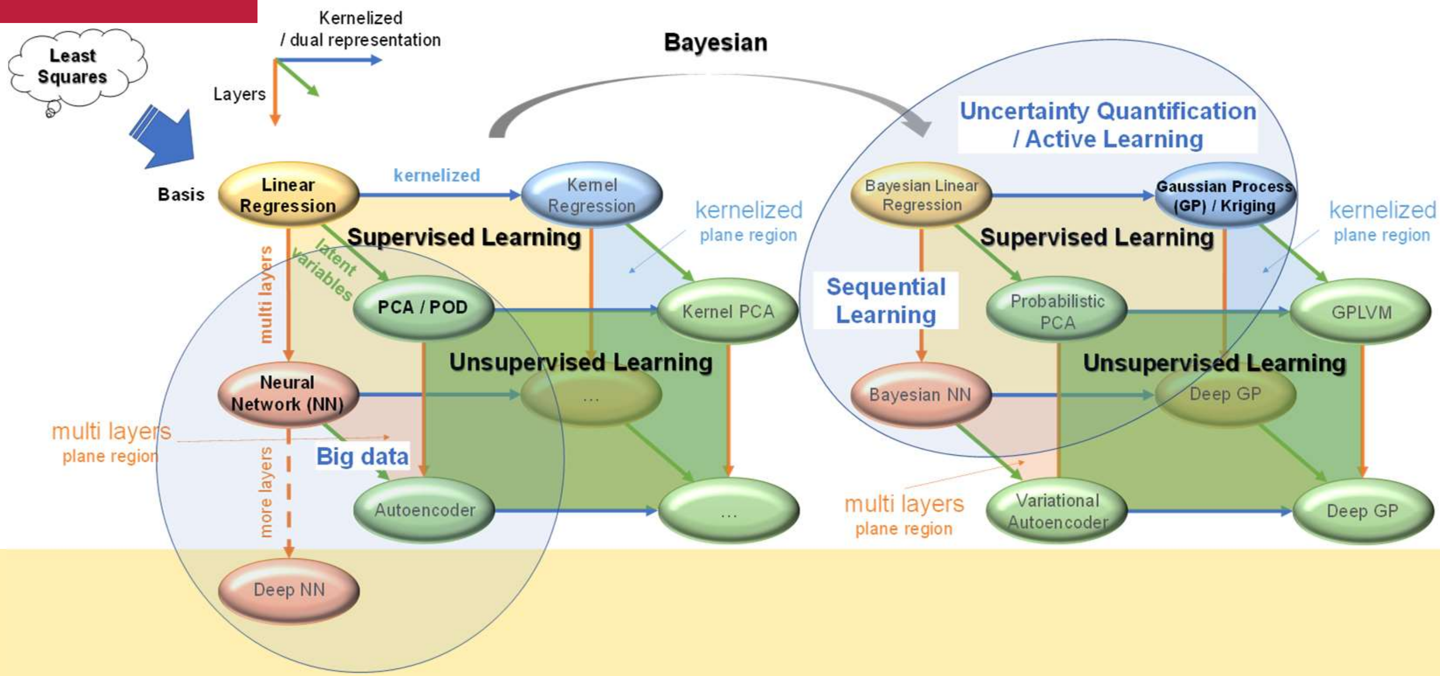


Scientific Machine Learning

Lecture 12: Numerical Methods, Bayesian Networks and Clustering

Dr. Daigo Maruyama

Prof. Dr. Ali Elham

Lecture content

- Bayesian networks
- Numerical methods for computing posterior distributions
- Clustering
  - highly related to the topics of Lecture 13

The lecture of this time partially follows the Chapter 8, Chapter 11, and Section 9.1 of the book:
Christopher M. Bishop "Pattern Recognition And Machine Learning" Springer-Verlag (2006)
The name of this book is shown as "PRML" when it is referred in the slides.

The lecture slides contains original topics in addition to the contents of the book.

Lecture content

- Bayesian networks

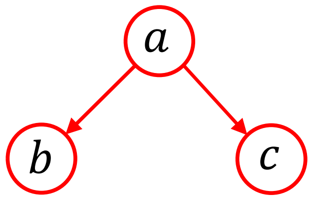
Graphical Models

The roles of the graphical models:

- Some properties in complicated probabilistic models can be visually clarified.
  (e.g. conditional independence)

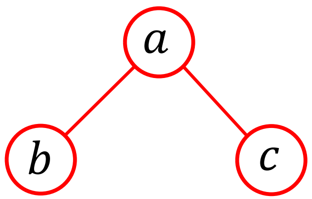
- Visualization of the above properties can assist to design new models.

able to describe causal relationships



directed graphical model

Bayesian network

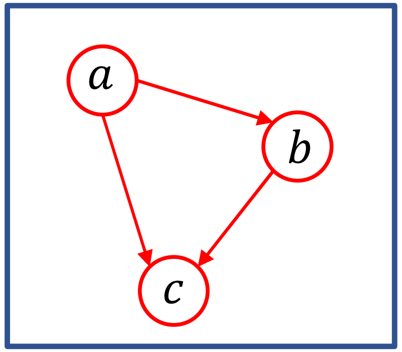


undirected graphical model

: stochastic variable

Bayesian Network

a, b, c: all stochastic variables



$p(a, b, c)$

The rules of probability

sum rule $\quad p(y) = \int p(x, y)\mathrm{d}x$

$$\boxed{\textbf{product rule} \quad p(x, y) = p(x|y)p(y)}$$

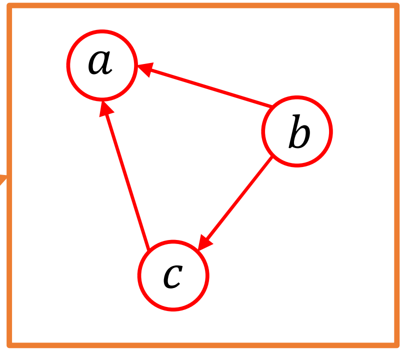
Let's consider the joint distribution:

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

$$p(a, b) = p(b|a)p(a)$$

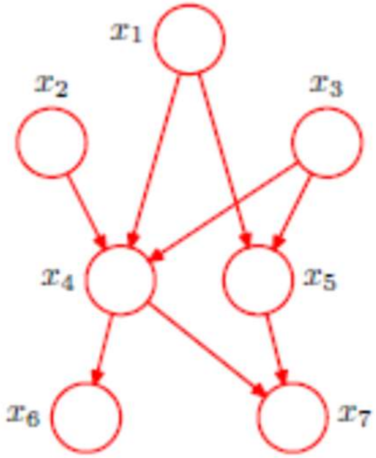
$$\underbrace{p(a, b, c)}_{\text{symmetric}} = \underbrace{p(c|a, b)p(b|a)p(a)}_{\text{not symmetric}}$$

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



$p(a, b, c)$

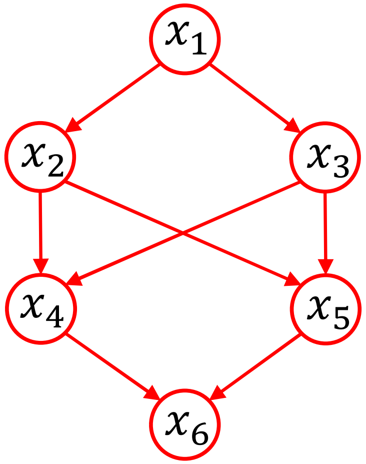
Bayesian Network: Some Examples



PRML, Fig. 8.2

product rule $p(x, y) = p(x|y)p(y)$

$$p(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$$
$$= p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)$$



practice

Describe the joint probability of the left Bayesian network.

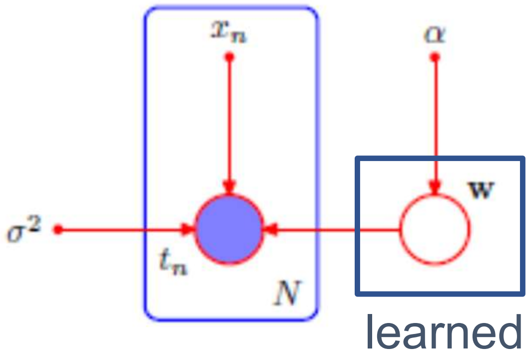
$$p(x_1, x_2, x_3, x_4, x_5, x_6) =?$$

Example: Bayesian Linear Regression

Let's see an example of graphical models using the Bayesian linear regression.

Probabilistic model

$$p(t|x,\boldsymbol{w}) = \mathcal{N}(t|y(x,\boldsymbol{w}),\sigma^2)$$



PRML, Fig. 8.6

$\boldsymbol{w}$: stochastic (prior is therefore introduced)
σ: deterministic

stochastic variables: open circles
deterministic variables: smaller solid circles

observed variables: shading
latent variables: no shading

$$p(\boldsymbol{w}) \longrightarrow p(\boldsymbol{w}|\mathbf{X},\mathbf{T})$$

Latent Variables

In a global sense, non-observed variables can be classified as latent variables

e.g. so-called parameters (e.g. $\boldsymbol{w}$) are also latent variables.

just detailed notes:
$\boldsymbol{w}$: intensive variables (fixed in number independent of the size of the data set)
$\boldsymbol{z}$: extensive variables (scale in number with of the size of the data set)

In the Bayesian perspective, all the variables are classified only as:

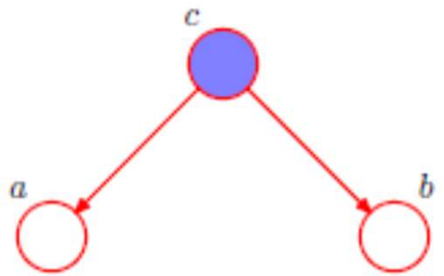
- **Observed**
- **Non-observed** (i.e. latent variables)

The probabilistic models for unsupervised learning become clear.
(shown in Lecture 13)

Three Important Properteis (Conditional Independence)

Common for the three cases: Describe $p(a, b, c)$, then compute $p(a, b|c) = \frac{p(a,b,c)}{p(c)}$

tail-to-tail

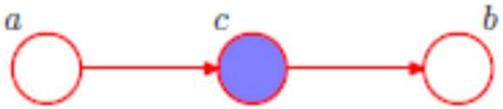


$$p(a, b, c) = p(a|c)p(b|c)p(c)$$

$$\Rightarrow \quad p(a, b|c) = p(a|c)p(b|c)$$

independent

head-to-tail

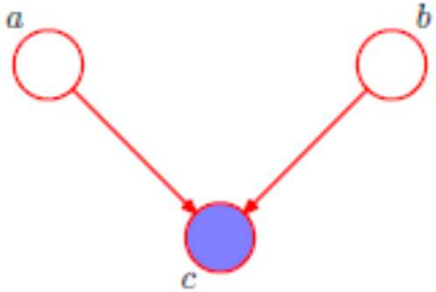


$$p(a, b, c) = p(a)p(c|a)p(b|c)$$

$$\Rightarrow \quad p(a, b|c) = p(a|c)p(b|c)$$

independent

head-to-head



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

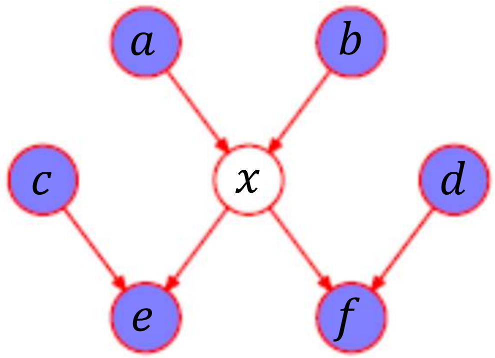
$$\Rightarrow \quad p(a, b|c) = \frac{p(a)p(b)p(c|a, b)}{p(c)}$$

Not independent

Important properties when c was observed!

