Fundamentals of Artificial Intelligence

Introduction to Machine Learning

Adam Dahlgren
Department of Computing Science

Umeå University

Outline

- Different types of learning
- Concrete example of supervised learning
    - k-nearest neighbour-algorithm
- Some learning theory (over-/underfitting)
- Introduction of Assignment 2

Rules during this lecture

- Whenever there is a question, at least one person must answer.
- I might lose track of space and time if interrupted
- But, whenever you do not understand, interrupt me.
    - You are most likely not alone
    - I might not understand either
    - Ask in Swedish if you are uncomfortable with English.
    - I want two volunteers that keep an eye on the chat in case I miss questions.
- Asking the right questions is as important as knowing the right answers.
- Let me know if I start to mumble.
- Remind me when you need a break

Introduction

What is machine learning?

- Webster's Dictionary:
  . . . modification of a behavioral tendency by experience.
- What does this mean in our context?
  - Behaviour is the output of our system (e.g. actions for robots)
  - Experience is previously sensed data
- At what stage do we learn?
  - Online vs. offline learning

We want to modify the mapping from sensing to action.

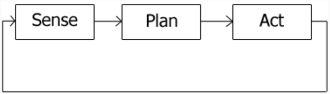

Figure 1: Traditional view of an intelligent agent.

Machine learning: examples

Can someone give an example of machine learning in practice?

- Machine translation
- Data center management
- Computer-aided diagnostic systems
- Insurance premiums
- Control systems

What drives these applications? Data.

Detexify

The tip of the day

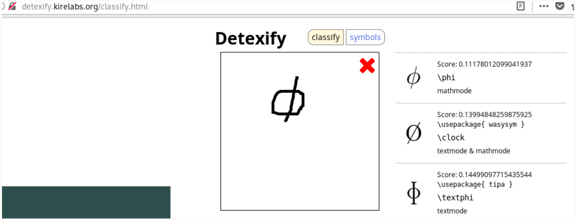

Figure 2: Example of a machine learning powered application

Types of Learning

We have three main categories of machine learning

- Supervised Learning
  - Using a teacher to correct a classifier until it can handle given example input/output pairs.
- Unsupervised Learning
  - We have no labeled data and want to e.g. do clustering (recognize communities in social networks).
- Reinforcement Learning
  - Given e.g. an action, does that lead to a positive or negative outcome? Adjust accordingly, similarly to supervised learning. Common in robotics.

Types of classifiers

We can define different types of classification tasks

- Binary classification
- Multiclass classification
- Regression
- Ranking
- Structured prediction

Binary/multiclass classification

Given an input x, distinguish between two (or more) classes

- Binary classification
    - Determine whether an email (x) is spam or not (y).
    - Here y is a discrete value yes/no.
- Multiclass classification
    - Given an image (x), what object (y) is shown? (E.g. cat, football, et c.)
    - Here y belongs to a category.

Regression/Ranking

We do not always want to label input with a certain class

- Regression
  - Given the temperature (x), predict the electric energy consumption (y)
  - Here y is a continous output.
- Ranking
  - Given a number of matching queries (x), rank them (y) according to relevancy.
  - Mathematically speaking y is a permutation applied to x.

We usually have structure in input (relation between pixels, words in sentence, et c.).

- In structured prediction you also have structure in output.
- Given a sentence (x), label each word (y) with its role in the sentence.

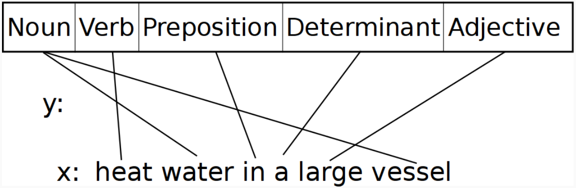

Figure 3: Part-of-Speech tagging

Supervised Learning

Supervised Learning

A set of examples is provided

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

- Examples are divided in a *traning set* and a *test set*.
- Some true function $f : \mathcal{X} \to \mathcal{Y}$
  - We want a hypothesis h such that $h \approx f$ (by looking at examples).
- Our hypothesis h should generalize, i.e. perform well on both training data *and* unseen examples.
- The test set is used to estimate the generalized performance of h.

Q: How does supervise learning tackle the frame problem?

Learning theory

We have a couple of concepts

- Hypothesis space
  - E.g. all linear functions
- Empirical risk/loss minimization
  - Loss function that measures how well the prediction matches the true output.
  - Choose a hypothesis that minimizes this loss over all samples.
- Computational complexity
  - How complex is the relationship between the input- and output space?
  - How expressive is the classifier?

