Machine Learning

Parametric Nonparametric Techniques- Lecture III

Course Outline

Basic Concepts

- Parametric Method,
- Bayesian Learning and Nonparametric Methods

Classical Approaches

- Clustering and Mixture of Gaussians
- Linear Discriminants

Ensemble Methods

- Ensemble Methods and Boosting
- Randomized Trees, Forest

Reinforcement Learning

- Classical Reinforcement Learning

Neural Networks and Deep Learning

- Foundations
- Optimization

Today’s topics

Recap: Bayes decision theory

Parametric Methods

- Recap: Maximum Likelihood approach
- Bayesian Learning

Non-parametric Methods

- Histograms
- Kernel density estimation
- K-Nearest Neighbours
- k-NN for Classification

Part 1, Video ParamNonparam_p1

- Decision rule for multiple classes
- Clyssifying with loss function
- Maximum Likelihood

Recap: Bayes Decision Theory

Optimal decision rule

Decide for C_1 if

$$p(\mathcal{C}_1|x) > p(\mathcal{C}_2|x)$$

This is equivalent to

$$p(x|\mathcal{C}_1)p(\mathcal{C}_1) > p(x|\mathcal{C}_2)p(\mathcal{C}_2)$$

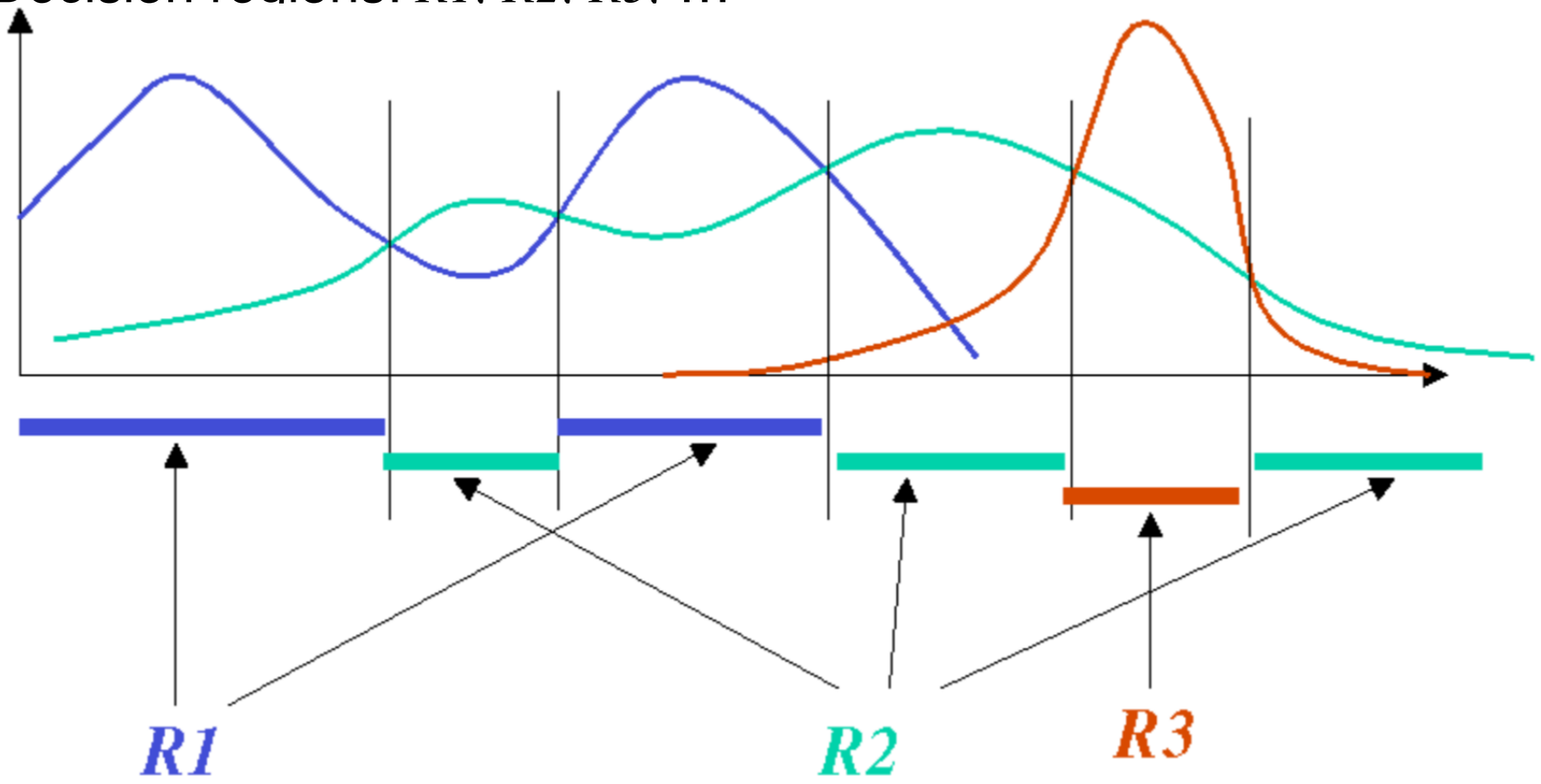
Which is again equivalent to (**Likelihood-Ratio test**)

$$\frac{p(x|\mathcal{C}_1)}{p(x|\mathcal{C}_2)} > \underbrace{\frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}}_{\text{Decision threshold } \theta}$$

Slide source: B. Leibe Slides ML, 2015

Recap: Bayes Decision Theory

Decision regions: $R1, R2, R3, \ldots$



Slide source: B. Leibe Slides ML, 2015

Recap: Classifying with Loss Function

In general, we can formalize this by introducing a loss matrix L_{kj}

$$L_{kj} = loss\ for\ decision\ \mathcal{C}_j\ \ if\ \ truth\ is\ \mathcal{C}_k$$

Example cancer diagnosis:

$$L_{cancer\ \ diagnosis} = \begin{array}{c} \\ \text{Truth} \end{array} \begin{array}{c} \begin{array}{cc} \text{cancer} & \text{normal} \end{array} \\ \begin{array}{l} \text{cancer} \\ \text{normal} \end{array} \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}$$

Decision

Slide source: B. Leibe Slides ML, 2015

Recap: Classifying with Loss Function

- Optimal solution is the one that minimizes the loss.
  - But: loss function depends on the true class, which is unknown.
- Solution: **Minimize the expected loss**

$$\mathbb{E}[L] = \sum_k \sum_j \int_{R_j} L_{kj} p(x, C_k) \mathrm{dx}$$

- This can be done by choosing the regions R_j such that

$$\mathbb{E}_j[L] = \sum_k L_{kj} p(C_k|x)$$

=> Adapted decision rule:

$$\frac{p(\mathbf{x}|\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)} > \frac{(L_{21} - L_{22})}{(L_{12} - L_{11})} \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}$$

Slide source: B. Leibe Slides ML, 2015

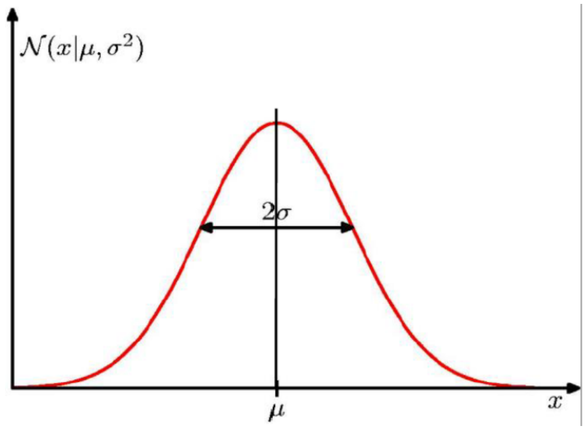
Recap: Gaussian (or Normal) Distribution