Markov Blanket

to know these properties effectively by the graphical models



Markov blanket

When all the variables except for x were observed, the nodes that have correlation with x are as shown in the left figure:

- the parent a, b,
- the child e, f,
- the co-parent with x as e, f.

can be checked by the previous **three properties**!

All the other variables outside of the variables from a to f do not affect anything on the conditional distribution of x.

When the probabilistic model becomes complicated, these properties of the graphical models make the conditional independence clear by visual effects.

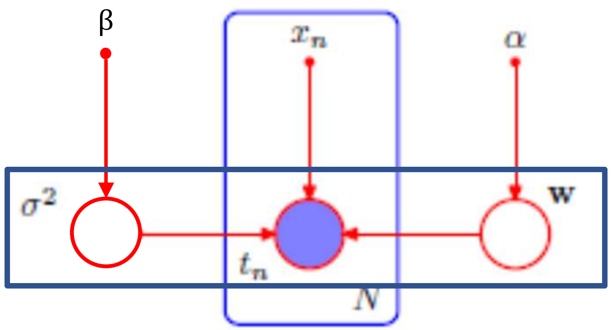
Example: Bayesian Linear Regression

Let’s see an example of graphical models using the Bayesian linear regression.

Probabilistic model

another case when σ is also stochastic.

$$\underline{p(t|x,\boldsymbol{w},\sigma)} = \mathcal{N}(t|y(x,\boldsymbol{w}),\sigma^2)$$



PRML, Fig. 8.6 with modification

$\boldsymbol{w}$: stochastic (prior is therefore introduced)

σ: stochastic (prior is therefore introduced)

head-to-head

prior

$$\begin{cases} p(\boldsymbol{w}) \\ p(\sigma) \end{cases} \Rightarrow$$

posterior: a joint distribution

$$\underline{p(\boldsymbol{w},\sigma|\mathbf{X},\mathbf{T})}$$

not like $p(\boldsymbol{w}|\mathbf{X},\mathbf{T})$, $p(\sigma|\mathbf{X},\mathbf{T})$

$\boldsymbol{w}, \sigma$ became correlated when $\mathbf{T}$ was observed.

Predictive distribution

$$p(t|x,\mathbf{X},\mathbf{T}) = \iint \underline{p(t|x,\boldsymbol{w},\sigma)}\,\underline{p(\boldsymbol{w},\sigma|\mathbf{X},\mathbf{T})}\mathrm{d}\boldsymbol{w}\mathrm{d}\sigma$$

Lecture content

- Numerical methods for computing posterior distributions

Posterior Distribution and Predictive Distribution

$\boldsymbol{w}$: stochastic
σ: stochastic

Let's think about **the curve fitting problem**.

Probabilistic model

$$p(t|x, \boldsymbol{w}, \sigma) = \mathcal{N}(t|\underline{y(x, \boldsymbol{w})}, \sigma^2)$$

e.g. **neural network**

Likelihood function

$$p(\mathbf{T}|\mathbf{X}, \boldsymbol{w}, \sigma) = \prod_{i=1}^{N} \mathcal{N}(t_i|y(x_i, \boldsymbol{w}), \sigma^2)$$

need optimizer

MLE: $\max_{\boldsymbol{w},\sigma} p(\mathbf{T}|\mathbf{X}, \boldsymbol{w}, \sigma)$

Posterior distribution $p(\boldsymbol{w}, \sigma|\mathbf{X}, \mathbf{T}) = \boldsymbol{complicated}$

Predictive distribution

$$p(t|x, \mathbf{X}, \mathbf{T}) = \iint p(t|x, \boldsymbol{w}, \sigma) p(\boldsymbol{w}, \sigma|\mathbf{X}, \mathbf{T}) \mathrm{d}\boldsymbol{w} \mathrm{d}\sigma = \boldsymbol{complicated}$$

probabilistic model $\times$ **posterior**

Posterior Distribution and Predictive Distribution

$\boldsymbol{w}$: stochastic
σ: **deterministic**

Let's think about **the curve fitting problem**.

Probabilistic model

$$p(t|x,\boldsymbol{w}) = \mathcal{N}(t|\underline{y(x,\boldsymbol{w})},\sigma^2)$$

e.g. **neural network**

Likelihood function

$$p(\mathbf{T}|\mathbf{X},\boldsymbol{w}) = \prod_{i=1}^{N} \mathcal{N}(t_i|y(x_i,\boldsymbol{w}),\sigma^2)$$

need optimizer

MLE: $\max_{\boldsymbol{w},\sigma} p(\mathbf{T}|\mathbf{X},\boldsymbol{w})$

Posterior distribution $p(\boldsymbol{w}|\mathbf{X},\mathbf{T}) = \boldsymbol{complicated}$

Predictive distribution

$$p(t|x,\mathbf{X},\mathbf{T}) = \int p(t|x,\boldsymbol{w})p(\boldsymbol{w}|\mathbf{X},\mathbf{T})\mathrm{d}\boldsymbol{w} = \boldsymbol{complicated}$$

probabilistic model × **posterior**

Posterior Distribution and Predictive Distribution

$\boldsymbol{w}$: stochastic
σ: **deterministic**

Let's think about **the curve fitting problem**.

Probabilistic model

$$p(t|x,\boldsymbol{w}) = \mathcal{N}(t|\underline{y(x,\boldsymbol{w})},\sigma^2)$$

e.g. **linear regression** as $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$

Likelihood function

$$p(\mathbf{T}|\mathbf{X},\boldsymbol{w}) = \prod_{i=1}^{N}\mathcal{N}(t_i|y(x_i,\boldsymbol{w}),\sigma^2)$$

no need of optimizer

MLE: $\max_{\boldsymbol{w},\sigma} p(\mathbf{T}|\mathbf{X},\boldsymbol{w})$

⇒ Posterior distribution $p(\boldsymbol{w}|\mathbf{X},\mathbf{T}) = \boldsymbol{Guassian}$

Predictive distribution

$$p(t|x,\mathbf{X},\mathbf{T}) = \underbrace{\int p(t|x,\boldsymbol{w})p(\boldsymbol{w}|\mathbf{X},\mathbf{T})\mathrm{d}\boldsymbol{w}}_{\textbf{probabilistic model} \times \textbf{posterior}} = \boldsymbol{Gaussian}$$

