Topics of This Lecture

- Recap: Bayes Decision Theory
- Parametric Methods
  - Recap: Maximum Likelihood approach
  - Bayesian Learning
- Non-Parametric Methods
  - Histograms
  - Kernel density estimation
  - K-Nearest Neighbors
  - k-NN for Classification
  - Bias-Variance tradeoff

Recap: Bayes Decision Theory

- Optimal decision rule
  - Decide for $C_i$ if
    \[ p(C_i | x) > p(C_j | x) \]
  - This is equivalent to
    \[ p(x | C_i) p(C_i) > p(x | C_j) p(C_j) \]
  - Which is again equivalent to (Likelihood-Ratio test)
    \[ \frac{p(x | C_i)}{p(x | C_j)} > \frac{p(C_i)}{p(C_j)} \]
  - Decision threshold $\theta$

Recap: Classifying with Loss Functions

- We can formalize the intuition that different decisions have different weights by introducing a loss matrix $L_{kj}$
  \[ L_{kj} = \text{loss for decision } C_j \text{ if truth is } C_k. \]
- Example: cancer diagnosis
  \[ L_{\text{cancer diagnosis}} = \begin{pmatrix} 3 & 0 & 1000 \\ 0 & 1 & 0 \end{pmatrix} \]
Recap: Minimizing the Expected Loss

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

- This can be done by choosing the regions \( R_j \) such that
  \[ \mathbb{E}[L] = \sum_k L_{kj} p(C_k) \]

\( \Rightarrow \) Adapted decision rule:

\[ \frac{p(x; C_1)}{p(x; C_2)} > \frac{(L_{21} - L_{22}) \, p(C_2)}{(L_{12} - L_{11}) \, p(C_1)} \]

Recap: Maximum Likelihood Approach

- Computation of the likelihood
  - Single data point: \( p(x_n \mid \hat{\theta}) \)
    - Assumption: all data points \( X = \{x_1, \ldots, x_n\} \) are independent
      \[ L(\hat{\theta}) = p(X \mid \hat{\theta}) = \prod_{n=1}^N p(x_n \mid \hat{\theta}) \]
    - Log-likelihood
      \[ L(\hat{\theta}) = \sum_{n=1}^N \ln p(x_n \mid \hat{\theta}) \]
  - Estimation of the parameters \( \hat{\theta} \) (Learning)
    - Maximize the likelihood (minimize the negative log-likelihood)
      \( \Rightarrow \) Take the derivative and set it to zero.
      \[ \frac{\partial}{\partial \theta} \mathbb{E}(\hat{\theta}) = - \sum_{n=1}^N \frac{\partial p(x_n \mid \hat{\theta})}{\partial \hat{\theta}} = 0 \]

Recap: Gaussian (or Normal) Distribution

- One-dimensional case
  - Mean \( \mu \)
  - Variance \( \sigma^2 \)
  \[ \mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left\{ - \frac{(x - \mu)^2}{2\sigma^2} \right\} \]

- Multi-dimensional case
  - Mean \( \mu \)
  - Covariance \( \Sigma \)
  \[ \mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{D/2} \mid \Sigma \mid^{1/2}} \exp \left\{ - \frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\} \]

Topics of This Lecture

- Recap: Bayes Decision Theory
- Parametric Methods
  - Recap: Maximum Likelihood approach
  - Bayesian Learning
- Non-Parametric Methods
  - Histograms
  - Kernel density estimation
  - K-Nearest Neighbors
  - k-NN for Classification
  - Bias-Variance tradeoff

Bayesian Approach to Parameter Learning

- Conceptual shift
  - Maximum Likelihood views the true parameter vector \( \theta \) to be unknown, but fixed.
  - In Bayesian learning, we consider \( \theta \) to be a random variable.

- This allows us to use knowledge about the parameters \( \theta \)
  - i.e. to use a prior for \( \theta \)
  - Training data then converts this prior distribution on \( \theta \) into a posterior probability density.

  \[ p(\theta \mid X) \]

- The prior thus encodes knowledge we have about the type of distribution we expect to see for \( \theta \).

Bayesian Learning Approach

- Bayesian view:
  - Consider the parameter vector \( \theta \) as a random variable.
  - When estimating the parameters, what we compute is
    \[ p(x \mid X) = \int p(x \mid \theta, X) \, d\theta \]
    \( \text{Assumption: given } \theta, \text{ this doesn’t depend on } X \text{ anymore} \)
    \[ p(x \mid \theta, X) = p(x \mid \theta, X) \cdot p(\theta \mid X) \]
    \[ p(x \mid \theta, X) = \int p(x \mid \theta, X) \, d\theta \]

  - This is entirely determined by the parameter \( \theta \)
    (i.e. by the parametric form of the pdf).
Bayesian Learning Approach

\[ p(x|X) = \int p(x|\theta)p(\theta|X)d\theta \]

\[ p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)} \]

\[ p(X) = \int p(X|\theta)p(\theta)d\theta = \int L(\theta)p(\theta)d\theta \]

Inserting this above, we obtain

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

Bayesian Density Estimation

- Sample mean: \( \bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \)
- Bayes estimate:
  \[ \mu_N = \frac{\sigma_x^2 \mu_0 + N \sigma_0^2 \bar{x}}{\sigma_x^2 + N \sigma_0^2} \]
- Note:
  \[ \frac{1}{\sigma_N} = \frac{1}{\sigma_0} + \frac{N}{\sigma_x^2} \]

