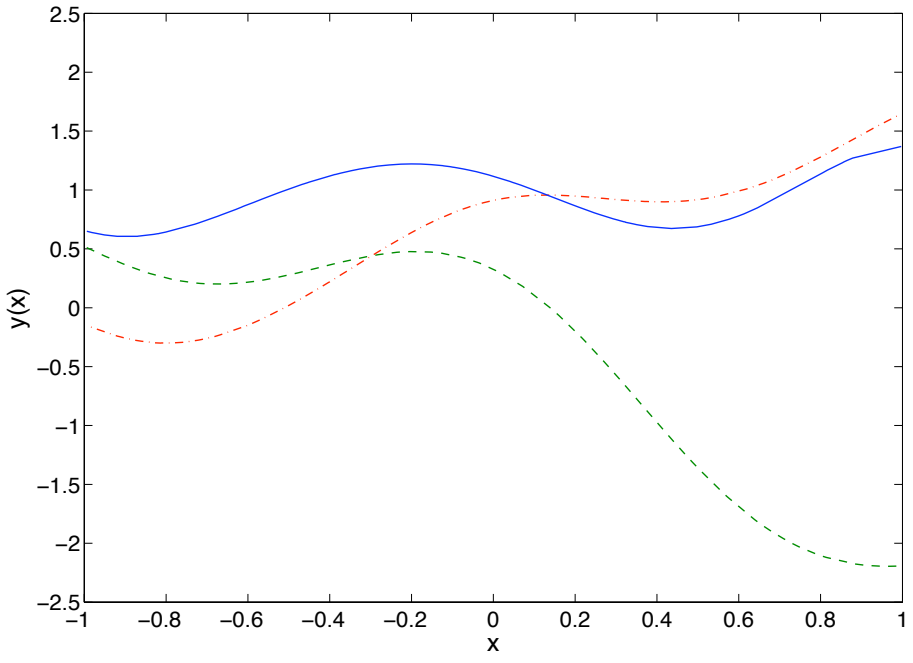


Figure: Three random functions generated from a $\mathcal{GP}$ with $m(x) = 0$ and a squared exponential covariance function ($c = 1$ and $l = 0.5$).

Gaussian processes for regression

- The kernel defines a prior over function space:

$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)).$$

- We assume a finite number of observations and iid Gaussian noise:

$$\mathbf{t}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I}_N),$$

  where $\mathbf{f} \equiv (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^\top$ are the latent function values.
- The **posterior process** is again a Gaussian process:

$$f(\cdot)|\mathbf{t} \sim \mathcal{GP}(\widetilde{m}(\cdot), \tilde{k}(\cdot, \cdot)),$$

  where

$$\widetilde{m}(\cdot) = \mathbf{k}^\top(\cdot)(\mathbf{K} + \sigma^2 \mathbf{I}_N)^{-1}\mathbf{t},$$
$$\tilde{k}(\cdot, \cdot) = k(\cdot, \cdot) - \mathbf{k}^\top(\cdot)(\mathbf{K} + \sigma^2 \mathbf{I}_N)^{-1}\mathbf{k}(\cdot).$$

Any latent function value $f(\mathbf{x})$ is jointly Gaussian with the finite subset $\mathbf{f}$:

$$p(\mathbf{f}, f(\mathbf{x})) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{k}(\mathbf{x}) \\ \mathbf{k}^\top(\mathbf{x}) & k(\mathbf{x},\mathbf{x}) \end{bmatrix}\right),$$

where $\mathbf{k}(\mathbf{x}) \equiv (k(\mathbf{x},\mathbf{x}_1),\ldots,k(\mathbf{x},\mathbf{x}_N))^\top$.

