Organization

- Lecture 2 hours/week
  - Wed: 14:00 – 16:00, Room: E1.4 024
- Exercises 2 hours/week
  - Thu: 10:00 – 12:00, Room E1.4 024
  - Starts next Thursday
- Course web page: http://www.d2.mpi-inf.mpg.de/gm
  - Slides
  - Pointers to Books and Papers
  - Homework assignments
- “Semesterapparat” in library
- Mailing list: see webpage how to subscribe
Exercises & Exam

▶ Exercises:
  ▶ Typically one assignment per week
  ▶ Typically from Wednesday → Wednesday
  ▶ Theoretical and Practical Exercises
  ▶ Starts this Thursday with Matlab primer
  ▶ Final Grade: 50% exercises, 50% oral exam
    (oral exam has to be passed obviously !)

▶ Exam
  ▶ Oral exam at the end of the semester
  ▶ Can be taken in English or German

▶ Tutors
  ▶ Eldar Insafutdinov (eldar@mpi-inf.mpg.de)
  ▶ Evgeny Levinkov (levinkov@mpi-inf.mpg.de)
Related Classes @UdS

- High-Level Computer Vision (SS), Fritz & Schiele
- Machine Learning (WS), Hein
- Statistical Learning I+II (SS,WS), Lengauer
- Optimization I+II, Convex Optimization (SS,WS), …
- Pattern and Speech Recognition (WS), Klakow
Offers in our Research Group

- Master- and Bachelor Theses
- HiWi-positions, etc.

in
- Topics in machine learning
- Topics in computer vision
- Topics in machine learning applied to computer vision

- Come, talk to us
Literature

- All books in a “Semesterapparat”
- Main book for the graphical model part
- Extra References
Literature

- Bayesian Reasoning and Machine Learning by David Barber
- Pattern Recognition and Machine Learning by Christopher M. Bishop
- Probabilistic Graphical Models: Principles and Techniques by David M. Blei, Peter D. Cunningham, Martin J. Wainwright
- Information Theory, Inference, and Learning Algorithms by David J. C. MacKay
Topic overview 2016/17

- Recap: Probability and Decision theory (today)
- Graphical Models
  - Basics (Directed, Undirected, Factor Graphs)
  - Inference
  - Learning
- Inference
  - Deterministic Inference (Sum-Product, Junction Tree)
  - Approximate Inference (Loopy BP, Sampling, Variational)
- Application to Computer Vision Problems
  - Body Pose Estimation
  - Object Detection
  - Semantic Segmentation
  - Image Denoising
  - ...
Today’s topics

- Overview: Machine Learning
  - What is machine learning?
  - Different problem settings and examples
- Probability theory
- Decision theory, inference and decision
Machine Learning

Overview
Machine learning – what’s that?

- Do you use machine learning systems already?
- Can you think of an application?
- Can you define the term “machine learning”?
Goal of machine learning:
- Machines that learn to perform a task from experience

We can formalize this as

$$y = f(x; w)$$  \hspace{1cm} (1)

$y$ is called *output variable*,
$x$ the *input variable* and
$w$ the model parameters (typically learned)

Classification vs regression:
- regression: $y$ continuous
- classification: $y$ discrete (e.g. class membership)
Goal of machine learning:
- Machines that learn to perform a task from experience

We can formalize this as

\[ y = f(x; w) \]  \hspace{1cm} (2)

- \( y \) is called output variable,
- \( x \) the input variable and
- \( w \) the model parameters (typically learned)

- learn... adjust the parameter \( w \)
- ... a task ... the function \( f \)
- ... from experience using a training dataset \( \mathcal{D} \), where of either
  \[ \mathcal{D} = \{x_1, \ldots, x_n\} \] or \[ \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \]
Different Scenarios