An algorithm for (supervised) learning

1. Collect a (large) set of examples
2. Divide them randomly into a training set and a test set
3. Generate a hypothesis using the training set
4. Measure the performance on examples from the test set (e.g. number of correctly classified examples).
5. Repeat until satisfactory performance is achieved.

This produces a *model* of the relationship between the input/output space that the classifier can use later.

Example: Curve fitting

Construct a h that fits the traning set (i.e., h is consistent with f on all examples).

- We have a couple of samples (x_i, y_i).
- We would like to fit a line that explains this data.
- This approximation should ideally generalize well.

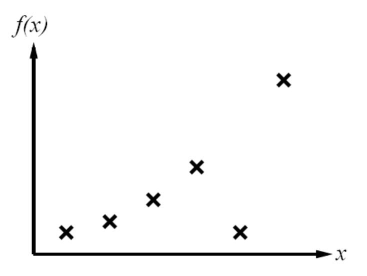

Figure 4: Some samples from a 2D data set

Example: Curve fitting

Linear regression using e.g. the least squares method

$$h(x) = k_0 + k_1 x$$

$$S = \sum_{i=1}^{n} r_i^2$$

What about the bottom-rightmost sample?

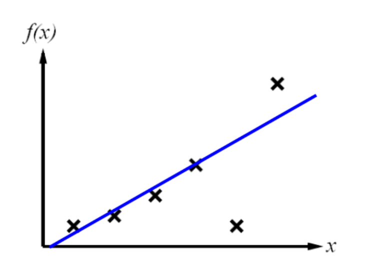

Figure 5: Linear fit to sample data

Example: Curve fitting

We can try quadratic function as our hypothesis.

$$h(x) = k_0 + k_1 x + k_2 x^2$$

Given that the bottom-rightmost sample is just noise (or unseen), this hypothesis is a much better alternative.

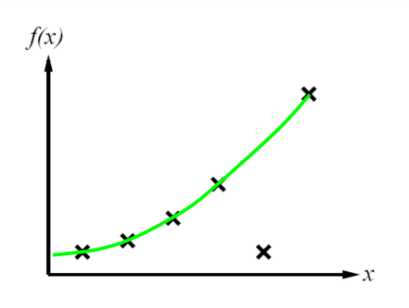

Figure 6: Quadratic hypothesis function.

However, if the bottom-rightmost sample is legit data, we need a better hypothesis if we want to minimize the training error.

We can choose a n-degree polynomial

$$h(x) = k_0 + k_1 x + k_2 x^2 + \ldots + k_n x^n$$

Why might this be a bad idea?

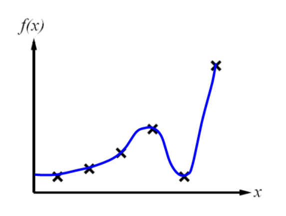

Figure 7: High-degree polynomial hypothesis.

Example: Curve fitting

It is easy to construct a hypothesis function that fits the training data perfectly

Ockham's razor

- Maximize the combination of predictive accuracy with simplicity.
- Penalize too complex hypotheses, either implicitly or explicitly.
- We cannot only minimize the training error.

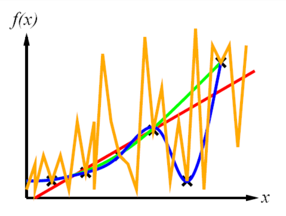

Figure 8: The orange zigzag fits the training data perfectly, but does it generalize well?

Overfitting

In this example the polynomial classifier has a training error of zero, while the linear classifier has a non-zero error.

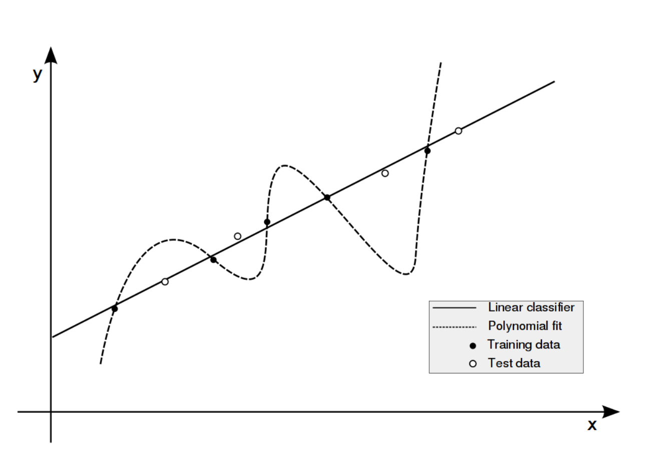

Figure 9: Example of overfitting

Underfitting

Underfitting can occur

- when we have too little training data
- if our classifier cannot express the true complexity of f

We cannot generalize properly.