The mean and the variance of the conditional Gaussian $p(f(\mathbf{x})|\mathbf{f})$ are given by

$$\mu(\mathbf{x}) = \mathbf{k}^\top(\mathbf{x})\mathbf{K}^{-1}\mathbf{y},$$
$$\kappa(\mathbf{x},\mathbf{x}) = k(\mathbf{x},\mathbf{x}) - \mathbf{k}^\top(\mathbf{x})\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}).$$

We have the $p(\mathbf{y}) = \mathcal{N}(\mathbf{0},\mathbf{K})$ and the $p(\mathbf{t}|\mathbf{y}) = \mathcal{N}(\mathbf{f},\sigma^2\mathbf{I}_N)$, such that

$$p(\mathbf{f}|\mathbf{t}) = \mathcal{N}(\sigma^{-2}\mathbf{\Sigma}\mathbf{t},\mathbf{\Sigma}),$$

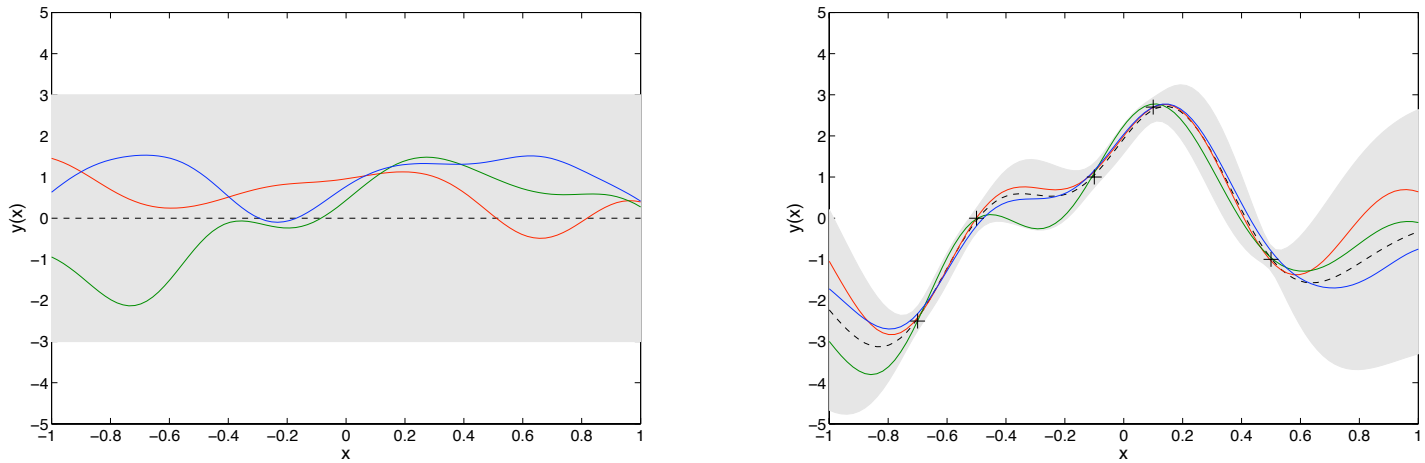
where $\mathbf{\Sigma} = (\mathbf{K}^{-1} + \sigma^{-2}\mathbf{I}_N)^{-1}$.

Hence, the marginal posterior $p(f(\mathbf{x})|\mathbf{t}) = \int p(f(\mathbf{x})|\mathbf{f})p(\mathbf{f}|\mathbf{t})d\mathbf{f}$ is a Gaussian with mean and variance given by

$$\widetilde{m}(\mathbf{x}) = \mathbf{k}^\top(\mathbf{x})(\mathbf{K}+\sigma^2\mathbf{I}_N)^{-1}\mathbf{t},$$
$$\tilde{k}(\mathbf{x},\mathbf{x}) = k(\mathbf{x},\mathbf{x}) - \mathbf{k}^\top(\mathbf{x})(\mathbf{K}+\sigma^2\mathbf{I}_N)^{-1}\mathbf{k}(\mathbf{x}),$$

where the *Woodbury identity* was invoked.

Example



(a) Prior. (b) Posterior.

Figure: Three random functions generated from (a) the prior $\mathcal{GP}$ and (b) the posterior $\mathcal{GP}$. An observation is indicated by a $+$, the mean function by a dashed line and the 3 standard deviation error bars by the shaded regions. We used a squared exponential covariance function ($c = 1$ and $l = 0.25$).