One-dimensional case

Mean μ

Variance σ^2

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

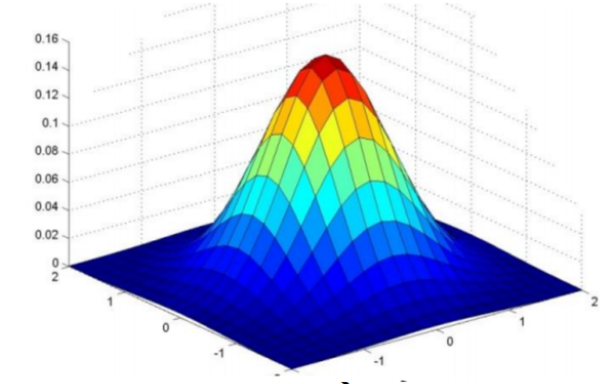


Multi-dimensional case

Mean μ

Covariance Σ

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



Slide source: B. Leibe Slides ML, 2015

Recap: Maximum Likelihood Approach

- Computation of the likelihood
  - Single data point: $p(x_n|\theta)$
  - Assumption: all data points $\mathrm{X} = \{x_1, ..., x_n\}$ are independent

$$L(\theta) = p(\mathrm{X}|\theta) = \prod_{n=1}^{N} p(x_n|\theta)$$

  - Log-likelihood

$$E(\theta) = -\ln L(\theta) = -\sum_{n=1}^{N} \ln p(x_n|\theta)$$

- Estimation of the parameters μ (Learning)
  - Maximize the likelihood (=minimize the negative log-likelihood)

=> Take the derivative and set it to zero.

$$\frac{\partial}{\partial\theta} E(\theta) = -\sum_{n=1}^{N} \frac{\frac{\partial}{\partial\theta} p(x_n|\theta)}{p(x_n|\theta)} \stackrel{!}{=} 0$$

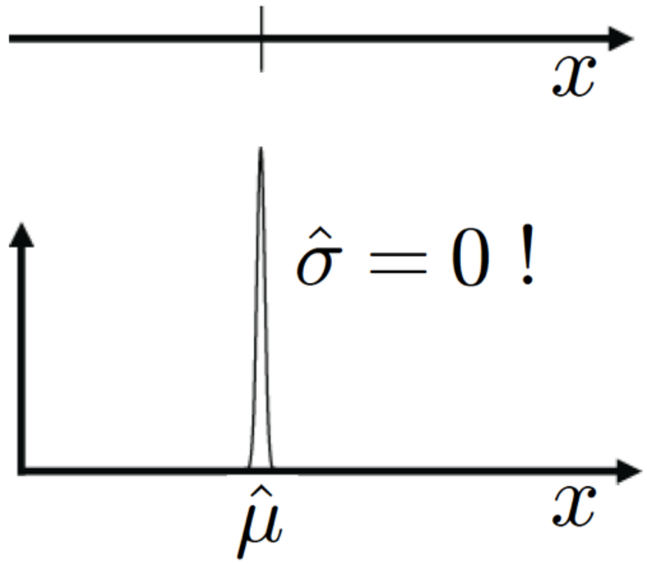
Slide source: B. Leibe Slides ML, 2015

Recap: Maximum Likelihood Limitations

- Maximum Likelihood has several significant limitations
  - It systematically underestimates the variance of the distribution!
  - E.g. consider the case

$$N = 1, \mathrm{X} = \{x_1\}$$

=> Maximum-likelihood estimate



  - We say ML *overfits to the observed data.*
  - We will still often use ML, but it is important to know about this effect.

Slide source: B. Leibe Slides ML, 2015

Part 2, Bayesian vs. Frequentist

- Problem with Maximum likelihood and deeper reason behind it
- Bayesian learning approach
- Maximum likelihood vs. Bayesian learning

Today’s topics

Recap: Bayes decision theory

Parametric Methods

- Recap: Maximum Likelihood approach
- **Bayesian Learning**

Non-parametric Methods

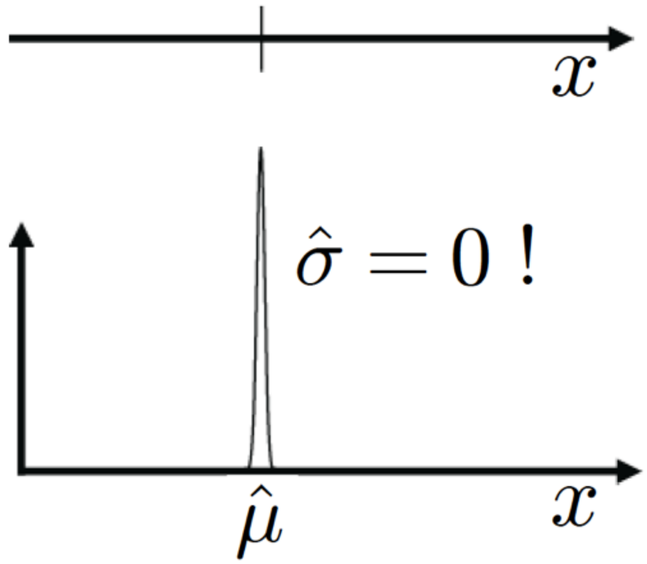
- Histograms
- Kernel density estimation
- K-Nearest Neighbours
- k-NN for Classification

Recap: Maximum Likelihood Limitations

- Maximum Likelihood has several significant limitations
  - It systematically underestimates the variance of the distribution!
  - E.g. consider the case

$$N = 1, \mathrm{X} = \{x_1\}$$

=> Maximum-likelihood estimate



  - We say ML *overfits to the observed data.*
  - We will still often use ML, but it is important to know about this effect.

Slide source: B. Leibe Slides ML, 2015

Deeper Reason

Maximum Likelihood is a **Frequentist** concept

- In the **Frequentist view**, probabilities are the frequencies of random, repeatable events.
- These frequencies are fixed, but can be estimated more precisely when more data is available.

This is in contrast to the **Bayesian** interpretation

- In the **Bayesian view**, probabilities quantify the uncertainty about certain states or events.
- This uncertainty can be revised in the light of new evidence.

Bayesians and Frequentists do not like each other too well…

Slide source: B. Leibe Slides ML, 2015

Bayesian vs. Frequentist View

To see the difference…

- Suppose we want to estimate the uncertainty whether the Arctic ice cap will have disappeared by the end of the century.
- This question makes no sense in a Frequentist view, since the event cannot be repeated numerous times.
- In the Bayesian view, we generally have a prior, e.g. from calculations how fast the polar ice is melting.
- If we now get fresh evidence, e.g. from a new satellite, we may revise our opinion and update the uncertainty from the prior

$$Posterior \propto Likelihood \times Prior$$

- This generally allows to get better uncertainty estimates for many situations.

Main Frequentist criticism

- The prior has to come from somewhere and if it is wrong, the result will be worse.

