

Literature

There is no exact textbook for the material of the lecture. The introduction is most similar to the draft text "A Course in Machine Learning" by Hal Daumé III: <http://ciml.info/>

Afterwards, we'll also use material from:

- Shai Shalev-Shwartz, Shai Ben-David, "Understanding Machine Learning", 2014.
- Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar "Foundations of Machine Learning", 2012.
- Kevin Murphy, "Machine Learning: A Probabilistic Perspective", 2012.

1 Decision Trees

These are training and test data from the *dating* example in the lecture.

	person	eyes	handsome	height	sex	soccer	date?
TRAINING:	Apu	blue	yes	tall	M	no	yes
	Bernice	brown	yes	short	F	no	no
	Carl	blue	no	tall	M	no	yes
	Doris	green	yes	short	F	no	no
	Edna	brown	no	short	F	yes	no
	Prof. Frink	brown	yes	tall	M	yes	no
	Gil	blue	no	tall	M	yes	no
	Homer	green	yes	short	M	no	yes
	Itchy	brown	no	short	M	yes	yes
	TESTING:	Jimbo	blue	no	tall	M	no
Krusty		green	yes	short	M	yes	no
Lisa		blue	yes	tall	F	no	no
Moe		brown	no	short	M	no	no
Ned		brown	yes	short	M	no	yes
Quimby		blue	no	tall	M	no	yes

Use the training data to construct decision trees and test them on the test data in the following situations (in cases of ties between attributes, choose by alphabetic order)

- if there had been no attribute "soccer".
- if there had been no attribute "eye color".
- if "Itchy" had the label **no** instead of **yes**.

- if there had been one more training example:

person	eyes	handsome	height	sex	soccer	date?
Ralph	green	no	short	M	yes	no

Assume there were D additional attributes with random values **yes** or **no** ($p(\text{yes}) = p(\text{no}) = 0.5$)?

- What is the probability that the training stops (zero training error) after a single split for $D = 10$, for $D = 100$, for $D = 1353$?

2 Nearest Neighbor Classification

- Find three examples where humans perform (more or less) nearest-neighbor classification. What about k -NN?
- What are the advantages and disadvantages of k -NN with $k > 1$ versus 1-NN.
- What is the error rate of 1-NN when applying it to the *training set*? Is the same true for k -NN?
- Assume the following tie breaking rule: if there's no unique majority label for K -NN, use the $(K - 1)$ -decision. Show: for binary classification, $2K$ -NN classification is identical to $(2K - 1)$ -NN classification for any $K \geq 1$.
- Give an example of a real-life problem where K -NN classification would fail but a different classifier from the ones we've seen would succeed.

3 Capacity & Overfitting

Definition 1. We say that a learning system *memorizes* a training set if it can achieve 0 training error, no matter how the training examples were labeled.

Definition 2. The *capacity* of a learning system is the largest number of training point that the learning system can *memorize*, or ∞ , if there is no largest number. (Note: for capacity K it's enough to find any set of K points that the learner can memorize. This construct makes the definition robust against degenerate situations, such as multiple identical points, etc.)

a) For $\mathcal{X} = \mathbb{R}^2$, what is the *capacity* of decision trees, 1-NN, k -NN, the perceptron and Boosting? For decision trees, use binary splits along single coordinate exist with arbitrary threshold $\llbracket x_i \geq \theta \rrbracket$. For Boosting, use the same checks with output ± 1 as weak classifiers.

b) Relate their capacity and the effect of *overfitting* observed during decision tree learning.

A more intuitive (but unfortunately not very good) way to measure the capacity of a learning system would be its *number of free parameters*.

e) What's the number of free parameters for a Perceptron in \mathbb{R}^2 ?

f) What's the number of free parameters for a decision tree with binary splits and L leaves?

g) Can you find a learning system for $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \{-1, +1\}$ that has very few parameters (e.g. just 1) that can still memorize arbitrarily many points?

4 Practical Experiments I

For the rest of the course and for the final project you will need to create your own implementation of several learning methods, including

- Decision Trees*,
- k-Nearest Neighbor* (for $k \in \{1, 3, 5, 9\}$),
- Perceptron*
- AdaBoost*,
- Naive Bayes*,
- Logistic Regression*.

- For a start, pick at least two from a) to d) and implement them in a programming language of your choice.
- Apply them to the following training set:

$$\begin{aligned}x^1 &= (0, 0, 0), & y^1 &= -1, \\x^2 &= (0, 0, 0), & y^2 &= -1, \\x^3 &= (0, 1, 0), & y^3 &= +1, \\x^4 &= (0, 1, 0), & y^4 &= +1, \\x^5 &= (0, 1, 1), & y^5 &= +1, \\x^6 &= (1, 0, 1), & y^6 &= +1, \\x^7 &= (1, 0, 0), & y^7 &= +1, \\x^8 &= (1, 0, 1), & y^8 &= +1, \\x^9 &= (1, 1, 0), & y^9 &= +1.\end{aligned}$$

Plot the curves of complexity-vs-training error, using as complexity measure: a) the number of interior nodes, b) k , c) the number of passes through the dataset d) the number of boosting iterations

- Test the classifiers on the following examples

$$x^{10} = (1, 1, 1), \quad y^{10} = +1,$$

$$x^{11} = (0, 0, 1), \quad y^{11} = -1,$$

$$x^{12} = (0, 1, 0), \quad y^{12} = -1,$$

$$x^{13} = (0, 1, 1), \quad y^{13} = +1$$

and plot the complexity-vs-test error curves.

- Do the same again, but with y^9 switched from +1 to -1. How does the trained classifier change? How do the decisions change?

5 Practical Experiments II

- Download the *wine* dataset from the homepage.
 - Each row in each file is an example.
 - The first column are the labels, the other 13 columns are features.

Train one of the classifiers you programmed on the *train* part of the data, evaluate it on the *test*, and report the results.