Learning the parameters by type II ML

- We view the latent functions as nuisance parameters (thus integrate them out).
- The **log-marginal likelihood** is given by

$$\ln p(\mathbf{t}) = -\frac{N}{2}\ln 2\pi \underbrace{-\frac{1}{2}\ln|\mathbf{K}(\boldsymbol{\theta})+\sigma^2\mathbf{I}_N|}_{\text{complexity penality}} \underbrace{-\frac{1}{2}\mathbf{t}^\top(\mathbf{K}(\boldsymbol{\theta})+\sigma^2\mathbf{I}_N)^{-1}\mathbf{t}}_{\text{data fit}}.$$

- The noise variance σ^2 and the kernel parameters $\boldsymbol{\theta}$ can be learned by means of **gradient ascent** techniques:

$$\frac{\partial \ln p(\mathbf{t})}{\partial \sigma^2} = -\frac{1}{2}\mathrm{tr}\left\{(\mathbf{K}+\sigma^2\mathbf{I}_N)^{-1}\right\} + \frac{1}{2}\boldsymbol{\nu}^\top\boldsymbol{\nu},$$
$$\frac{\partial \ln p(\mathbf{t})}{\partial \theta_k} = -\frac{1}{2}\mathrm{tr}\left\{\left((\mathbf{K}+\sigma^2\mathbf{I}_N)^{-1}-\boldsymbol{\nu}\boldsymbol{\nu}^\top\right)\frac{\partial \mathbf{K}}{\partial \theta_k}\right\},$$

where $\boldsymbol{\nu} \equiv (\mathbf{K}(\boldsymbol{\theta})+\sigma^2\mathbf{I}_N)^{-1}\mathbf{t}$.

Computational considerations

- No need to explicitly invert the kernel matrix (no numerical instabilities), but $\mathbf{K} + \sigma^2 \mathbf{I}_N$.
- Use of standard nonlinear optimisation tools, but objective is **non-convex** (no guarantee of attaining a global maximum).
- Main bottleneck is the computational complexity of the kernel matrix, which is $\mathcal{O}(N^3)$ for a training set of size N.
- The computation of the derivatives requires only time $\mathcal{O}(N^2)$ per hyperparameter.

Useful tricks include:

- Matrix inversion lemma (Woodbury identity):

$$(\boldsymbol{\Psi} + \mathbf{V}\boldsymbol{\Phi}\mathbf{W})^{-1} = \boldsymbol{\Psi}^{-1} - \boldsymbol{\Psi}^{-1}\mathbf{V}(\boldsymbol{\Phi}^{-1} + \mathbf{W}\boldsymbol{\Psi}^{-1}\mathbf{V})^{-1}\mathbf{W}\boldsymbol{\Psi}^{-1},$$
$$|\boldsymbol{\Psi} + \mathbf{V}\boldsymbol{\Phi}\mathbf{W}| = |\boldsymbol{\Psi}|\,|\boldsymbol{\Phi}|\,|\boldsymbol{\Phi}^{-1} + \mathbf{W}\boldsymbol{\Psi}^{-1}\mathbf{V}|,$$

  where $\boldsymbol{\Psi} \in \mathbb{R}^{N\times N}$, $\boldsymbol{\Phi} \in \mathbb{R}^{M\times M}$, $\mathbf{V} \in \mathbb{R}^{N\times M}$ and $\mathbf{W} \in \mathbb{R}^{M\times N}$.
- Cholesky decomposition $\boldsymbol{\Lambda} = \mathbf{Q}^\top\mathbf{Q}$, where the Cholesky factor $\mathbf{Q} \in \mathbb{R}^{D\times D}$ is upper triangular.

Predictive distribution

The predictive distribution at $\mathbf{x}$ for type II ML estimates of the hyperparameters is given by