Slide source: B. Leibe Slides ML, 2015

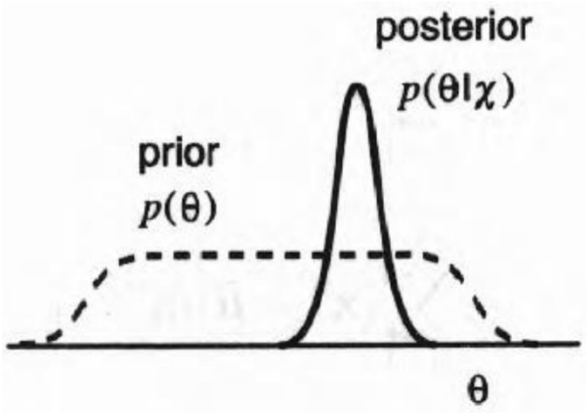
Bayesian Approach to Parameter Learning

Conceptual shift

- Maximum Likelihood views the true parameter vector ɵ to be **unknown, but fixed**.
- In Bayesian learning, we consider ɵ to be a **random variable**.

This allows us to use knowledge about the parameters ɵ

- i.e., to use a prior for ɵ
- Training data than converts this prior distribution on ɵ into a posterior probability density.
- The prior thus encodes knowledge we have about the type of distribution we expect to see for ɵ.



Slide source: B. Leibe Slides ML, 2015

Bayesian Learning Approach

Bayesian view:

- Consider the parameter vector ɵ as a random variable.
- When estimating the parameters from a dataset X, we compute

$$p(x|X) = \int p(x, \theta|X) d\theta$$

Assumption: given ɵ, this doesn't depend on X anymore

$$p(x, \theta|X) = p(x|\theta, \cancel{X}) p(\theta|X)$$

$$p(x|X) = \int p(x|\theta) p(\theta|X) d\theta$$

This is entirely determined by the parameter ɵ (i.e., by the parametric form of the pdf).

Slide source: B. Leibe Slides ML, 2015

Bayesian Learning Approach

$$p(x|X) = \int p(x|\theta)p(\theta|X)d\theta$$

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)} = \frac{p(\theta)}{p(X)}L(\theta)$$

$$p(X) = \int p(X|\theta)p(\theta)d\theta = \int L(\theta)p(\theta)d\theta$$

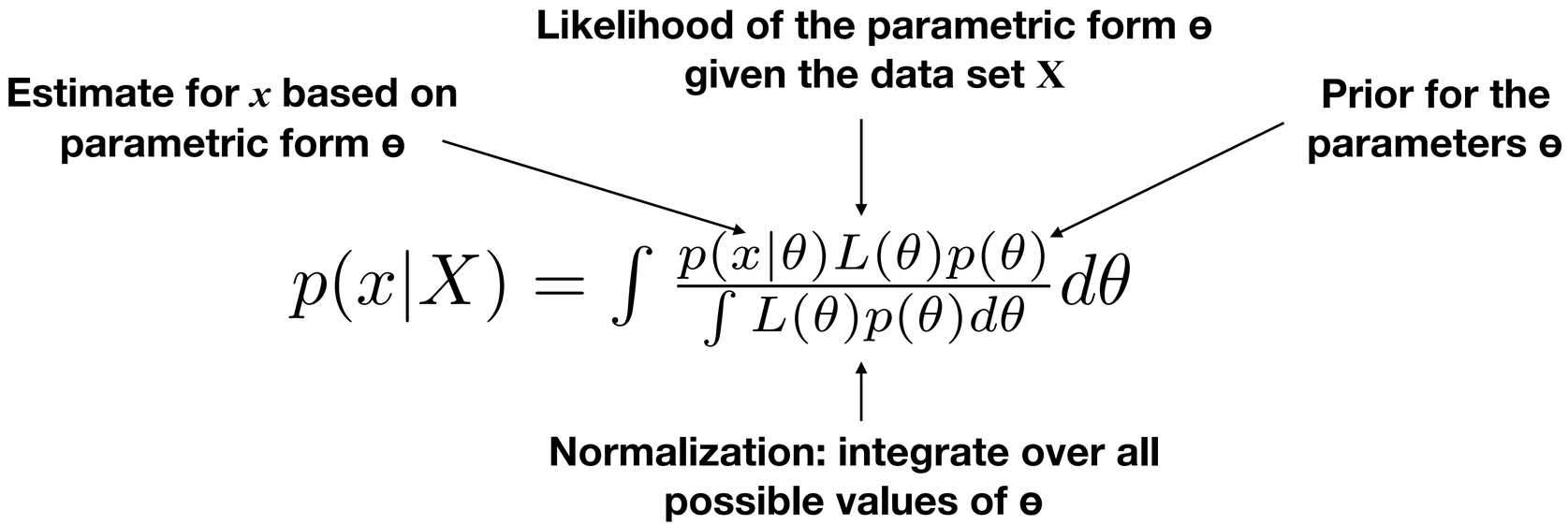
Inserting this above, we obtain

$$p(x|X) = \int \frac{p(x|\theta)L(\theta)p(\theta)}{p(X)}d\theta = \int \frac{p(x|\theta)L(\theta)p(\theta)}{\int L(\theta)p(\theta)d\theta}d\theta$$

Slide source: B. Leibe Slides ML, 2015

Bayesian Learning Approach

Discussion



If we now plug in a (suitable) prior $p(\theta)$, we can estimate $p(x|X)$ from the data set X.

Bayesian Density Estimation

Discussion

$$p(x|X) = \int p(x|\theta)p(\theta|X)d\theta = \int \frac{p(x|\theta)L(\theta)p(\theta)}{\int L(\theta)p(\theta)d\theta}d\theta$$

The probability $p(\theta|X)$ makes the dependency of the estimate on the data explicit.

If $p(\theta|X)$ is very small everywhere, but is large for one $\hat{\theta}$, then

$$p(x|X) \approx p(x|\hat{\theta})$$

$\Rightarrow$ In this case, the estimate is determined entirely by $\hat{\theta}$

$\Rightarrow$ The more uncertain we are about θ, the more we average over all parameter values.

Slide source: B. Leibe Slides ML, 2015

Bayesian Density Estimation

- Problem
  - In the general case, the integration over θ is not possible (or only possible stochastically).

- Example where an analytical solution is possible
  - Normal distribution for the data, σ^2 assumed known and fixed.
  - Estimate the distribution of the mean:

$$p(\mu|X) = \frac{p(X|\mu)p(\mu)}{p(X)}$$

  - Prior: We assume a Gaussian prior over μ

$$p(\mu) = \mathcal{N}\left(\mu|\mu_0, \sigma_0^2\right)$$

Slide source: B. Leibe Slides ML, 2015

Bayesian Learning Approach

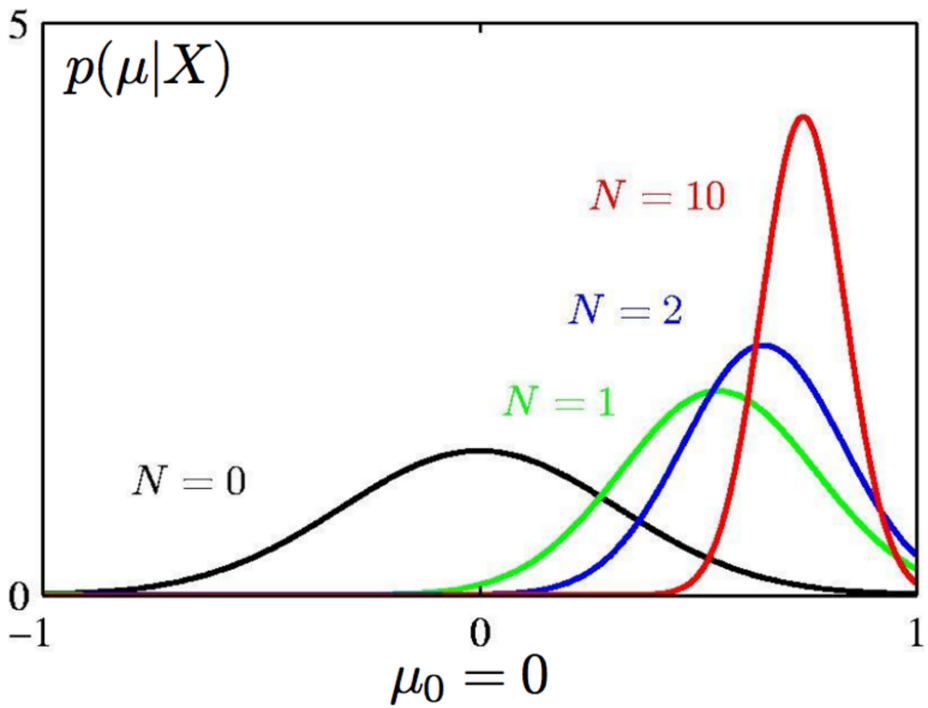
- Sample mean: $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$
- Bayes estimate:

$$\mu_N = \frac{\sigma^2 \mu_0 + N\sigma_0^2 \bar{x}}{\sigma^2 + N\sigma_0^2}$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2}$$

- Note:

	$N = 0$	$N \to \infty$
μ_N	μ_0	μ_{ML}
σ_N^2	σ_0^2	0



Slide source: B. Leibe Slides ML, 2015

Summary: ML vs. Bayesian Learning

- **Maximum Likelihood**
  - Simple approach, often analytically possible
  - Problem: estimation is biased, tends to overfit to the data
    - Often needs some correction or regularization
  - But:
    - Approximation gets accurate for $N \rightarrow \infty$
- **Bayesian Learning**
  - General approach, avoids the estimation bias through a prior
  - Problems:
    - Need to choose a suitable prior (not always obvious)
    - Integral over θ often not analytically feasible anymore
  - But:
    - Efficient stochastic sampling techniques available

Slide source: B. Leibe Slides ML, 2015

Part 3, Non-parametric Approaches

- Histograms
- Kernel methods
- K-nearest neighbor

Today’s topics

Recap: Bayes decision theory

Parametric Methods

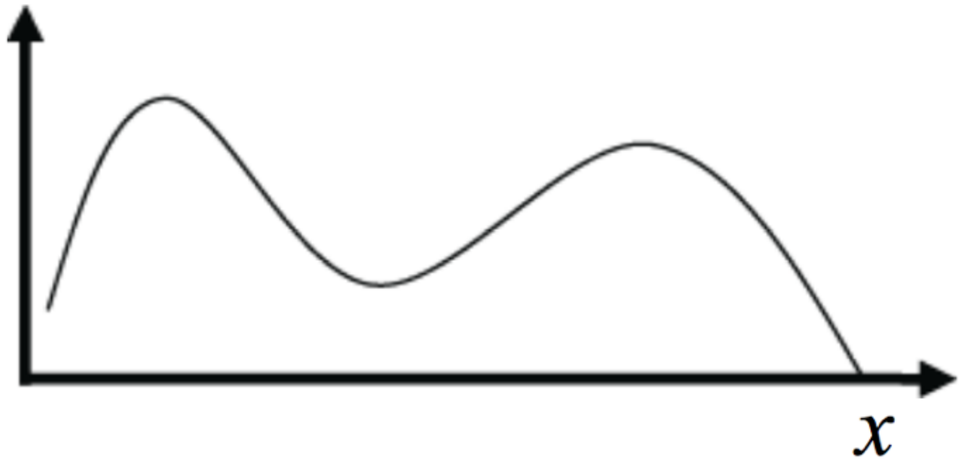
- Recap: Maximum Likelihood approach
- Bayesian Learning

Non-parametric Methods

- **Histograms**
- **Kernel density estimation**
- **K-Nearest Neighbours**
- **k-NN for Classification**

Non-Parametric Methods

- Non-parametric representations
  - Often the functional form of the distribution is unknown



- Estimate probability density from data
  - Histograms
  - Kernel density estimation (Parzen window / Gaussian kernels)
  - k-Nearest-Neighbor

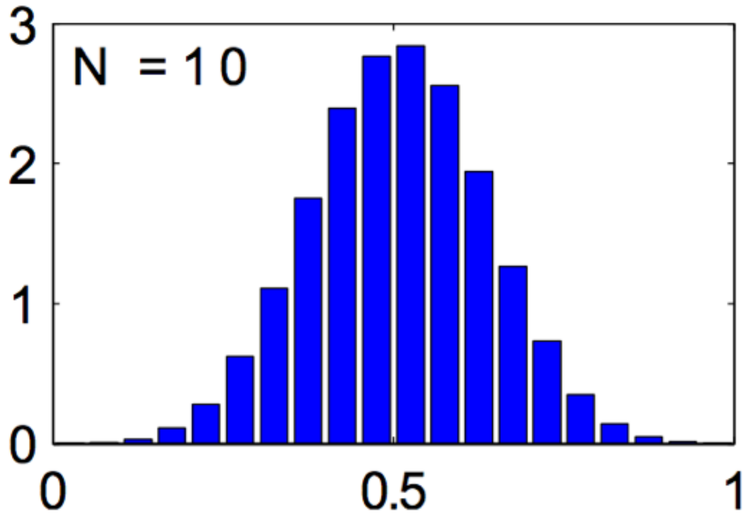
Slide source: B. Leibe Slides ML, 2015

Histograms

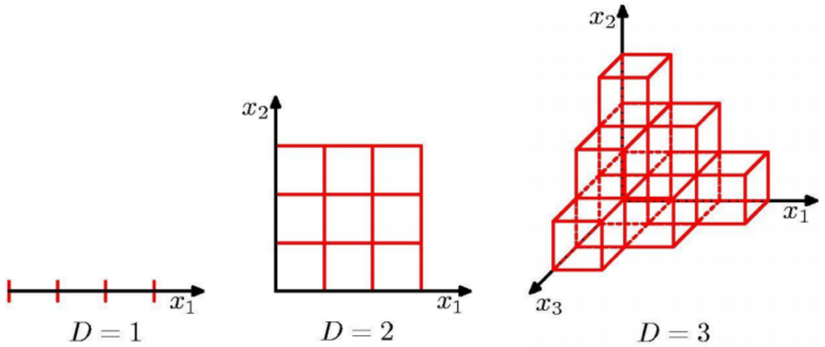
Basic idea:

- Partition the data space into distinct bins with width Δ_i and count the number of observations, n_i, in each bin

$$p_i = \frac{n_i}{N\Delta_i}$$



- Often, the same width is used for all bins, $\Delta_i = \Delta$
- This can be done, in principle, for any dimensionality *D…*



…but the required number of bins grows exponentially with D!