- Unsupervised Learning
- Supervised Learning
- Reinforcement Learning

- Let's discuss
Supervised Learning

- Given are pairs of training examples from $\mathcal{X} \times \mathcal{Y}$

$$
\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad (3)
$$

- Goal is to learn the relationship between $x$ and $y$
- Given a new example point $x$ predict $y$

$$
y = f(x; w) 
$$

- We want to generalize to unseen data
Supervised Learning – Examples

Face Detection
Supervised Learning – Examples
Supervised Learning – Examples

Semantic Image Segmentation
Supervised Learning – Examples

Body Part Estimation (in Kinect)
Figure from *Decision Tree Fields*, Nowozin et al., ICCV11
Supervised Learning – Examples

- Person identification
- Credit card fraud detection
- Industrial inspection
- Speech recognition
- Action classification in videos
- Human body pose estimation
- Visual object detection
- Prediction survival rate of a patient
- ...

Andres & Schiele (MPII)  Probabilistic Graphical Models  October 26, 2016
Supervised Learning - Models

Flashing more keywords

- Multilayer Perceptron (Backpropagation)
- (Deep) Convolutional Neural Networks (Backpropagation)
- Linear Regression, Logistic Regression
- Support Vector Machine (SVM)
- Boosting
- Graphical models
Unsupervised Learning

- We are given some input data points

\[ D = \{ x_1, x_2, \ldots, x_n \} \]  \hspace{1cm} (5)

- Goals:
  - Determine the data distribution \( p(x) \) → density estimation
  - Visualize the data by projections → dimensionality reduction
  - Find groupings of the data → clustering
Unsupervised Learning – Examples

Image Priors for Denoising
Unsupervised Learning – Examples

Image Priors for Inpainting

Image from “A generative perspective on MRFs in low-level vision”, Schmidt et al., CVPR2010

black line: statistics form original images, blue and red: statistics after applying two different algorithms
Human Shape Model

SCAPE: Shape Completion and Animation of People, Anguelov et al.
Unsupervised Learning – Examples

- Clustering scientific publications according to topics
- A generative model for human motion
- Generating training data for Microsoft Kinect xBox controller
- Clustering flickr images
- Novelty detection, predicting outliers
  - Anomaly detection in visual inspection
  - Video surveillance
Just *flashing* some keywords (→ Machine Learning)

- Mixture Models
- Neural Networks
- K-Means
- Kernel Density Estimation
- Principal Component Analysis (PCA)
- Graphical Models (here)
Reinforcement Learning

- **Setting**: given a situation, find an action to maximize a reward function
- **Feedback**:
  - we only get feedback of how well we are doing
  - we do *not* get feedback what the best action would be ("indirect teaching")
- **Feedback given as reward**:
  - each action yields reward, or
  - a reward is given at the end (e.g. robot has found his goal, computer has won game in Backgammon)

- **Exploration**: try out new actions
- **Exploitation**: use known actions that yield high rewards
- Find a good trade-off between exploration and exploitation
Variations of the general theme

- All problems fall in these broad categories
- But your problem will surely have some extra twists
- Many different variations of the aforementioned problems are studied separately
- Let’s look at some ...
Semi-Supervised Learning

- We are given a dataset of $l$ labeled examples
  \[ D_l = \{(x_1, y_1), \ldots, (x_l, y_l)\} \]
  as in supervised learning
- Additionally we are given a set of $u$ unlabeled examples
  \[ D_u = \{x_{l+1}, \ldots, x_{l+u}\} \]
  as in unsupervised learning
- Goal is $y = f(x; w)$
- Question: how can we utilize the extra information in $D_u$?
Semi-Supervised Learning: Two Moons

- Two labeled examples (red and blue) and additional unlabeled black dots
Transductive Learning

- We are given a set of labeled examples

\[ D = \{ (x_1, y_1), \ldots, (x_n, y_n) \} \]  \hspace{1cm} (6)

- Additionally we know the test data points \( \{ x_{te_1}, \ldots, x_{te_m} \} \)
  (not their labels!)

- Can we do better, including this knowledge?