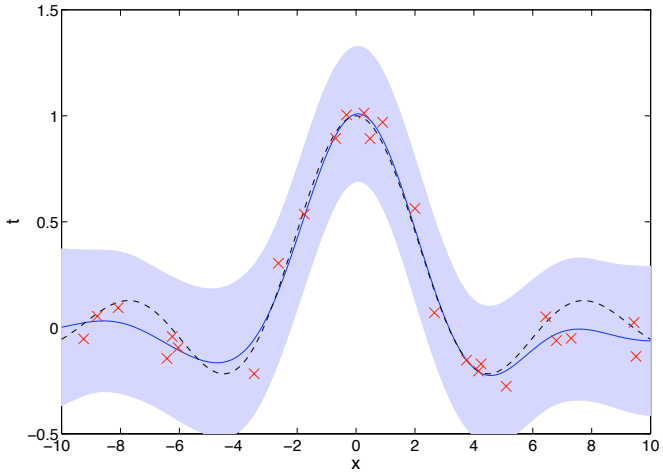
$$p(t|\mathbf{t}) \approx p(t|\mathbf{t}, \sigma_{\mathrm{ML2}}, \boldsymbol{\theta}_{\mathrm{ML2}}) = \mathcal{N}(\widetilde{m}_{\mathrm{ML2}}(\mathbf{x}), \tilde{k}_{\mathrm{ML2}}(\mathbf{x}, \mathbf{x}) + \sigma^2_{\mathrm{ML2}}).$$

The predictive variance has three components:

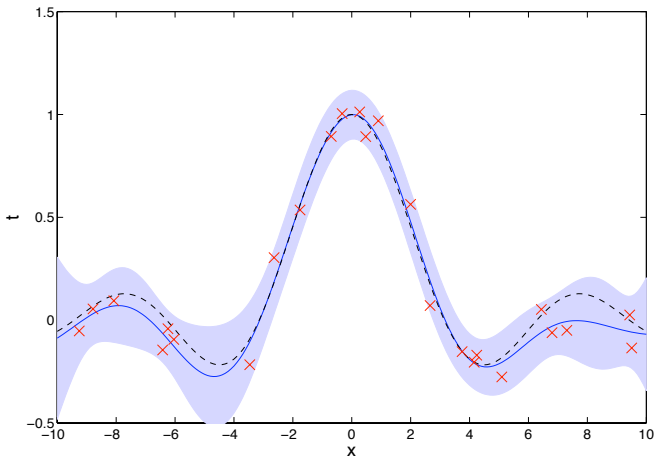
- The prior variance $k_{\mathrm{ML2}}(\mathbf{x}, \mathbf{x})$.
- The term $-\mathbf{k}^\top_{\mathrm{ML2}}(\mathbf{x})(\mathbf{K}_{\mathrm{ML2}} + \sigma^2_{\mathrm{ML2}}\mathbf{I}_N)^{-1}\mathbf{k}_{\mathrm{ML2}}(\mathbf{x})$, which reduces the prior uncertainty and tells us how much is explained by the data.[3]
- The noise σ^2_{ML2} on the observations.

[3] This term is independent of the targets!

Sinc example



(a) Variational linear regression.



(b) $\mathcal{GP}$ regression.

Figure: Comparison of the optimal solutions found by (a) variational linear regression with squared exponential basis functions ($\lambda = 0.4095$) and by (b) Gaussian process regression with a squared exponential kernel ($\lambda = 0.1804$).

Covariance functions

A valid kernel should satisfy *Mercer's condition* (see e.g. *Shawe-Taylor and Cristianini*, 2004).

In practice we require the kernel to induce a **symmetric** and **positive semidefinite** kernel matrix.

Examples of other kernels:

- Non-stationary kernels (e.g. sigmoidal kernel).
- Kernels for structured inputs (e.g. string kernels).
- Some rules for **kernel design**:

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= c k_1(\mathbf{x}, \mathbf{x}'), \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}'), \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}'), \\
k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}'), \\
&\vdots
\end{aligned}
$$

where $c > 0$ is a constant and $f(\cdot)$ is a deterministic function.

Periodic covariance functions

- Construct a periodic signal of x:

$$\mathbf{u}(x) = (\sin x, \cos x)^\top.$$

- Plug into the squared exponential kernel:

$$k(\mathbf{x}, \mathbf{x}') = c^2 \exp\left\{ -\frac{2\sin^2\left(\frac{x-x'}{2}\right)}{l^2} \right\},$$

where we used $\|\mathbf{u}(x) - \mathbf{u}(x')\|^2 = 4\sin^2(\frac{x-x'}{2})$.