Slide source: B. Leibe Slides ML, 2015

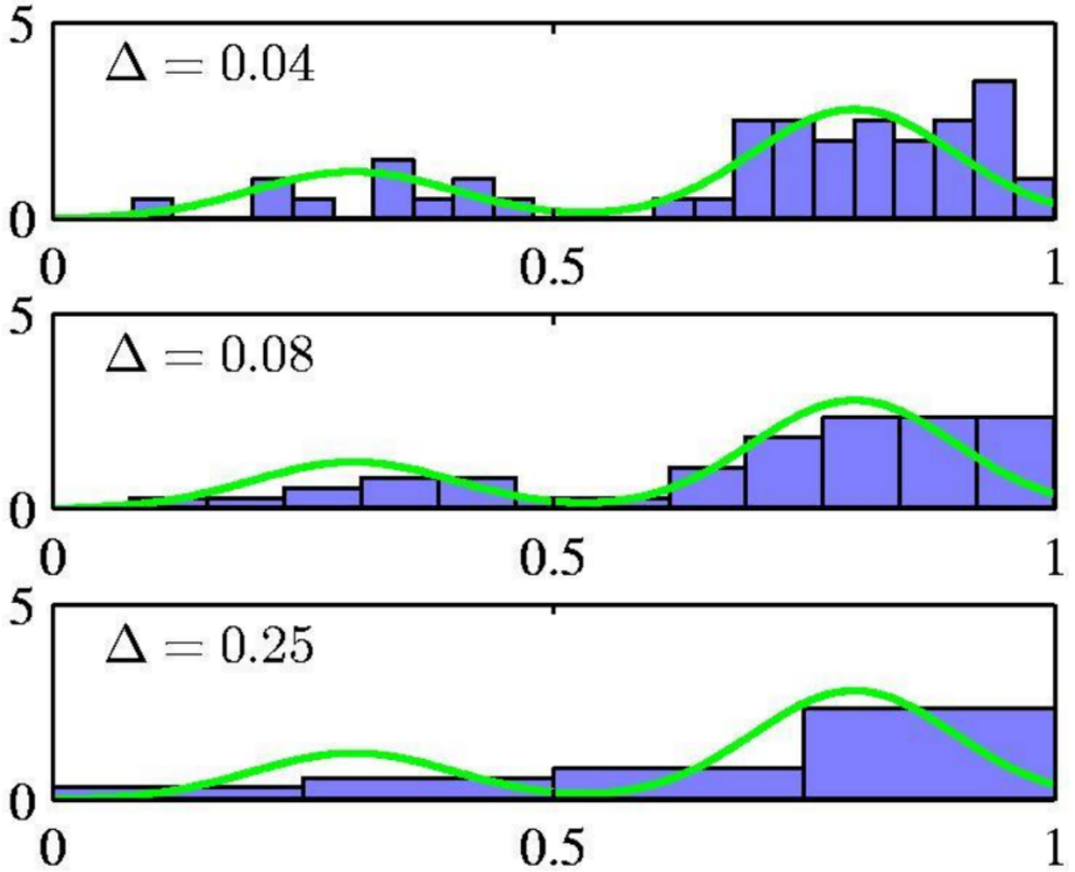
Histograms

The bin width $\triangle$ acts as a smoothing factor

not smooth enough

about OK

too smooth



Slide source: B. Leibe Slides ML, 2015

Summary: Histograms

- **Properties**
  - Very general. In the limit (N->*infinity*), every probability density can be represented.
  - No need to store the data points once histogram is computed.
  - Rather brute-force

- **Problems**
  - High-dimensional feature spaces
    – D-dimensional space with M bins/dimension will require M^D bins!
    => Requires an exponentially growing number of data points
    => “Curse of dimensionality”
  - Discontinuities at bin edges
  - Bin size?
    – too large: too much smoothing
    – too small: too much noise

Slide source: B. Leibe Slides ML, 2015

Statistically Better-Founded Approach

- **Data points** x **comes from pdf** p(x)
  - Probability that x falls into small region R

$$P = \int_{\mathcal{R}} p(y)dy$$

- **If** R **is sufficiently small,** p(x) **is roughly constant**
  - Let V be the volume of R

$$P = \int_{\mathcal{R}} p(y)dy \approx p(\mathbf{x})V$$

- **If the number** N **of samples is sufficiently large, we can estimate** P **as**

$$P = \frac{K}{N} \qquad \Rightarrow p(\mathbf{x}) \approx \frac{K}{NV}$$

Slide source: B. Leibe Slides ML, 2015

Statistically Better-Founded Approach

$$p(\mathbf{x}) \approx \frac{K}{NV}$$

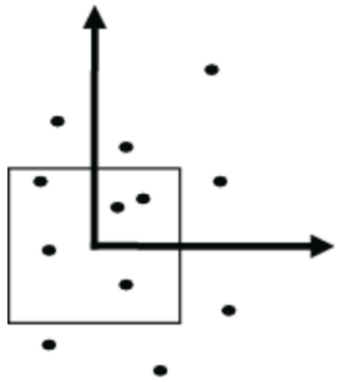
fixed V
determine K

Kernel Methods

fixed K
determine V

K-Nearest Neighbor

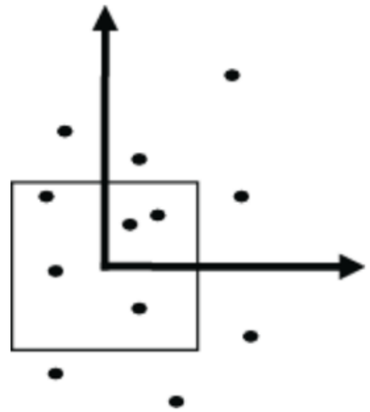
- **Kernel methods**
  - Example: Determine the number K of data points inside a fixed hypercube



Slide source: B. Leibe Slides ML, 2015

Kernel Methods

- **Parzen Window**
  - Hypercube of dimension D with edge length h:



$$k(\mathbf{u}) = \begin{cases} 1, & |u_i| \leq \frac{1}{2}h, \quad i = 1, \ldots, D \\ 0, & else \end{cases}$$

'Kernel function'

$$K = \sum_{n=1}^{N} k(\mathbf{x} - \mathbf{x}_n) \qquad V = \int k(\mathbf{u}) d\mathbf{u} = h^D$$

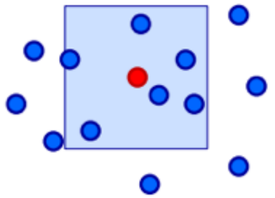
  - Probability density estimate:

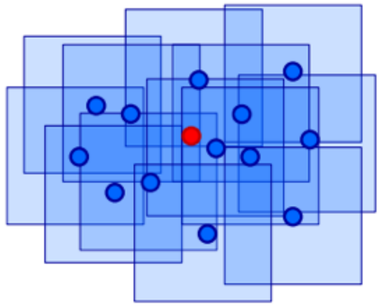
$$p(\mathbf{x}) \approx \frac{K}{NV} = \frac{1}{Nh^D} \sum_{n=1}^{N} k(\mathbf{x} - \mathbf{x}_n)$$

Slide source: B. Leibe Slides ML, 2015

Kernel Methods: Parzen Window

- **Interpretations**
  1. We place a kernel window k at location x and count how many data points fall inside it .
  2. We place a kernel window k around each data point x_n and sum up their influences at location x.
     => Direct visualization of the density





- **Still, we have artificial discontinuities at the cube boundaries…**
  - We can obtain a smoother density model if we choose a smoother kernel function, e.g. a Gaussian

Slide source: B. Leibe Slides ML, 2015

Kernel Methods: Gaussian Kernel

- **Gaussian kernel**
  - Kernel function

$$k(\mathbf{u}) = \frac{1}{(2\pi h^2)^{1/2}} \exp\left\{-\frac{\mathbf{u}^2}{2h^2}\right\}$$

$$K = \sum_{n=1}^{N} k(\mathbf{x} - \mathbf{x}_n) \qquad V = \int k(\mathbf{u}) d\mathbf{u} = 1$$