⇒ **Gaussian process** by dual representation

Numerical Approximation of Posterior Distributions

Posterior distribution

$$\underline{p(\boldsymbol{w}|\mathcal{D})} = \boldsymbol{complicated}$$

Predictive distribution

$$p(t|x,\mathcal{D}) = \int p(t|x,\boldsymbol{w})\underline{p(\boldsymbol{w}|\mathcal{D})}\mathrm{d}\boldsymbol{w} = \boldsymbol{complicated}$$

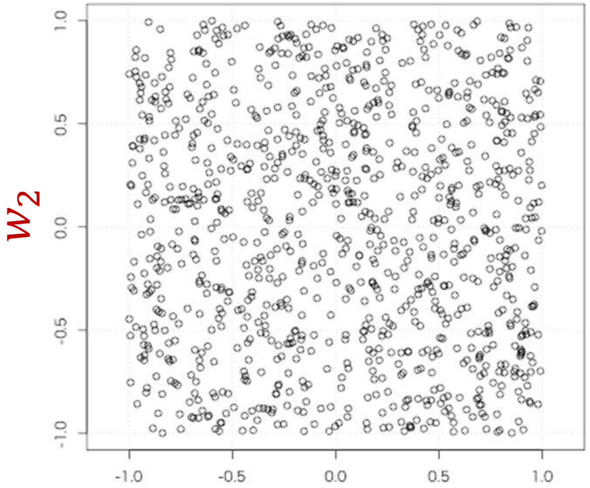
Numerical approximation of the posterior:

- Markov-Chain Monte Carlo — approximation by <u>sampling</u>
- Variational inference
- Laplace approximation

(Variational inference and Laplace approximation:) approximation by <u>parametric pdf</u>
e.g. a Gaussian distribution

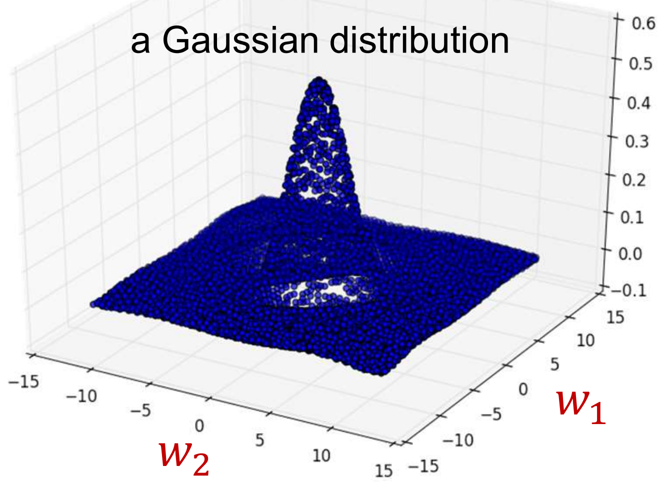
Markov-Chain Monte Carlo (MCMC)

Monte Carlo sampling (random sampling)

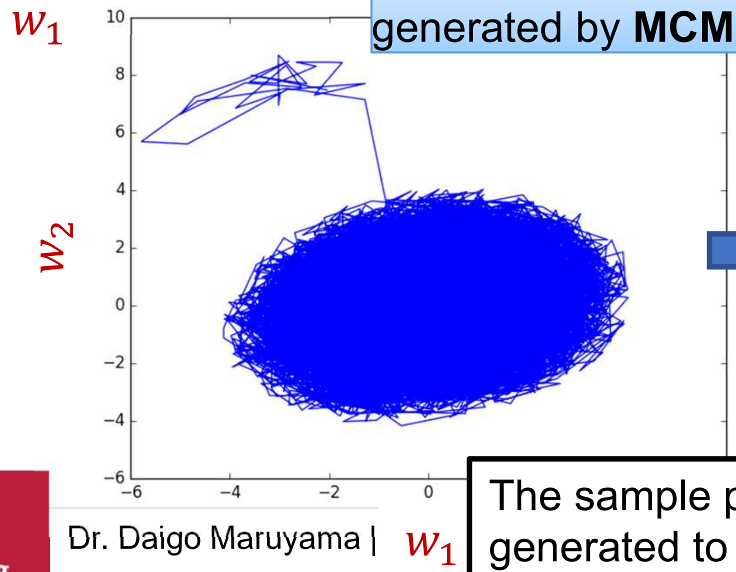


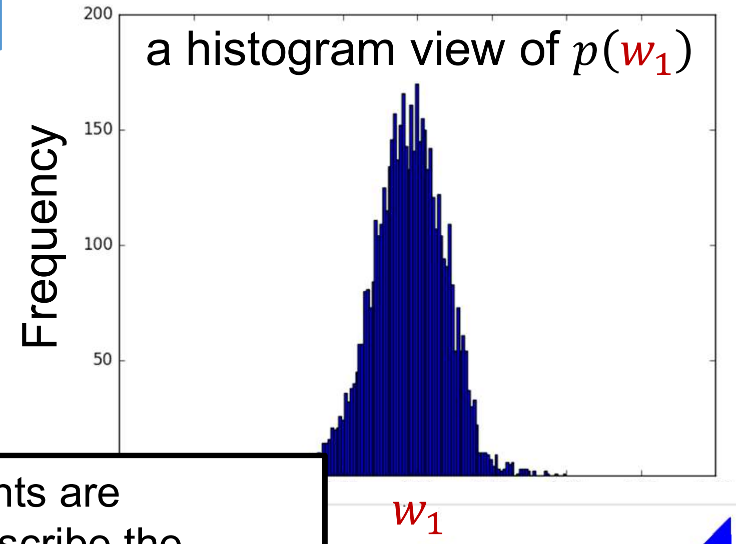
$p(\boldsymbol{w}) = 1$
(**uniform distribution**)

visualization of the **target pdf** $p(\boldsymbol{w})$



Points (and the trace) generated by **MCMC**





The sample points are generated to describe the function $p(\boldsymbol{w})$ by the frequency.