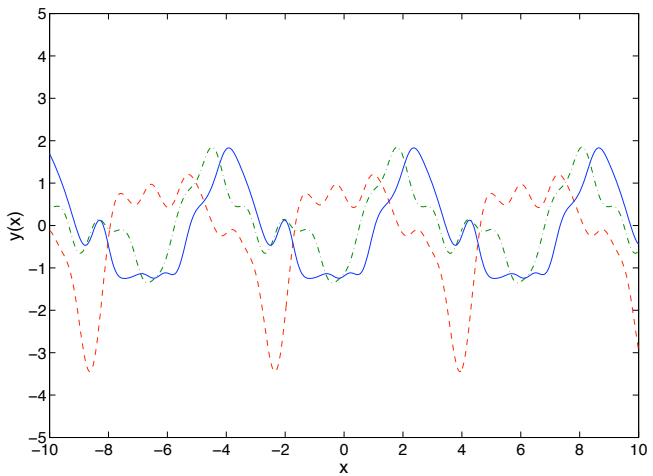


Figure: 3 random functions generated with a periodic kernel ($c = 1$, $l = 0.5$).

Rational quadratic covariance functions

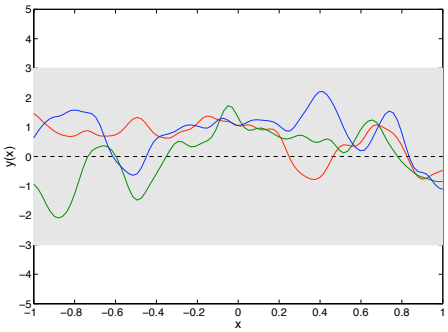
$$k(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\nu l^2}\right)^{-\frac{\nu+D}{2}},$$

- $\nu > 0$ is the shape parameter, $l > 0$ the scale parameter and D is the dimension of the input space.
- The kernel corresponds to an infinite mixture of scaled squared exponentials:

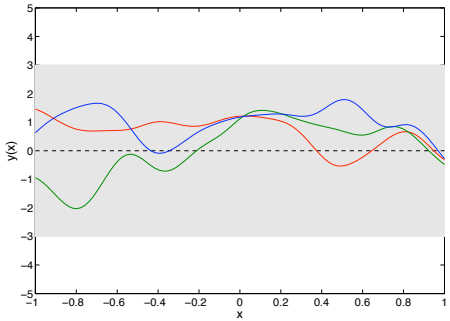
$$\int p(r|u, l)\ p(u|\nu)\ du = \int \mathcal{N}(0, l^2/u)\mathcal{G}(\tfrac{\nu}{2}, \tfrac{\nu}{2})\ du$$
$$\propto \left(1 + \frac{r^2}{\nu l^2}\right)^{-\frac{\nu+D}{2}}.$$

  where $r \equiv \|\mathbf{x} - \mathbf{x}'\|$.
- The shape parameter ν defines the **thickness** of the kernel tails. The squared exponential is recovered for $\nu \to \infty$.

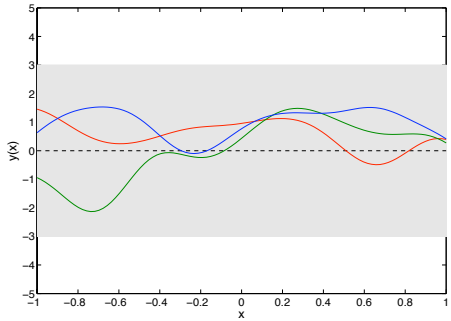
Example revisited



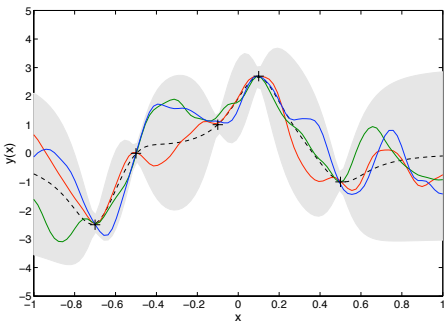
(a) Prior $\nu = \frac{1}{3}$.