- This should be easier than making predictions for the entire set \( \mathcal{X} \)
On-line Learning

- The training data is presented step-by-step and is never available entirely.
- At each time-step $t$ we are given a new datapoint $x_t$ (or $(x_t, y_t)$).
- When is online learning a sensible scenario?
  - We want to continuously update the model – we can train a model with little data, but the model should become better over time when more data is available (similar to how humans learn).
  - We have limited storage for data and the model – a viable setting for large-scale datasets (e.g. the size of the internet).
- How do we learn in this scenario?
Large-Scale Learning

- Learning with millions of examples
- Study fast learning algorithms (e.g. parallelizable, special hardware)
- Problems of storing the data, computing the features, etc.
- There is no strict definition for “large-scale”
- Small-scale learning: limiting factor is number of examples
- Large-scale learning: limited by maximal time for computation (and/or maximal storage capacity)
Active Learning

- We are given a set of examples
  \[ \mathcal{D} = \{x_1, \ldots, x_n\} \]  

- Goal is to learn \( y = f(x; w) \)

- Each label \( y_i \) costs something, e.g. \( C_i \in \mathbb{R}_+ \)

- Question: How to learn well while paying little?

- This is almost always the case, labeling is expensive
Structured Output Learning

- We are given a set of training examples
  \[ \mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}, \]
  but \( y \in \mathcal{Y} \) contains more structure than \( y \in \mathbb{R} \) or \( y \in \{-1, 1\} \).

- Consider binary image segmentation
  - \( y \) is entire image labeling
  - \( \mathcal{Y} \) is the set of all labelings \( 2^{\#\text{pixels}} \).

- Other examples: \( y \) could be a graph, a tree, a ranking, \ldots

- Goal is to learn a function \( f(x, y; w) \) and predict
  \[ y = \arg\max_{\bar{y} \in \mathcal{Y}} f(x, \bar{y}; w) \]
Some final comments

- All topics are under active development and research
- Supervised classification: basically understood
- Broad range of applications, many exciting developments
- Adopting a “ML view” has far reaching consequences, it touches problems of empirical sciences in general
Probability Theory

Brief Review
A random variable (RV) \( X \) can take values from some discrete set of outcomes \( \mathcal{X} \).

We usually use the short-hand notation

\[
p(x) \quad \text{for} \quad p(X = x) \in [0, 1]
\]

for the probability that \( X \) takes value \( x \)

With

\[
p(X),
\]

we denote the probability distribution over \( X \)
Brief Review

- Two random variables (RVs) are called independent if

\[ p(X = x, Y = y) = p(X = x)p(Y = y) \]  \hspace{1cm} (10)

- Joint probability (of \(X\) and \(Y\))

\[ p(x, y) \text{ instead } p(X = x, Y = y) \]  \hspace{1cm} (11)

- Conditional probability

\[ p(x|y) \text{ instead } p(X = x|Y = y) \]  \hspace{1cm} (12)
The Rules of Probability

- **Sum rule**

\[
p(X) = \sum_{y \in Y} p(X, Y = y) \tag{13}
\]

we “marginalize out \(y\)”. \(p(X = x)\) is also called a marginal probability

- **Product Rule**

\[
p(X, Y) = p(Y|X)p(X) \tag{14}
\]

- **And as a consequence:** **Bayes Theorem or Bayes Rule**

\[
p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \tag{15}
\]
Vocabulary

- **Joint Probability**
  \[ p(x_i, y_j) = \frac{n_{ij}}{N} \]

- **Marginal Probability**
  \[ p(x_i) = \frac{c_i}{N} \]

- **Conditional Probability**
  \[ p(y_j \mid x_i) = \frac{n_{ij}}{c_i} \]

\[ c_i = \sum_j n_{ij} \]
\[ N = \sum_{ij} n_{ij} \]
Now $X$ is a continuous random variable, e.g., taking values in $\mathbb{R}$.