Markov-Chain Monte Carlo (MCMC)

Random walk

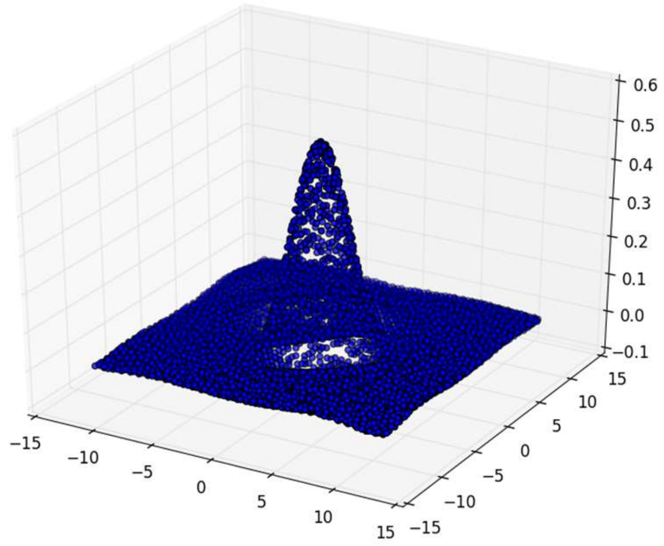
$$candidate = current + \mathcal{N}(0, \sigma)$$

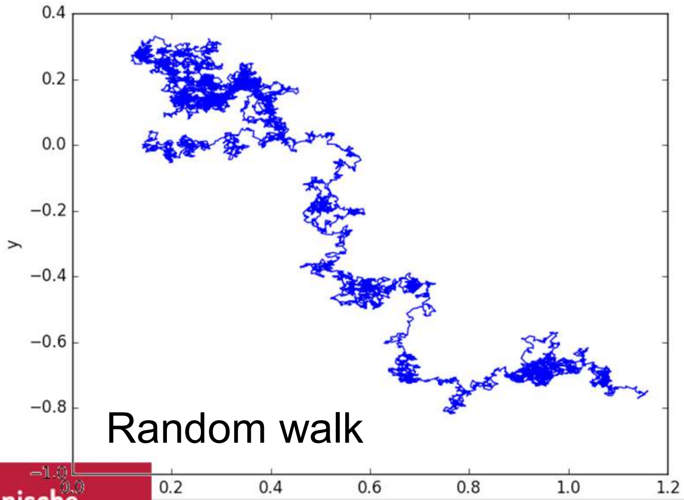
Metropolis Hastings

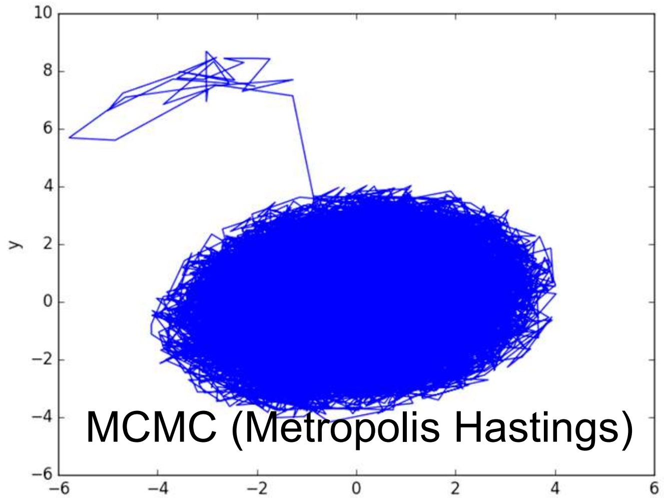
$$a = \frac{p(candidate)}{p(current)}$$

If $a > 1$, or $a > r$: a random number $r \in (0,1)$:

$$candidate \rightarrow current$$



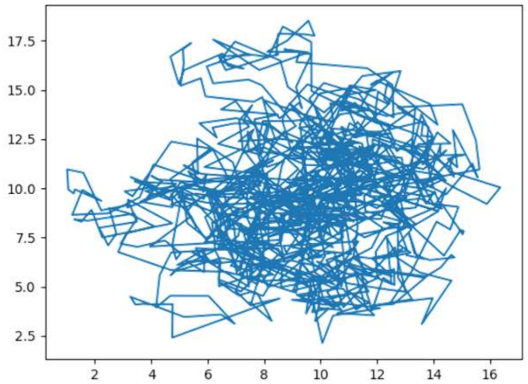




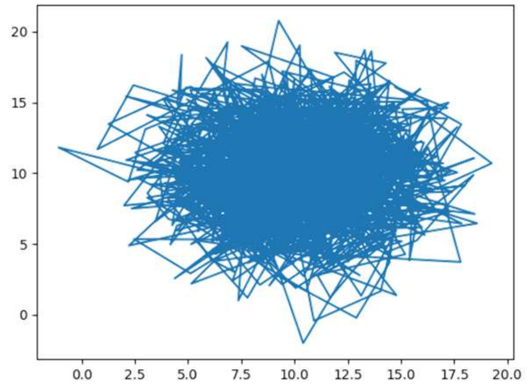
Some Variations of Algorithms in MCMC

There are many algorithms in MCMC (like many algorithms in optimizer).

<u>12 dimensional</u> Gaussian distributions (analytical function test case) as the target posterior (extracted 2 input parameters to visualize)



by Metropolis-Hastings

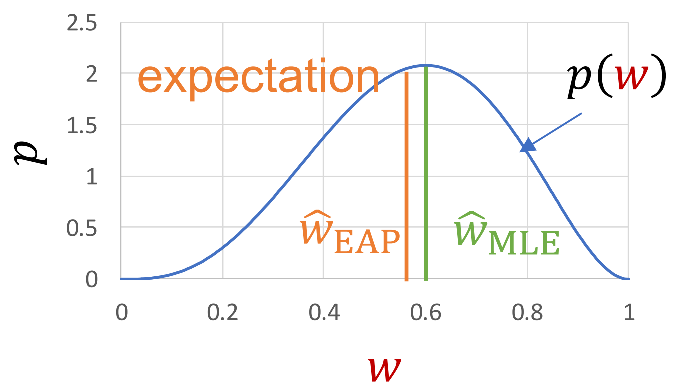


by Hamiltonian MC

e.g. Hamiltonian Monte Carlo
- More effective in high-dimensional spaces
- Requires gradient information of the target function wrt the parameters

Applications of MCMC

Example: There is a pdf $p(w)$ (prior $p(w)$ or posterior $p(w|\mathcal{D})$).