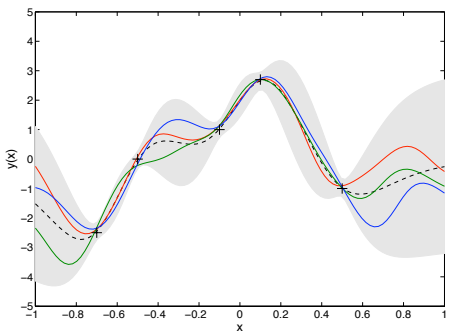
(b) Prior $\nu = 3$.



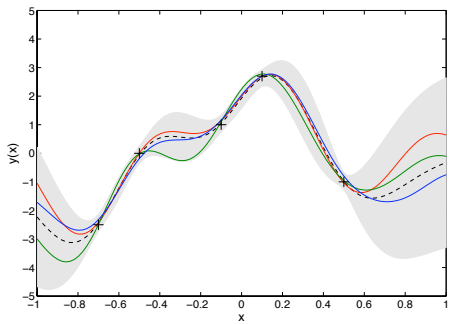
(c) Prior $\nu \to \infty$.



(d) Prior $\nu = \frac{1}{3}$.



(e) Prior $\nu = 3$.



(f) Prior $\nu \to \infty$.

Figure: Three random functions generated from (a) the prior $\mathcal{GP}$ and (b) the posterior $\mathcal{GP}$ with the rational quadratic kernel ($l = 0.25$). The observations are indicated by $+$, the means by a dashed lines and the 3 standard deviation error bars by the shaded regions.

Matérn covariance functions

$$k(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\ |\mathbf{x} - \mathbf{x}'|}{l} \right)^{\nu} \mathrm{K}_{\nu} \left(\frac{\sqrt{2\nu}\ |\mathbf{x} - \mathbf{x}'|}{l} \right),$$

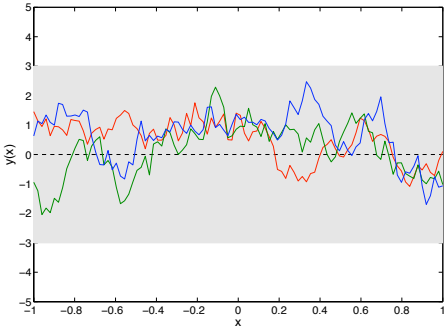
- $\mathrm{K}_{\nu}(\cdot)$ is the *modified Bessel function of the second kind*, $\nu > 0$ and $l > 0$.
- The order ν defines the **roughness** of the random functions as they are $\lfloor \nu - 1 \rfloor$ times differentiable:
  - We have the **Laplacian** or **Ornstein-Uhlenbeck**[4] kernel for $\nu = \frac{1}{2}$.
  - For $\nu = p + \frac{1}{2}$ with $p \in \mathbb{N}$, the covariance function takes the simple form of a product of an exponential and a polynomial of order p.

$$k(\mathbf{x}, \mathbf{x}') = \exp\left\{ -\frac{\sqrt{2\nu}\ |\mathbf{x} - \mathbf{x}'|}{l} \right\} \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\nu}\ |\mathbf{x} - \mathbf{x}'|}{l} \right)^{p-i}.$$

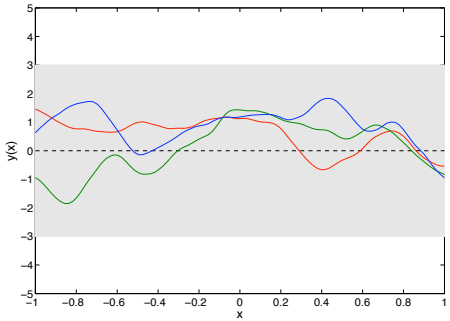
  - We recover the **squared exponential** kernel for $\nu \to \infty$.
- There is no closed form solution for the derivative of $\mathrm{K}_{\nu}(\cdot)$ wrt ν.