Probability that $X$ takes a value in the interval $(a, b)$ is

\[
p(X \in (a, b)) = \int_a^b p(x) \, dx
\]

and we call $p(x)$ the probability density over $x$. 

\[\text{(16)}\]
Probability Densities

- $p(x)$ must satisfy the following conditions

\[
p(x) \geq 0 \quad (17)
\]

\[
\int_{-\infty}^{\infty} p(x) \, dx = 1 \quad (18)
\]

- The probability that $x$ lies in $(-\infty, z)$ is given by the cumulative distribution function

\[
P(z) = \int_{-\infty}^{z} p(x) \, dx \quad (19)
\]
Probability Densities

Figure: Probability density of a continuous variable
Expectation and Variances

▶ Expectation

\[
\mathbb{E}[f] = \sum_{x \in \mathcal{X}} p(x) f(x) \tag{20}
\]

\[
\mathbb{E}[f] = \int_{x \in \mathcal{X}} p(x) f(x) \, dx \tag{21}
\]

▶ Sometimes we denote the distribution that we take the expectation over as a subscript, eg

\[
\mathbb{E}_{p(\cdot|y)}[f] = \sum_{x \in \mathcal{X}} p(x|y) f(x) \tag{22}
\]

▶ Variance

\[
\text{var}[f] = \mathbb{E} \left[ (f(x) - \mathbb{E}[f(x)])^2 \right] \tag{23}
\]
Decision Theory
Digit Classification

- Classify digits “a” versus “b”

Figure: The digits “a” and “b”

- Goal: classify new digits such that the error probability is minimized
Digit Classification - Priors

Prior Distribution

How often do the letters “a” and “b” occur?

Let us assume

\[ C_1 = a \quad p(C_1) = 0.75 \]  \hspace{1cm} (24)
\[ C_2 = b \quad p(C_2) = 0.25 \]  \hspace{1cm} (25)

The prior has to be a distribution, in particular

\[ \sum_{k=1,2} p(C_k) = 1 \]  \hspace{1cm} (26)
We describe every digit using some **feature vector**
- the number of black pixels in each box
- relation between width and height

Likelihood: How likely has $x$ been generated from $p(\cdot \mid a)$, resp. $p(\cdot \mid b)$?

![Graph showing distributions $p(x|a)$ and $p(x|b)$]
Digit Classification

- Which class should we assign $x$ to?
- The answer
- Class $a$
Digit Classification

Which class should we assign $x$ to?
- The answer
- Class b
Digit Classification

- Which class should we assign $x$ to?
- The answer
- Class a, since $p(a) = 0.75$
Bayes Theorem

How do we formalize this?

We already mentioned Bayes Theorem

\[ p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \quad (27) \]

Now we apply it

\[ p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)} \quad (28) \]
Bayes Theorem

- Some terminology! Repeated from last slide:

\[
p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)}
\]

- We use the following names

\[
\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalization Factor}}
\]

- Here the normalization factor is easy to compute. Keep an eye out for it, it will haunt us until the end of this class (and longer :)

- It is also called the Partition Function, common symbol \(Z\)
Bayes Theorem

\[ p(x|a)P(a) = p(x|b)P(b) = \text{Likelihood} \times \text{Prior} \]

\[ p(a|x) = \frac{\text{Likelihood} \times \text{Prior}}{\text{Normalization Factor}} \]

Likelihood

Posterior
How to Decide?

