IST Austria: Statistical Machine Learning 2020/21

Christoph Lampert <chl@ist.ac.at>
TAs: Alexandra Peste <alexandra.peste@ist.ac.at>,
Bernd Prach <bernd.prach@ist.ac.at>

Exercise Sheet 1/5 (due date 12/10/2020, 10:15am)
Please send your solutions via email to the TAs

Literature

There is no exact textbook for the material of the lecture. The introduction is most similar to the lecture notes *"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

1.1 Data

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

TRAINING:

person	`eyes`	`handsome`	`height`	`sex`	`soccer`	`date?`
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:

person	`eyes`	`handsome`	`height`	`sex`	`soccer`	`date?`
Jimbo	blue	no	tall	M	no	yes
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

1.2 Robustness

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).
a) if there had been no attribute "soccer".
b) if there had been no attribute "eye color".
c) if "Itchy" had the label `no` instead of `yes`.
d) if there had been one more training example:

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

Hint: this should not be a lot of work for you, because of lot of information can be re-used between the lecture and cases a)–d).

e) Let's call a learning method *robust*, if the classifiers it learns are similar when the training conditions change only slightly. Is decision tree learning *robust*?

f) Decision tree are often cited as classifiers which are *interpretable*, because it's easy to understand how any given decision tree comes to its decisions. Given the robustness result above, do you see a problem with this statement?

1.3 Complexity

Assume a situation in which the data consists of $n = 8$ examples with d attributes with possible values `yes` or `no`. By a technical problem, the values of all attributes get overwritten by random values ($p(\texttt{yes}) = p(\texttt{no}) = 0.5$) instead of their true values.
g) What is the probability that the decision tree construction stops with zero training error after a single split for $d = 2$, for $d = 10$, for $d = 881$ (derive this analytically or via simulation).
h) Intuitively, one might argue that providing *more data* to a learning system can only make the results *better*, since, after all, the system can simply chose to ignore any data that isn't helpful. What's your comment on this given the above observations?

2 Nearest Neighbor Classification

a) Find three examples where humans perform (more or less) nearest-neighbor classification. What about k-NN?
b) What are the advantages and disadvantages of k-NN with $k > 1$ versus 1-NN.
c) What is the error rate of 1-NN when applying it to the *training set*? Is the same true for k-NN?
d) 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 $K \geq 1$.
e) Give an example of a real-life problem where K-NN classification would likely fail but a different classifier from the ones we've seen would likely 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 a 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 with $k = 5$, and the perceptron? For decision trees, use binary splits along single coordinate exist with arbitrary threshold $[\![x_i \geq \theta]\!]$.
(Note: for this exercise, ignore how the classifiers would be trained, and only consider the parametric form of the resulting decision rule, e.g. linear functions for the Perceptron.)

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 parameters.*
e) What's the number of parameters for a Perceptron in $\mathbb{R}^2$?
f) What's the number of parameters for a decision tree with binary splits and L leafs?
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 Missing Proofs

Complete the proofs that were skipped in the lecture.

- The classifier $c^*(x) := \operatorname{argmax}_{y\in\mathcal{Y}} p(y|x)$ is identical to the Bayes classifier.
  Hint: show that $\mathcal{R}(c) \geq \mathcal{R}(c^*)$ for an arbitrary classifier, $c : \mathcal{X} \to \mathcal{Y}$.
- For $\mathcal{Y} = \{-1, +1\}$ the Bayes classifier can be written as

$$c^*(x) = \operatorname{sign}\big[\log\frac{p(x,+1)}{p(x,-1)}\big], \qquad \text{or equivalently} \qquad c^*(x) = \operatorname{sign}\big[\log\frac{p(+1|x)}{p(-1|x)}\big].$$

- For $\mathcal{Y} = \{-1, +1\}$ and $\ell(y, \bar{y}) = \begin{array}{|c|cc|}\hline y \setminus \bar{y} & -1 & +1 \\ \hline -1 & a & b \\ +1 & c & d \\ \hline \end{array}$ the classifier of minimal risk is

$$c^*_\ell(x) = \operatorname{sign}[\quad \log\frac{p(x,+1)}{p(x,-1)} + \log\frac{c-d}{b-a} \quad], \quad \text{or equivalently} \quad c^*_\ell(x) = \operatorname{sign}[\quad \log\frac{p(+1|x)}{p(-1|x)} + \log\frac{c-d}{b-a} \quad].$$

- Show: $\theta_z = \frac{1}{n}\sum_{i=1}^n [\![z^i = z]\!]$ are the maximum likelihood parameters for the multinomial model.
  Hint: you will need a Lagrangian multiplier to enforce the constraint $\sum_z \theta_z = 1$.

5 Practical Experiments I

For most common classifiers reference implementations are available in standard packages, such as `scikit-learn` (`https://scikit-learn.org/stable/`).
For the experiments marked as "practical", you can use these or your own implenentations, as you prefer.

Real world data

Download the *wine* dataset from the homepage.

- Each row in each file is an example.
- The first column are the labels (1, 2 or 3), the other 13 columns are features.

Train (on the *train* part of the data) and evaluate (on the *test* part of the data) the following classifiers from the lecture:

- Decision Tree (e.g. `sklearn.tree.DecisionTreeClassifier` if you use Python)
- k-Nearest Neighbor (e.g. `sklearn.neighbors.KNeighborsClassifier`) with $k = 1$ and $k = 5$
- AdaBoost (e.g. `sklearn.ensemble.AdaBoostClassifier`)

Please submit your code (in a language of your choice) as well as the resulting error rates.