[4] The Ornstein-Uhlenbeck (OU) process is a mathematical description of the velocity of a particle undergoing *Brownian motion.*

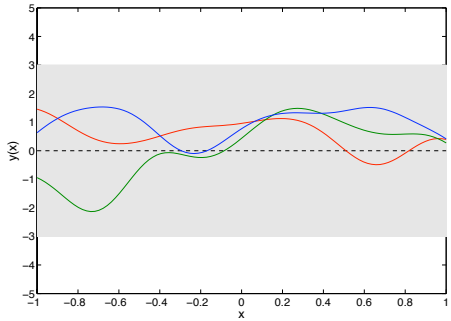
Example revisited



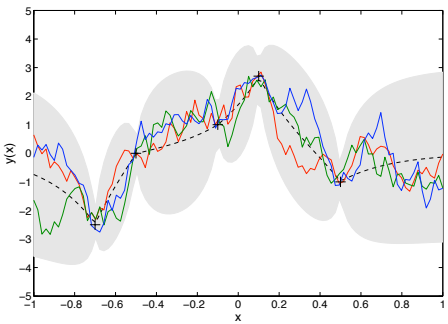
(a) Prior $\nu = \frac{1}{2}$.



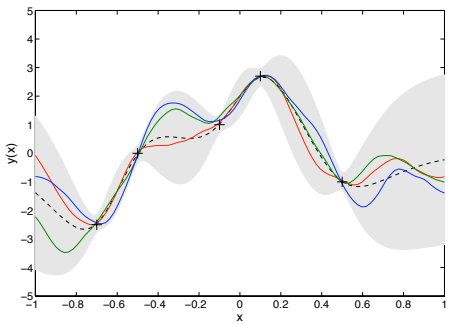
(b) Prior $\nu = \frac{5}{2}$.



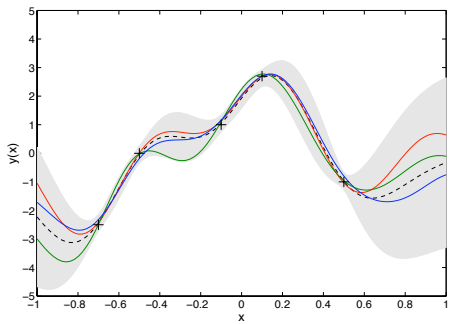
(c) Prior $\nu \to \infty$.



(d) Prior $\nu = \frac{1}{2}$.



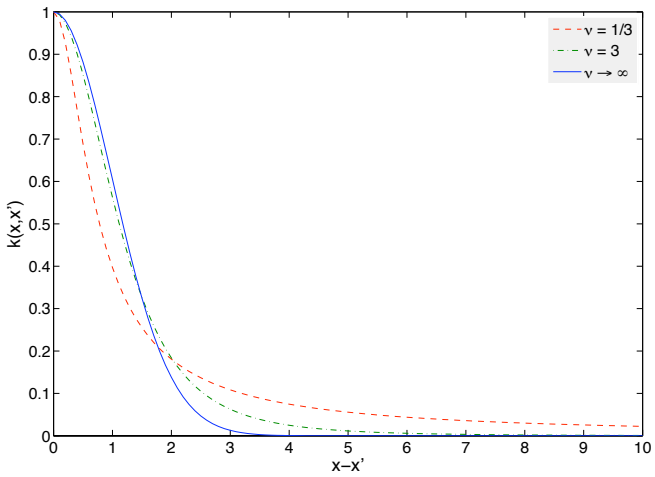
(e) Prior $\nu = \frac{5}{2}$.



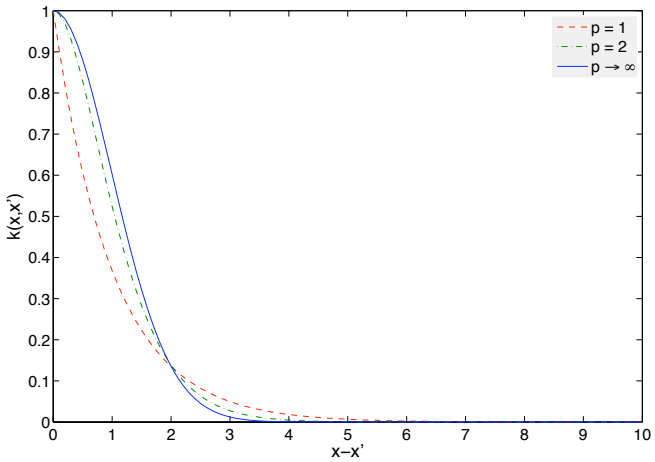
(f) Prior $\nu \to \infty$.