  - Probability density estimate:

$$p(\mathbf{x}) \approx \frac{K}{NV} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi)^{D/2} h} \exp\left\{-\frac{||\mathbf{x} - \mathbf{x}_n||^2}{2h^2}\right\}$$

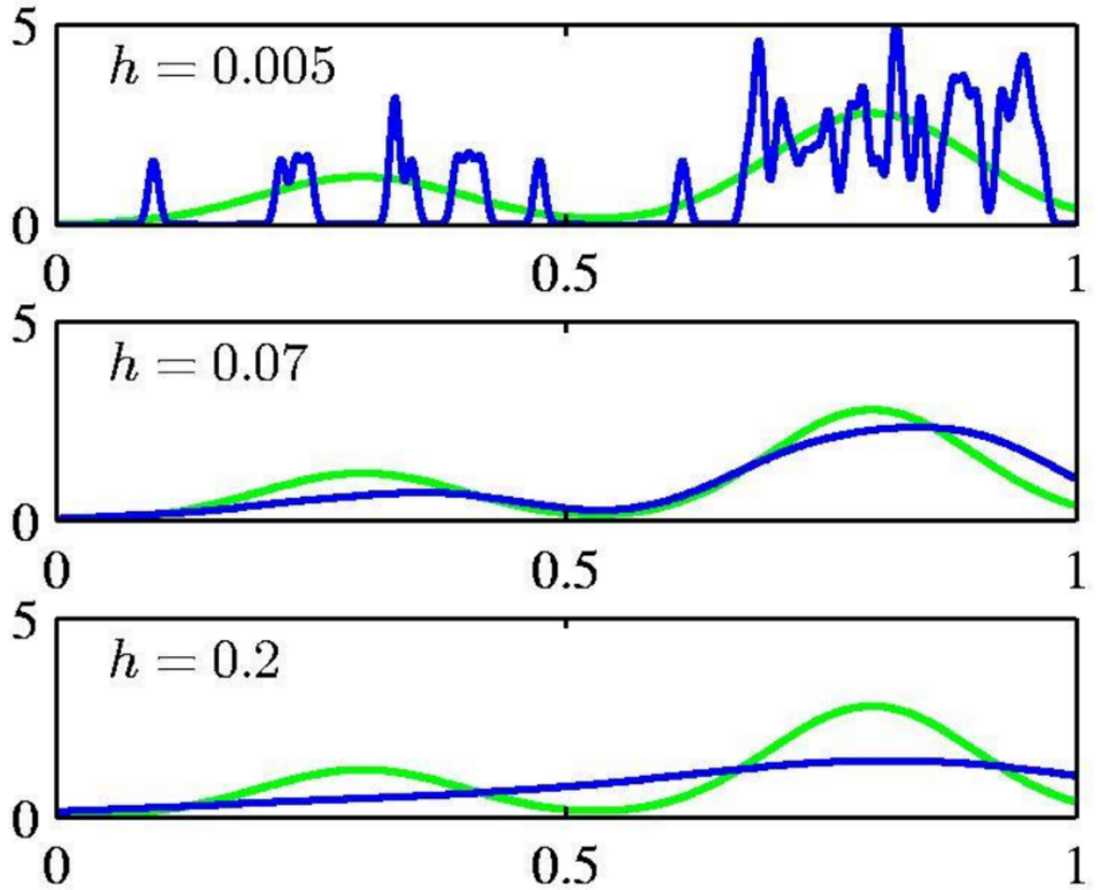
Slide source: B. Leibe Slides ML, 2015

Gaussian Kernel: Examples

not smooth enough

about OK

too smooth



h acts as a smoother.

Slide source: B. Leibe Slides ML, 2015

Kernel Methods

In general

- Any kernel such that

$$k(\mathbf{u}) \geqslant 0, \qquad \int k(\mathbf{u})\,\mathrm{d}\mathbf{u} = 1$$

  can be used. Then

$$K = \sum_{n=1}^{N} k(\mathbf{x} - \mathbf{x}_n)$$

- And we get the probability density estimate

$$p(\mathbf{x}) \approx \frac{K}{NV} = \frac{1}{N} \sum_{n=1}^{N} k(\mathbf{x} - \mathbf{x}_n)$$

Slide source: B. Leibe Slides ML, 2015

Statistically Better-Founded Approach

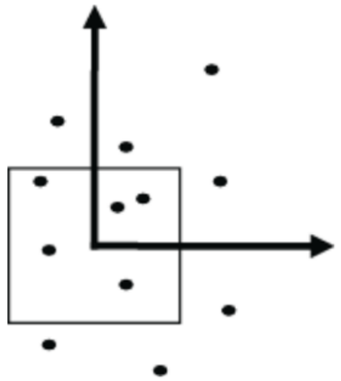
$$p(\mathbf{x}) \approx \frac{K}{NV}$$

fixed V
determine K

Kernel Methods

fixed K
determine V

K-Nearest Neighbor



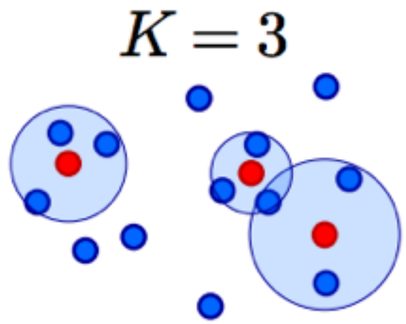
- **K-Nearest Neighbor**
  - Increase the volume V until the K next data points are found.

Slide source: B. Leibe Slides ML, 2015

K-Nearest Neighbor

- **Nearest-Neighbor density estimation**
  - Fix K, estimate V from the data.
  - Consider a hypersphere centred on x and let it grow to a volume V^* that includes K of the given N data points.
  - Then

$$p(\mathbf{x}) \simeq \frac{K}{NV^\star}$$



- **Side note**
  - Strictly speaking, the model produced by K-NN is not a true density model, because the integral over all space diverges.
  - E.g. consider $K = 1$ and a sample exactly on a data point x = x_j.