$\hat{w}_{\text{MLE}}$

The **point estimate** approaches
(w is <u>deterministic</u>)

$\hat{w}_{\text{MLE}} = \max_{w} p(w)$ by a optimizer

$\hat{w}_{\text{EAP}} = E[w]$ how?

Note: if no weights ⇨ conventional DoE

$$E[w] = \int w \mathrm{d}w \approx \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} w$$

Approximation by using e.g. Monte Carlo

$$E[w] = \int w \times p(w) dw \approx \frac{1}{N_{mcmc}} \sum_{i=1}^{N_{mcmc}} w$$

Sampling approximation by using MCMC

Applications of MCMC

Predictive distribution (the goal)

The **probability distribution** approaches
(w is <u>stochastic</u>)

only in special cases! (see Lectures 6,7)

$$p(t|x,\mathcal{D}) = \int p(t|x,\boldsymbol{w})p(\boldsymbol{w}|\mathcal{D})\mathrm{d}\boldsymbol{w} \qquad = \mathcal{N}\left(t \middle| \boldsymbol{m}_N{}^{\mathrm{T}}\phi(\boldsymbol{x}), \sigma_N{}^2(\boldsymbol{x})\right)$$

$$\approx \frac{1}{N_{mcmc}} \sum_{i=1}^{N_{mcmc}} p(t|x,\boldsymbol{w})$$

If you have the result of MCMC on the posterior $p(\boldsymbol{w}|\mathcal{D})$

weighted sum of the probability $p(t|x,\boldsymbol{w})$

- the posterior distribution
- the predictive distribution

represented by the sample points

Other applications of MCMC

A functino distribution can be revealed
(by weighted samples).

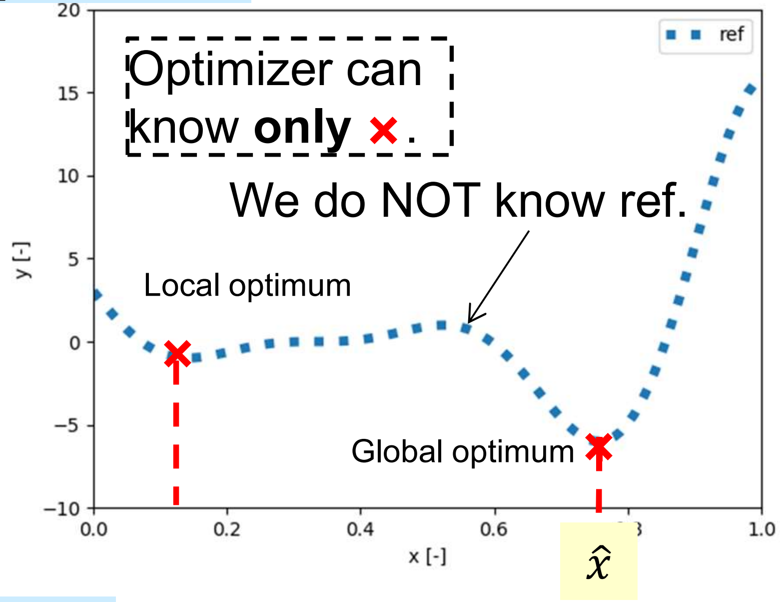
- Can be used to find **global optimum**
- Can be used for **robust design**

But expensive (since it is sampling method)

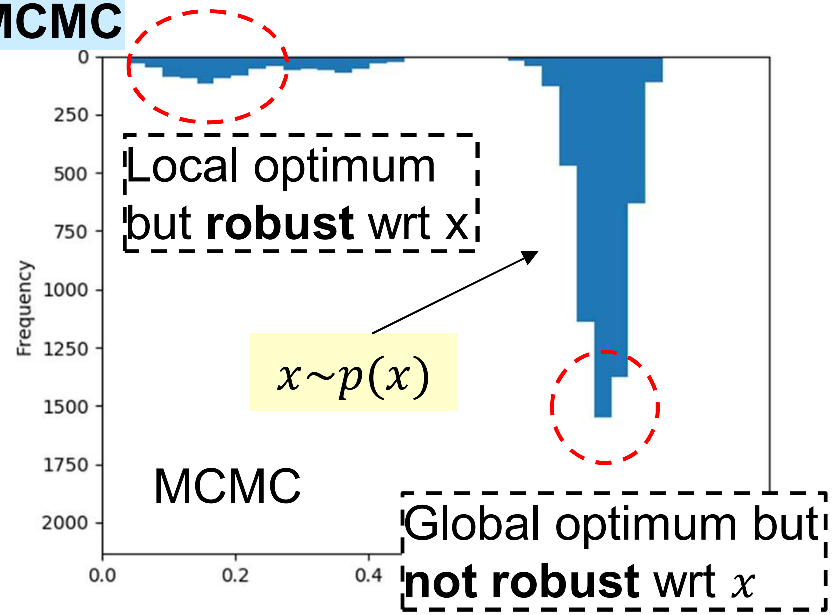
It is common with optimization that we have to care the parameters input to the algorithms.

Input parameters in MCMC:
- step length
- burn-in
- etc.