Figure: Three random functions generated from (a) the prior $\mathcal{GP}$ and (b) the posterior $\mathcal{GP}$ with the Matérn kernel ($l = 0.25$). The observations are indicated by $+$, the means by a dashed lines and the 3 standard deviation error bars by the shaded regions.

Matérn kernel *vs* rational quadratic kernel



(a) Rational quadratic.



(b) Matérn.

Figure: Comparison of the rational quadratic and the Matérn kernel with unit length scale ($l = 1$) for three values of respectively the shape and the roughness parameter.

- Both kernels are **less localised** than the squared exponential.
- Forcing the random latent functions to be **infinitely differentiable** might be **unrealistic** in practice.

Automatic relevance determination (ARD)

The principle denotes the idea that relevant input dimensions can be directly selected form the data.

ARD can be implemented as follows in the case of GPs with a squared exponential kernel:

$$k(\mathbf{x}, \mathbf{x}') = c^2 \exp\left\{ -\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{{l_d}^2} \right\},$$

where $\{l_d\}_{d=1}^{D}$ are allowed to be different.

The **characteristic length scale** l_d measures the distance for being uncorrelated along x_d. Hence, x_d is not relevant if $1/l_d$ is small.

Example

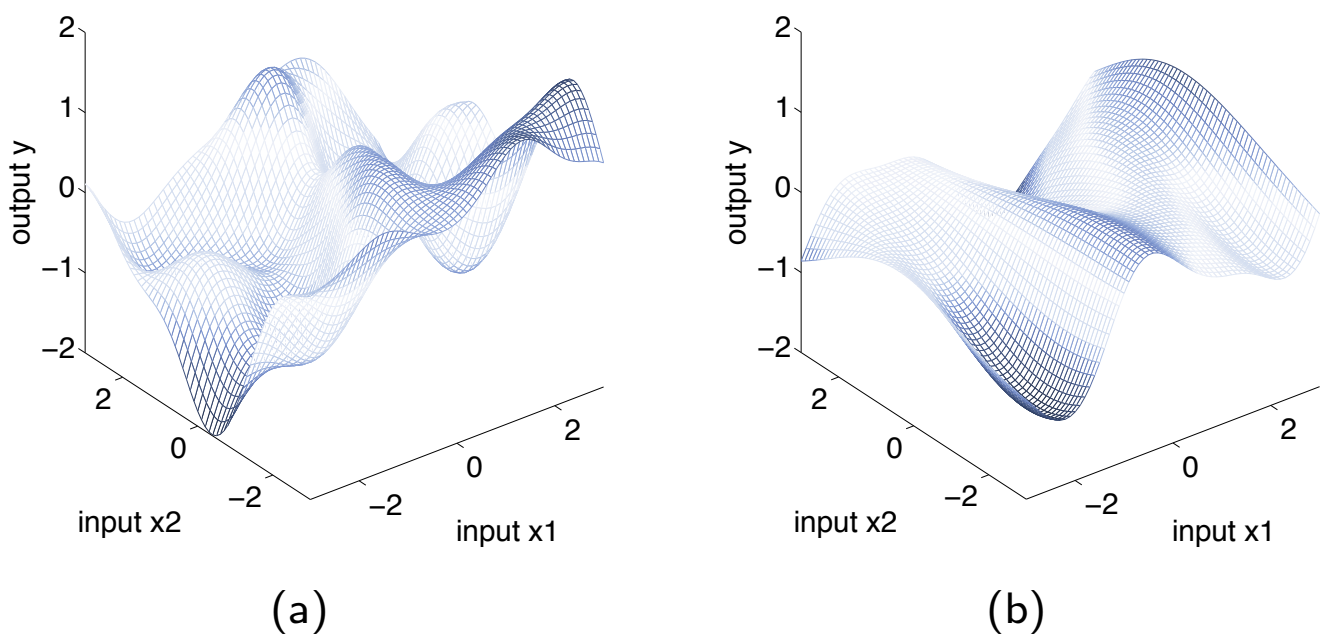


Figure: Latent function values $y(\mathbf{x})$ as a function of the input dimensions x_1 and x_2. In (a) both dimensions are relevant, while in (b) only x_1 is.