Bayesian Learning Approach

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

Summary: ML vs. Bayesian Learning

- In this lecture, we'll use both concepts wherever appropriate.
Topics of This Lecture

- Recap: Bayes Decision Theory
- Parametric Methods
  - Recap: Maximum Likelihood approach
  - Bayesian Learning
- Non-Parametric Methods
  - Histograms
  - Kernel density estimation
  - K-Nearest Neighbors
  - k-NN for Classification
  - Bias-Variance tradeoff

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

Histograms

- Basic idea:
  - Partition the data space into distinct bins with widths \( \Delta_i \) and count the number of observations, \( n_i \), in each bin.
  - Often, the same width is used for all bins, \( \Delta_i = \Delta \).
  - This can be done, in principle, for any dimensionality \( D \).

  \[
  P_i = \frac{n_i}{N\Delta_i}
  \]

...but the required number of bins grows exponentially with \( D! \)

Summary: Histograms

- Properties
  - Very general. In the limit \( (N \to \infty) \), 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
  - Discontinuities at bin edges
  - Bin size?
    - too large: too much smoothing
    - too small: too much noise

Statistically Better-Founded Approach

- Data point \( x \) comes from pdf \( p(x) \)
  - Probability that \( x \) falls into small region \( \mathcal{R} \)
  
  \[
  P = \int_{\mathcal{R}} p(y) dy
  \]

- If \( \mathcal{R} \) is sufficiently small, \( p(x) \) is roughly constant
  - Let \( V \) be the volume of \( \mathcal{R} \)
  
  \[
  P = \int_{\mathcal{R}} p(y) dy \approx p(x)V
  \]

- If the number \( N \) of samples is sufficiently large, we can estimate \( P \) as
  
  \[
  P = \frac{K}{N} \quad \Rightarrow p(x) \approx \frac{K}{NV}
  \]
Statistically Better-Founded Approach

Kernel Methods

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

Kernel Methods

- K-Nearest Neighbor

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 $x$.
     $\Rightarrow$ 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

Kernel Methods: Gaussian Kernel

- Gaussian kernel
  - Kernel function
    $$ k(u) = \frac{1}{(2\pi h^2)^{D/2}} \exp \left\{ -\frac{u^2}{2h^2} \right\} $$
  - $K = \sum_{n=1}^{N} k(x - x_n)$
  - Probability density estimate:
    $$ p(x) \approx \frac{K}{NV} = \frac{1}{N h^D} \sum_{n=1}^{N} k(x - x_n) $$

Gaussian Kernel: Examples

- Not smooth enough
- About OK
- Too smooth

Kernel Methods

- In general
  - Any kernel such that
    $$ k(u) \geq 0, \quad \int k(u) \, du = 1 $$
  - can be used. Then
    $$ K = \sum_{n=1}^{N} k(x - x_n) $$
  - And we get the probability density estimate
    $$ p(x) \approx \frac{K}{NV} = \frac{1}{N} \sum_{n=1}^{N} k(x - x_n) $$
Statistically Better-Founded Approach

\[ p(x) \approx \frac{K}{NV} \]

- Fixed \( V \) determine \( K \)
- Fixed \( K \) determine \( V \)

Kernel Methods
K-Nearest Neighbor

- K-Nearest Neighbor
  - Increase the volume \( V \) until the \( K \) next data points are found.

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
  \[ p(x) \approx \frac{K}{NV^*}. \]

- 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 \( x = x_j \).

Summary: Kernel and k-NN Density Estimation

- Properties
  - Very general. In the limit (\( N \to \infty \)), every probability density can be represented.
  - No computation involved in the training phase
  \( \Rightarrow \) Simply storage of the training set

- Problems
  - Requires storing and computing with the entire dataset.
  \( \Rightarrow \) Computational cost linear in the number of data points.
  \( \Rightarrow \) 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
  \[ p(C_j|x) = \frac{p(x|C_j)p(C_j)}{p(x)} \]

- Here we have
  \[ p(x) \approx \frac{K}{NV} \]
  \[ p(x|C_j) \approx \frac{K_j}{N_j V} \]
  \[ p(C_j) \approx \frac{N_j}{N} \]

k-Nearest Neighbor classification
**K-Nearest Neighbors for Classification**

- Results on an example data set

  ![Image](image1.png)

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

**Bias-Variance Tradeoff**

- Probability density estimation
  - Histograms: bin size?
    - Too large: too smooth
    - Too small: not smooth enough
  - Kernel methods: kernel size?
    - Too large: too smooth
    - Too small: not smooth enough
  - K-Nearest Neighbor: $K$?
    - Too large: too smooth
    - Too small: not smooth enough

- This is a general problem of many probability density estimation methods
  - Including parametric methods and mixture models

**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.
  - 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!
  - Next lecture...

**References and Further Reading**

- More information in 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.
- Additional information can be found in Duda & Hart
  - ML estimation: Ch. 3.2
  - Bayesian Learning: Ch. 3.3-3.5
  - Nonparametric methods: Ch. 4.1-4.5

*References*

Christopher M. Bishop
*Pattern Recognition and Machine Learning*
Springer, 2006

R.O. Duda, P.E. Hart, D.G. Stork
*Pattern Classification*
2nd Ed., Wiley-Interscience, 2000