Optimization



MCMC

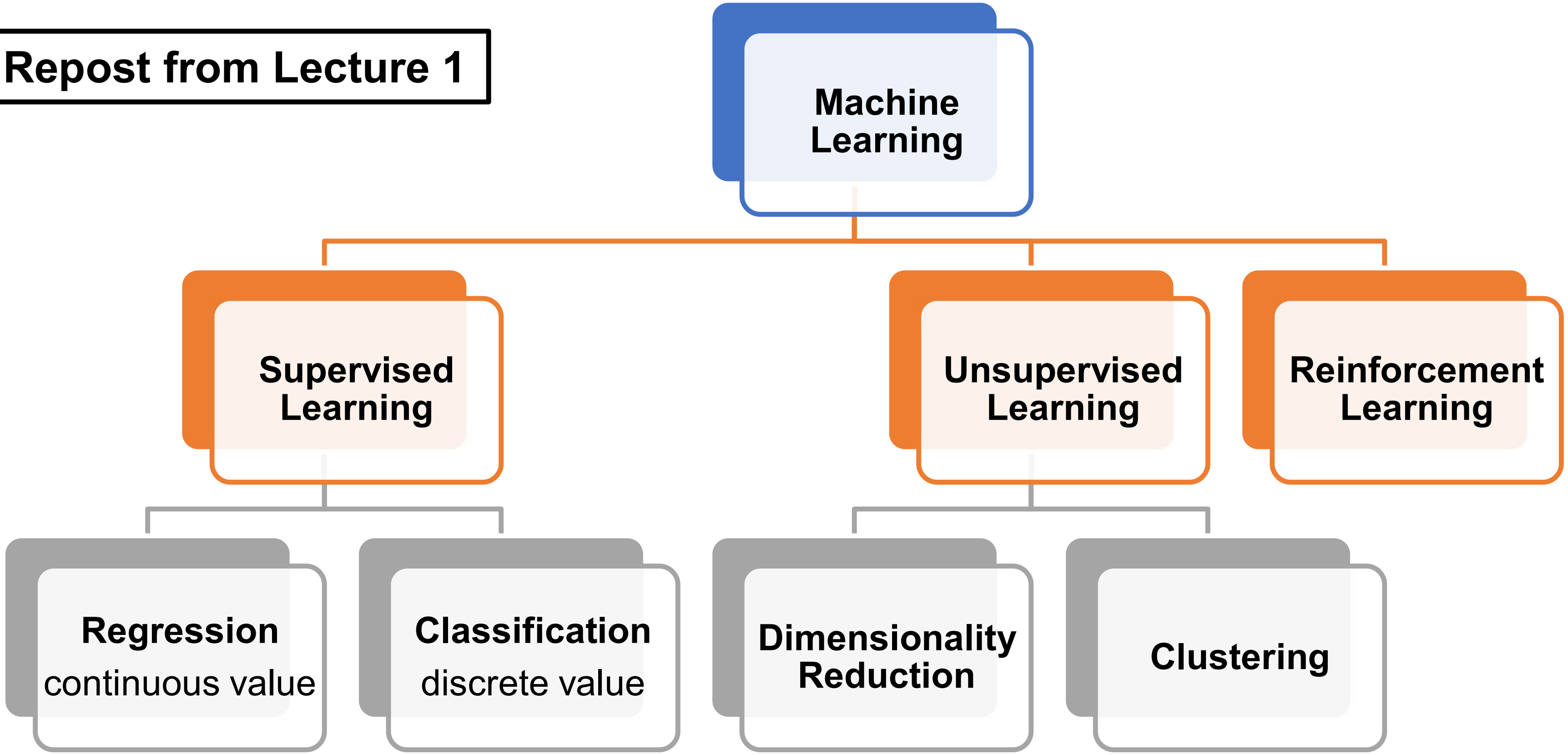


Lecture content

- Clustering
  - highly related to the topics of Lecture 13

Machine Learning Classification by Use/Application

Repost from Lecture 1



In this course, machine learning classification is done by **methods and their concepts**.

→ Then the use/application is naturally derived/understood.

Clustering (*K*-means algorithm)

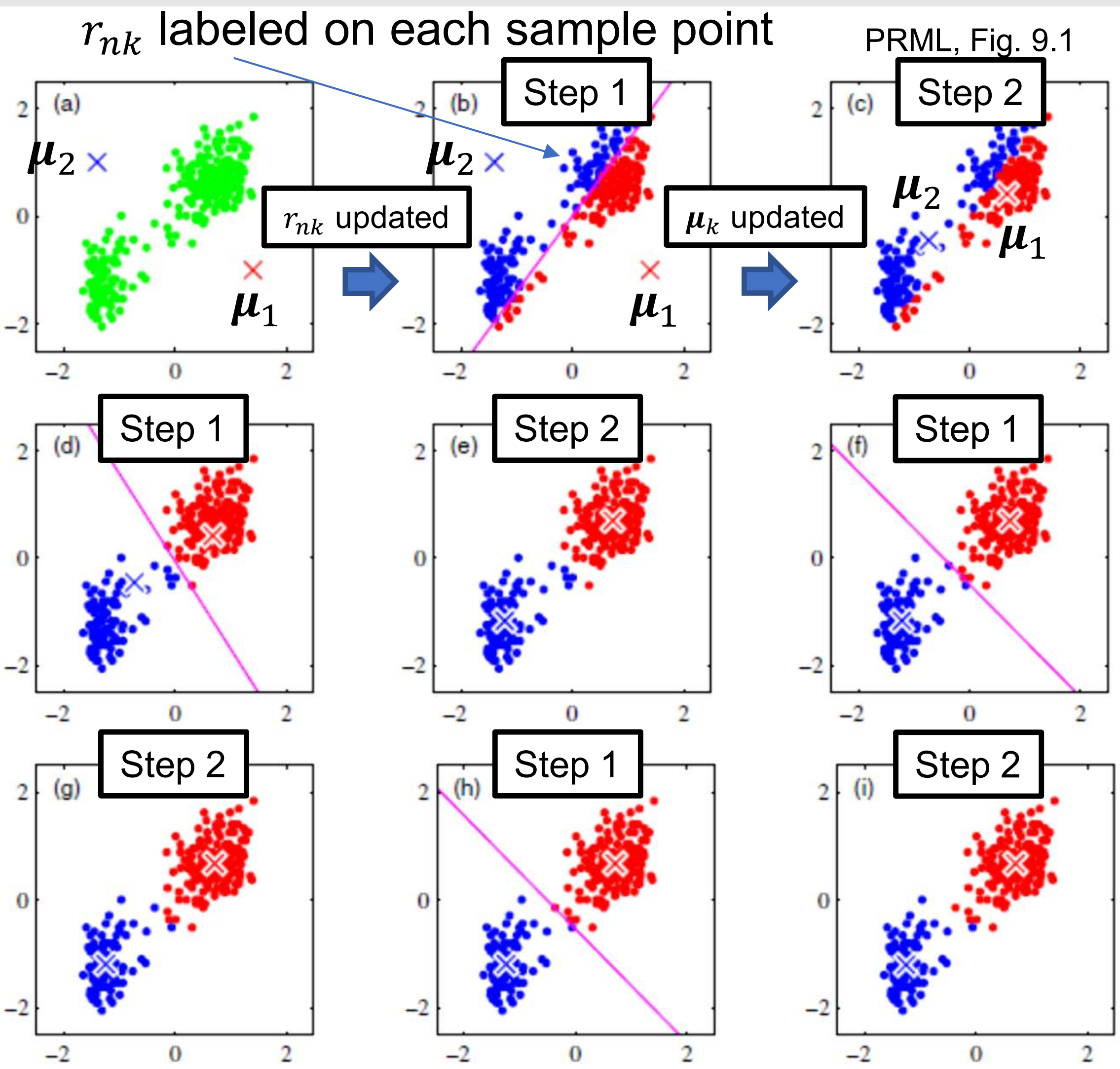
$$E(r_{nk}, \boldsymbol{\mu}_k) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k \right\|^2$$