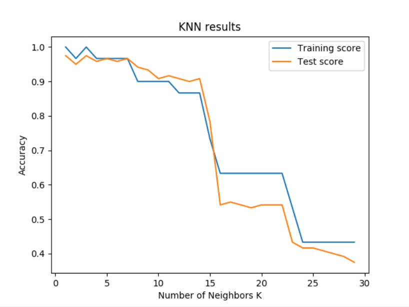

Figure 10: Example of underfitting

Engineering features

In many situations, we want to do some feature extraction/selection

- We can build in domain knowledge into our machine learning algorithm
- What part of the data is really telling us something important?
- Is the original data on a usable form?
- Again, the frame problem
- Bias in AI and machine learning partially stems from this

What might be good features for predicting student success on a course?

Feature engineering: example

Given a piece of text and the task to label it with a topic, what can we do to reduce the dimensionality?

- Remove common words like *is, the* with low information density
- Normalize words
  - Only lowercase
  - Word stemming - *manage* and *managed* $\rightarrow$ *manag*
- Application-dependent

Feature engineering: example

Given a piece of text and the task to label it with a topic, what can we do to reduce the dimensionality?

- Usually represented as a feature **vector**
  - Mathematical properties that we can utilize
  - Requires a feature extraction function mapping from the input space to a vector space.

$$\phi(\text{"Two by two matrix"}) \rightarrow \begin{bmatrix} \vdots \\ 0 \\ 2 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

Sentiment analysis of facial expressions in image

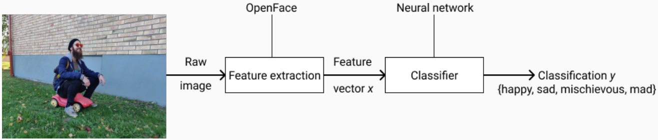

Figure 11: Classification pipeline for image to sentiment analysis

- Feature extraction could be made with e.g. OpenFace [1].
- The input space is the angle on eye brows, size of dimples et c.
- The output space is a set of predefined sentiments.
- Our classifier could be a neural network (as in your ANN assignment).

Intuitively, given an unseen data point it should probably be classified as its closest neighbor

- This idea can be generalized to consider **k** neighbors.
- We want to learn a decision boundary that separates the classifications.

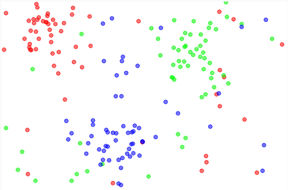

Figure 12: Data points plotted in the input space. Shared under Creative Commons [2].

What happens if we consider too few neighbors?

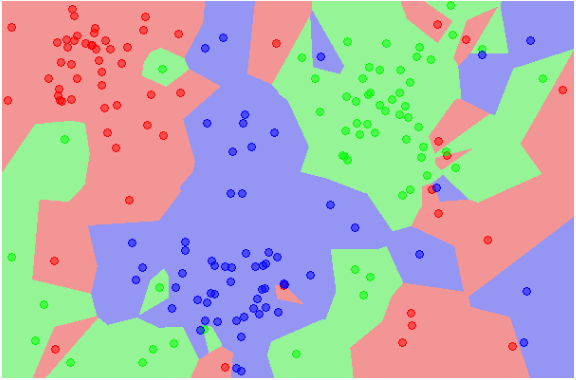

Figure 13: The decision boundaries of a 1-NN classifier. Shared under Creative Commons [3].

Visualization of k-NN decision boundaries

Larger *k* gives smoother decision boundaries

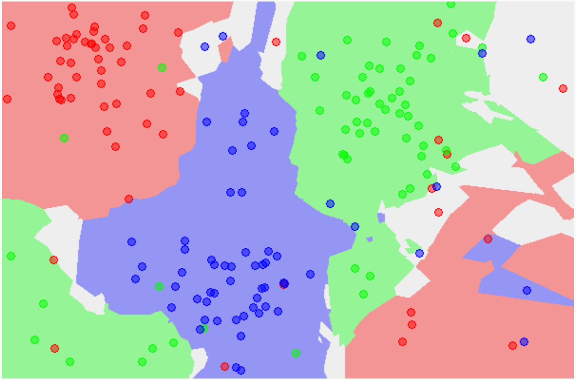

Figure 14: The decision boundaries of a 5-NN classifier. Shared under Creative Commons [4].