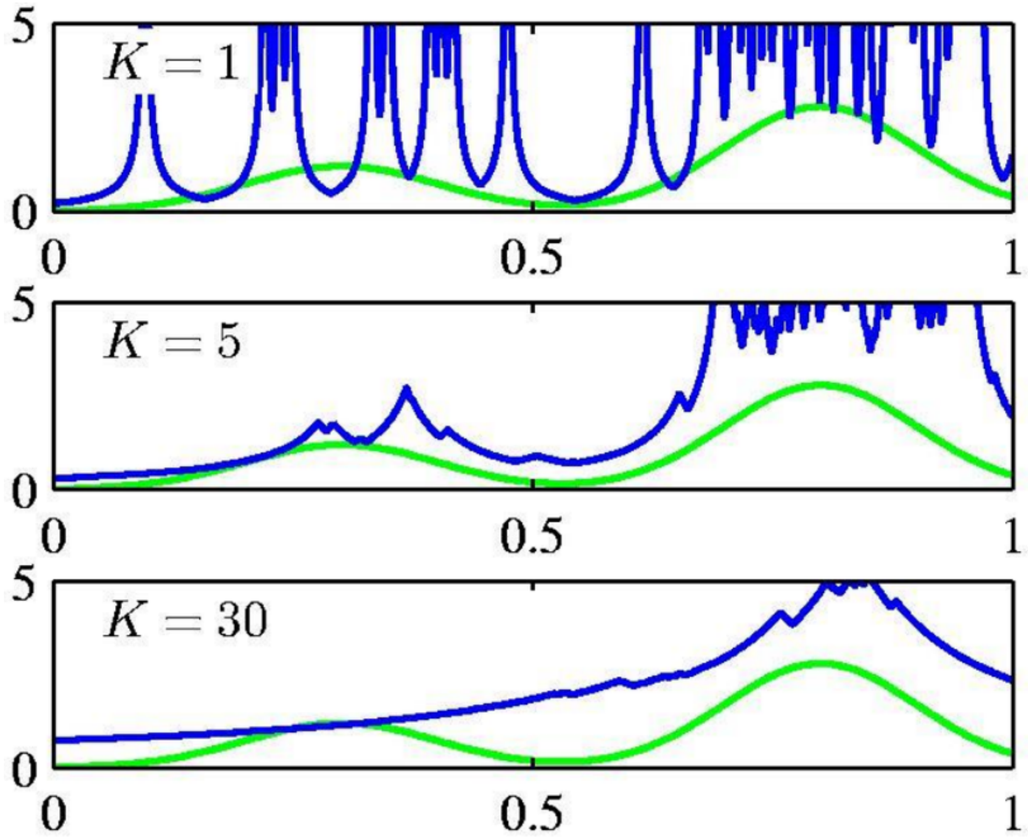
Slide source: B. Leibe Slides ML, 2015

K-Nearest Neighbor: Examples

not smooth enough

about OK

too smooth



K acts as a smoother.

Slide source: B. Leibe Slides ML, 2015

Summary:
Kernel and k-NN Density Estimation

- **Properties**
  - Very general. In the limit (N->*infinity*), every probability density can be represented.
  - No computation involved in the training phase.
    => Simply storage of the training set

- **Problems**
  - Requires storing and computing with the entire dataset.
    => Computational cost linear in the number of data points.
    => This can be improved, at the expense of some computation during training, by constructing efficient tree-based search structures.
  - Kernel size / K in K-NN?
    – too large: too much smoothing
    – too small: too much noise

K-Nearest Neighbor Classification

- **Bayesian Classification**

$$p(\mathcal{C}_j|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)}{p(\mathbf{x})}$$

- **Here we have**

$$p(\mathbf{x}) \approx \frac{K}{NV}$$

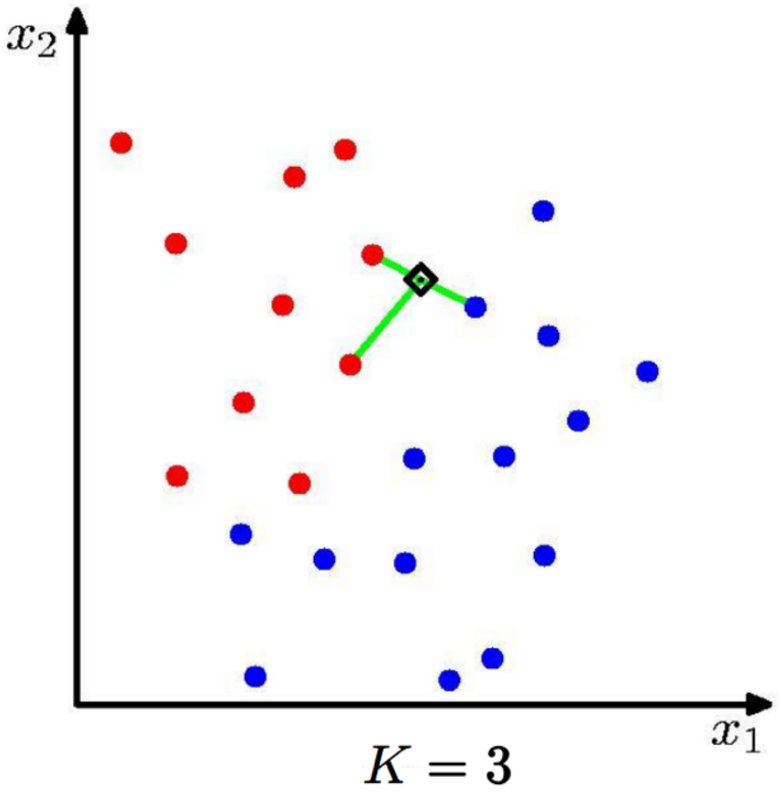
$$p(\mathbf{x}|\mathcal{C}_j) \approx \frac{K_j}{N_j V} \longrightarrow p(\mathcal{C}_j|\mathbf{x}) \approx \frac{K_j}{N_j V}\frac{N_j}{N}\frac{NV}{K} = \frac{K_j}{K}$$

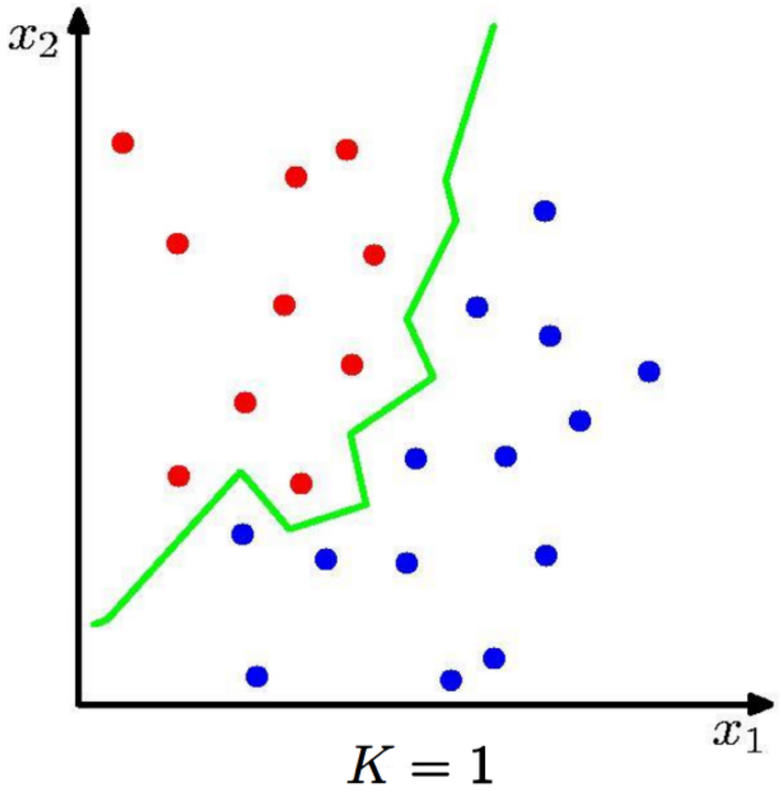
$$p(\mathcal{C}_j) \approx \frac{N_j}{N}$$