- Two class problem $C_1, C_2$, plotting Likelihood $\times$ Prior
Minimizing the Error

\[ p(\text{error}) = p(x \in R_2, C_1) + p(x \in R_1, C_2) \]
\[ = p(x \in R_2|C_1)p(C_1) + p(x \in R_1|C_2)p(C_2) \]  \hspace{1cm} (31)

\[ = \int_{R_2} p(x|C_1)p(C_1)dx + \int_{R_1} p(x|C_2)p(C_2)dx \] \hspace{1cm} (32)

\[ = \int_{R_2} p(x|C_1)p(C_1)dx + \int_{R_1} p(x|C_2)p(C_2)dx \] \hspace{1cm} (33)
General Loss Functions

- So far we considered misclassification error only
- This is also referred to as 0/1 loss
- Now suppose we are given a more general loss function

\[ \Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+ \]  
\[ (y, \hat{y}) \mapsto \Delta(y, \hat{y}) \]  

- How do we read this?
- \( \Delta(y, \hat{y}) \) is the cost we have to pay if \( y \) is the true class but we predict \( \hat{y} \) instead
Example: Predicting Cancer

\[ \Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \]
\[ (y, \hat{y}) \mapsto \Delta(y, \hat{y}) \]  

Given: X-Ray image, Question: Cancer yes or no? Should we have another medical check of the patient?

<table>
<thead>
<tr>
<th>diagnosis:</th>
<th>cancer</th>
<th>normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>truth:</td>
<td>cancer</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>normal</td>
<td>1</td>
</tr>
</tbody>
</table>

For discrete sets \( \mathcal{Y} \) this is a loss matrix.
Digit Classification

Which class should we assign $x$ to? ($p(a) = p(b) = 0.5$)

The answer

It depends on the loss
Minimizing Expected Loss (or Error)

- The expected loss for $x$ (averaged over all decisions)

$$
\mathbb{E}[\Delta] = \sum_{k=1,\ldots,K} \sum_{j=1,\ldots,K} \int_{R_j} \Delta(C_k, C_j) p(x, C_k) \, dx
$$

- And how do we predict? Decide on one $y$!

$$
y^* = \arg\min_{y \in \mathcal{Y}} \sum_{k=1,\ldots,K} \Delta(C_k, y) p(C_k | x)
$$

$$
= \arg\min_{y \in \mathcal{Y}} \mathbb{E}_{p(\cdot | x)}[\Delta(\cdot, y)]
$$
Inference and Decision

- We broke down the process into two steps
  - **Inference**: obtaining the probabilities \( p(C_k|x) \)
  - **Decision**: Obtain optimal class assignment
- Two steps !!
- The probabilites \( p(\cdot|x) \) represent our belief of the world
- The loss \( \Delta \) tells us what to do with it!
- 0/1 loss implies deciding for max probability (exercise)
Three Approaches to Solve Decision Problems

1. **Generative models**: infer the class conditionals

   \[ p(x|C_k), \ k = 1, \ldots, K \]  
   \[ \text{(41)} \]

   then combine using Bayes Theorem

   \[ p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} \]

2. **Discriminative models**: infer posterior probabilities directly

   \[ p(C_k|x) \]  
   \[ \text{(42)} \]

3. Find a **discriminative function** minimizing Expected Loss \( \Delta \)

   \[ f : \mathcal{X} \to \{1, \ldots, K\} \]  
   \[ \text{(43)} \]

Let’s discuss these options
Generative Models

Pros:
- The name *generative* is because we can *generate* samples from the learnt distribution
- We can infer $p(x|C_k)$ (or $p(x)$ for short)

Cons:
- With high dimensionality of $x \in \mathcal{X}$ we need a large training set to determine the class-conditionals
- We may not be interested in all quantities
Discriminative Models

Pros:
- No need to model $p(x|C_k)$ (i.e. in general easier)

Cons:
- No access to model $p(x|C_k)$
Discriminative Functions

When solving a problem of interest, do not solve a harder / more general problem as an intermediate step.

– Vladimir Vapnik

Pros:
▶ One integrated system, we directly estimate the quantity of interest

Cons:
▶ Need $\Delta$ during training time – revision requires re-learning
▶ No access to probabilities or uncertainty, thus difficult to reject decision?
▶ Prominent example: Support Vector Machines (SVMs)
Next Time ... 

▶ ... we will meet our new friends:

(a) Bayesian Network
(b) Markov Random Field
(c) Factor Graph