Can we consider too many neighbors?

Complexity of the decision boundary

What is the optimal decision boundary?

- Upper- and lower bounds can be formulated using e.g. Kolmogorov complexity, Rademacher complexity et c.
- Complexity usually grows with dimensionality.
- The lower the ratio between the amount of data and the complexity, the harder it is to predict the boundary
- Related to the concept of entropy
  - What if labels are assigned at random?
  - This cannot be generalized

The No Free Lunch theorem [5]

- *"Any two algorithms are equivalent when their performance is averaged across all possible problems"* - Wolpert, 1996.
- Without assumptions on the task, no classifier can be considered superior.
- There is no *best* classifier, choose like you choose your lunch.

Unsupervised Learning

Unsupervised Learning

We have no examples of input-output pairs

- The agent learns relationships between input and output
- Closely related to clustering

Unsupervised learning techniques such as k-means clustering have an array of applications.

- Community detection in social networks
- Anomaly detection in computer systems
- Parallel to e.g. studies on personality traits in psychology

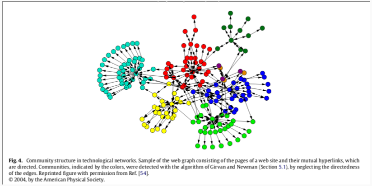

Figure 15: Clustering in social networks for community detection [6].

Reinforcement Learning

Our last category of learning algorithms

- A correct answer y is not provided for each input x.
- Instead, some sort of reward/scoring function is defined
- An agent explores the action space to determine valuable actions
- E.g. the sequences of actions that achieves the highest reward will determine the action y in a game like chess.

Reinforcement learning: example

Playing complex real-time strategy games has been proven really difficult. Recent advancements usually employ reinforcement learning.

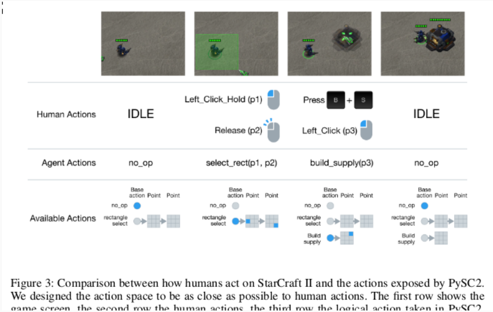

Figure 16: How people at Deepmind plays Starcraft II [7].

Deep learning

Deep Learning

We will not talk about deep learning today. Lets skip that until next lecture, and the ANN flipped classroom sessions.

Assignment 3: k-NN

You should start working on the next assignment today

- You will train a k-NN classifier on two datasets
- Perform an analysis on the output with the theory from this lecture
- Write a neat little report on your findings
- You have a short deadline (one week), working in pairs
- You should **not** spend more than 10 hours on this assignment.

What can be said about Figure 17?

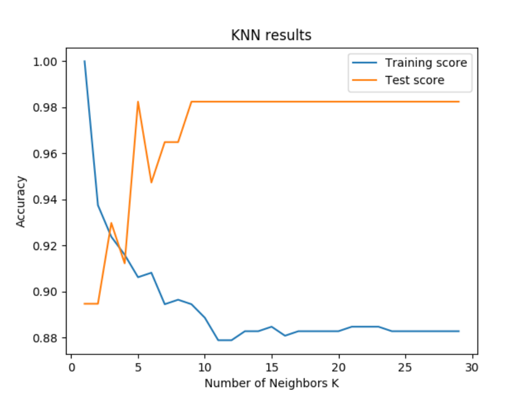

Figure 17: Example run of the given code.

QUESTIONS?

References i

- T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *IEEE Winter Conference on Applications of Computer Vision*, 2016.
- W. Commons, "Data3classes," 2013.
  File: `Data3classes.png`.
- W. Commons, "Map1nn," 2013.
  File: `Map1NN.png`.
- W. Commons, "Map5nn," 2013.
  File: `Map5NN.png`.
- D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Computation*, vol. 8, pp. 1341–1390, 1996.
- S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75 – 174, 2010.

O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, J. Quan, S. Gaffney, S. Petersen, K. Simonyan, T. Schaul, H. van Hasselt, D. Silver, T. P. Lillicrap, K. Calderone, P. Keet, A. Brunasso, D. Lawrence, A. Ekermo, J. Repp, and R. Tsing, "Starcraft II: A new challenge for reinforcement learning," *CoRR*, vol. abs/1708.04782, 2017.