K-Nearest Neighbor classification

Slide source: B. Leibe Slides ML, 2015

K-Nearest Neighbor for Classification

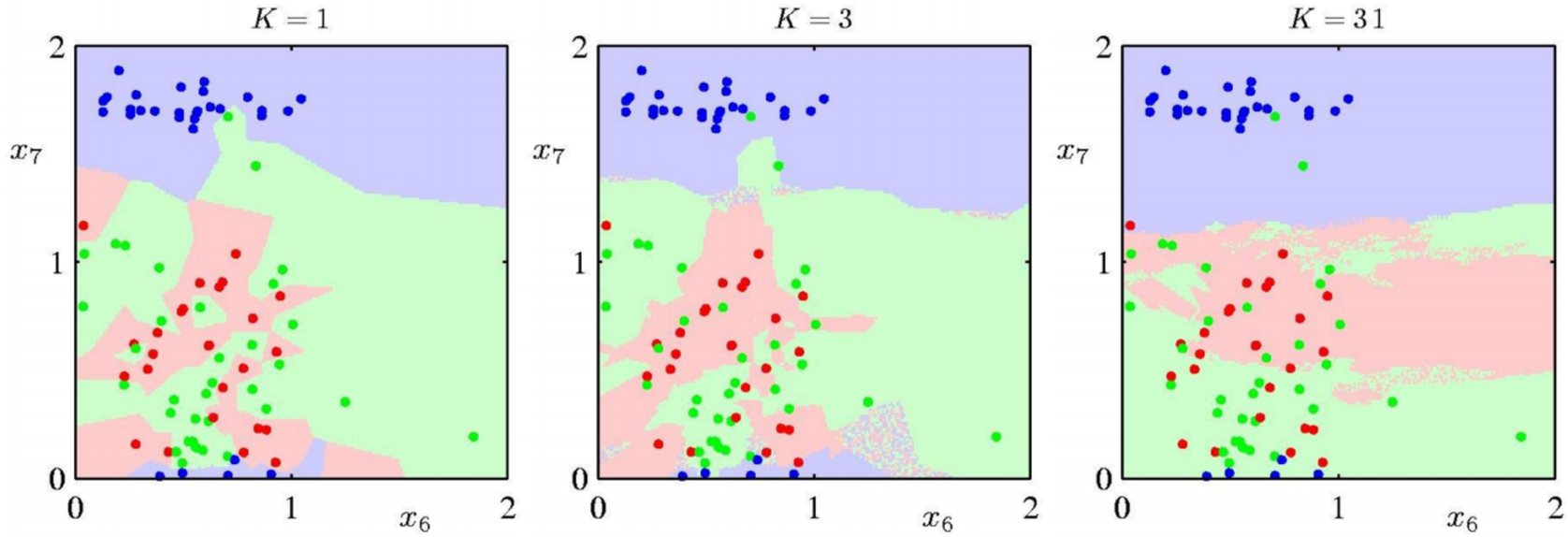




Slide source: B. Leibe Slides ML, 2015

K-Nearest Neighbor for Classification

- **Results on an example data set**



- K **acts as a smoothing parameter.**
- **Theoretical guarantee**
  - For N->*infinity*, the error rate of the 1-NN classifier is never more than twice the optimal error (obtained from the true conditional class distributions).

Slide source: B. Leibe Slides ML, 2015

Bias-Variance Tradeoff

- **Probability density estimation**
  - Histograms: bin size?
    - bin size too large => too smooth
    - bin size too small => not smooth enough
  - Kernel methods: kernel size?
    - kernel size too large => too smooth
    - kernel size too small => not smooth enough
  - K-Nearest Neighbor: K?
    - K size too large => too smooth
    - K size too small => not smooth enough
- **This is a general problem of many probability density estimation methods**
  - Including parametric methods and mixture models

Slide source: B. Leibe Slides ML, 2015

Discussion

- **The methods discussed so far are all simple and easy to apply. They are used in many practical applications.**
- **However…**
  - Histograms scale poorly with increasing dimensionality

    => Only suitable for relatively low-dimensional data.
  - Both k-NN and kernel density estimation require the entire data set to be stored.

    => Too expensive if the data set is large..
  - Simple parametric models are very restricted in what forms of distributions they can represent.

    => Only suitable if the data has the same general form.

- **We need density models that are efficient and flexible!**

Slide source: B. Leibe Slides ML, 2015

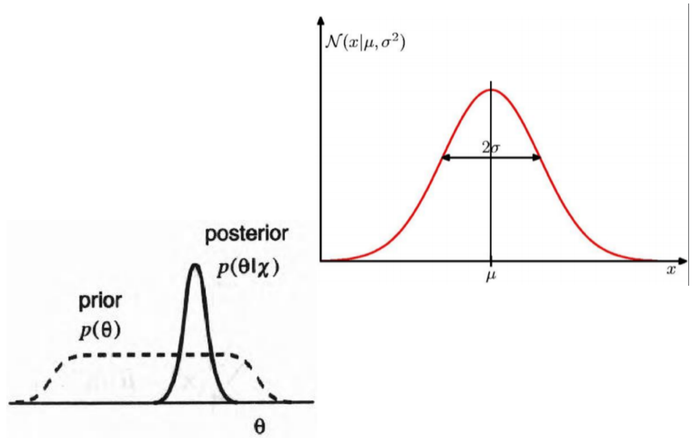
Summary

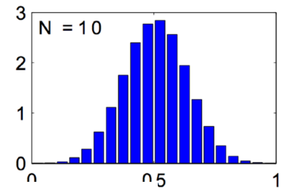
Parametric Methods

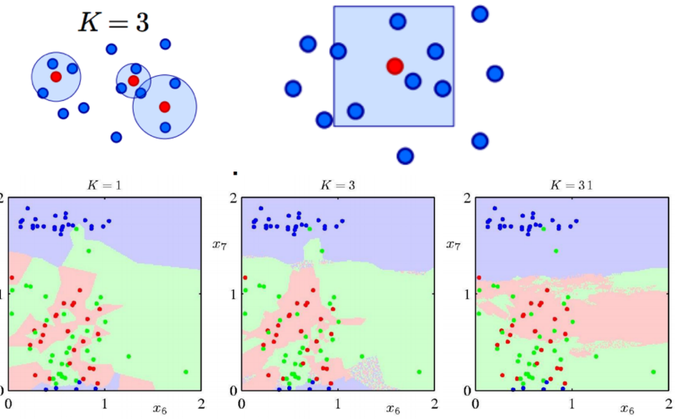
- Gaussian distribution
- Maximum Likelihood approach

Non-parametric Methods

- Histograms
- Kernel density estimation
- K-Nearest Neighbours
- k-NN for Classification







Next topics

Mixture distributions

- Mixture of Gaussians (MoG)
- Maximum Likelihood estimation attempt

K-Means Clustering

- Algorithm
- Applications

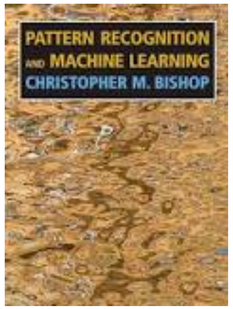
EM Algorithm

- Credit assignment problem
- MoG estimation
- EM Algorithm
- Interpretation of K-Means
- Technical advice

Applications

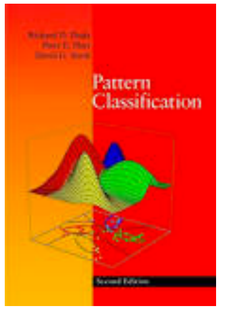
Readings

Bishop’s book



- Gaussian distribution and ML Ch 1.2.4 and 2.3.1-2.3.4
- Bayesian Learning: Ch 1.2.3. and 2.3.6
- Nonparametric methods Ch. 2.5.

Duda & Hart



- ML estimation Ch. 3.2.
- Bayesian Learning: Ch. 3.3-3.5
- Nonparametric methods: Ch. 4.1.-4.5.