One iteration is composed of two steps:

1. $\min_{r_{nk}} E\ (r_{nk}, \boldsymbol{\mu}_k)$
2. $\min_{\boldsymbol{\mu}_k} E\ (r_{nk}, \boldsymbol{\mu}_k)$

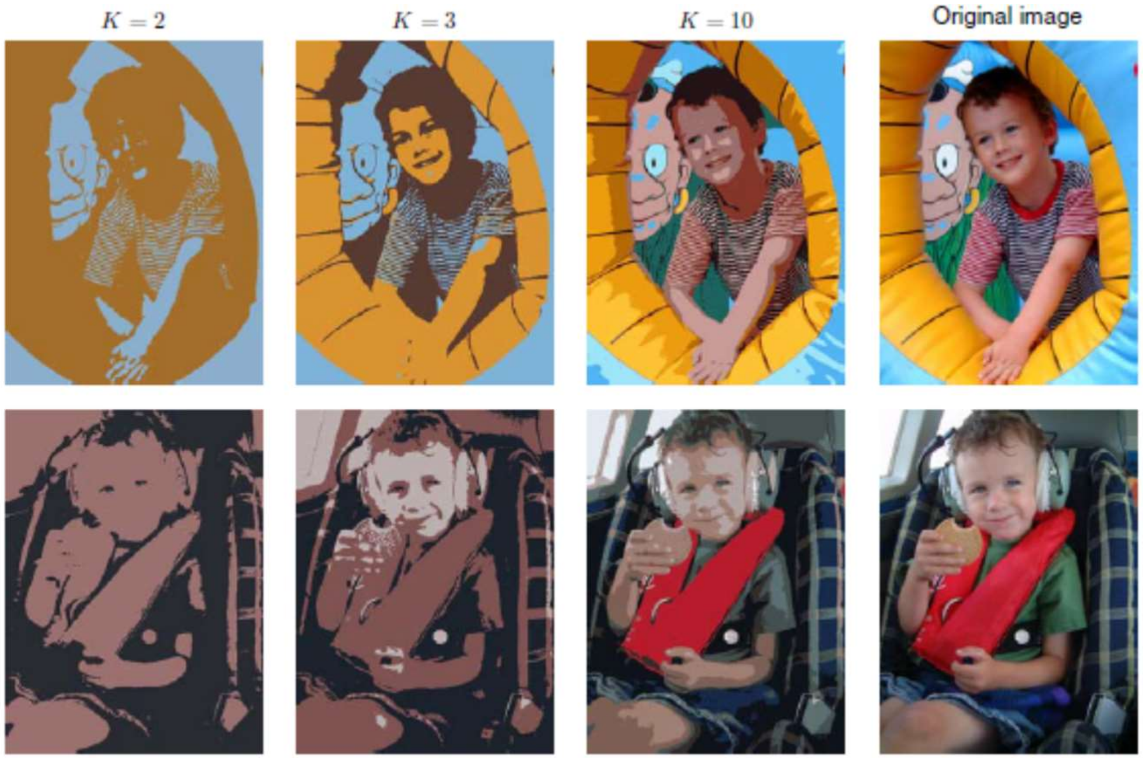
$\boldsymbol{\mu}_k$: the mean of $\boldsymbol{x}^{(n)}$ in the current cluster k

r_{nk}: when the nearest $\boldsymbol{\mu}_k$, 1. Otherwise 0.



Clustering (*K*-means algorithm) – Other Examples

compression data files



PRML, Fig. 9.3

Similar colors are summarized as one color, which corresponds to each cluster.

Clustering (*K*-means algorithm) as a Probabilistic Model

The message here is that:

Even this algorithm can be regarded as a special case, it can be a modeling using the probability theory.

Not from the classification from Use/ Application

➡ Lecture 13

Mixtures of Gaussians

Summary

- The graphical models were learned to assist to model complicated probabilistic models
  by **using know properties easily judged by the graph as visualization information**

- **Approximation methods** to compute **the posterior / predictive distributions** in the Bayesian approaches
  - The point estimate is fine since we can use optimizer but clarifying distributions needs more information
  - **Markov-Chain Monte Carlo (MCMC)** is expensive but can represent the distributions by sampling. Other methods are the variational inference, Laplace approximation, etc.

- Clustering from application viewpoint was introduced
  extended to mixture of Gaussians

⇨ more generalized perspective of probabilistic modeling in Lecture 13