MTAT.03.183 Data Mining

Machine Learning, Part I. Home Assignment.

Konstantin Tretyakov

Due: November 12, 2009

1. Concept of deductive and inductive reasoning. Answer the following questions. Feel free to use Google and Wikipedia, but try to come up with your own examples and formulations.

1. Bring an example (one sentence of the form "X, therefore Y") of *deductive reasoning*.
2. Bring an example of *inductive reasoning*.
3. Could you come with an example of some other conceptually different type of reasoning? Don't worry to be inventive.
4. What are the advantages and disadvantages of deductive reasoning? (At least one of each).
5. What are the advantages and disadvantages of inductive reasoning.
6. What area(s) of computer science deal with automated deductive reasoning. What for?
7. What area(s) of computer science deal with automated inductive reasoning. What for?

2. Challenges in inductive reasoning. Solve the following riddle (by T. de Bie).

Three friends are taking a sauna, having a nice beer. Andy is bragging that, on a walk through his garden this morning, he spotted a corn crake (a rare kind of bird). He even took a picture! Brad is not impressed and claims that he will be able to spot and make a photo of the same bird in his garden "any day". Charles challenges him to prove it. Next morning Brad sets out for a walk in his garden and indeed does spot a corn crake. Upon reporting this to Andy and Charles, Charles is getting jealous.

Can you advise Charles whose garden to visit? Is there a better choice?

3. Weka.

1. Download, install and run Weka (`http://www.cs.waikato.ac.nz/ml/weka/`).
2. If you haven't yet done that before, go through the introductory tutorial (`http://prdownloads.sourceforge.net/weka/Weka_a_tool_for_exploratory_data_mining.ppt`). Note: the tutorial uses the same `iris.arff` datafile that we used in the lecture. It is accessible in the `data` subdirectory of the default Weka installation.
3. Use the `trees.UserClassifier` to manually construct a tree for the classification of Iris data. How well can you do in just two splits? Attach the textual output of the resulting tree to the report.

4. Entropy. Information gain. Recollect or study the notion of entropy.

1. `http://en.wikipedia.org/wiki/Entropy_%28information_theory%29`
2. In particular, try to understand the claim of the *Source coding theorem* (no need to understand the proof).
   `http://en.wikipedia.org/wiki/Source_coding_theorem`
3. In the lecture we verified that the information gain of the split on *outlook* is 0.25. Compute and report the information gains of the other three splits.

5. Recollect probability theory. In the next lectures we shall need the notions from probability theory.

1. Make sure you are familiar with at least the following keywords. *Random variable, Probability density, Distribution, Mean, Variance, Covariance, Conditional probability, Chain rule of probability, Bernoulli distribution.*
2. Practice by solving the following classical puzzle:

   Suppose you are given a deck of three cards, consisting of one *black card*, which is black on both sides, one *white card*, which is white on both sides, and one *mixed card*, which is black on one side and white on the other. The cards are put into a hat, after which one is pulled at random and placed on the table. The side facing up is black. What is the probability that the other side is also black?

   Try to present a concise formal solution, fitting on one-two lines, starting as follows:

   $$\Pr(\text{black down}|\text{black up}) = \Pr(\text{black card}|\text{black up}) = \dots$$

6*. Machine learning case-study. OCR *(optical character recognizer)* is a system used to "read" the text off the scanned images (one of the most well known packages in this area is ABBYY FineReader). A typical workflow of an OCR system proceeds by first splitting the scanned image into images of separate letters and then recognizing each letter. Letter recognition can be regarded as a typical classification task – given a set of pixel values, classify the image into one of 26 classes, corresponding to letters `a` to `z`. Your task is to train an OCR classifier and assess its performance.

1. Download the dataset (by B. Taskar) from `http://ai.stanford.edu/~btaskar/ocr/`.

2. First you'll need to fix the input file somewhat in order to be able to load it into Weka. For that open it in Excel, add a header line with some descriptive attribute names, drop the columns "id", "nextid", "wordid", "position", and "fold", relocate the "letter" column to be the last and finally save the result in a tab-separated format.

3. Start Weka. You will probably have to give Weka more memory than what the default provides. For that, start Weka from the command line as follows:

   ```
   > java -Xmx500m -jar weka.jar
   ```

4. Import the file into Weka using the CSV loader. Mention in the parameters that the last attribute is nominal.

5. Finally, use whatever means you deem appropriate to build a classifier for the letters.

6. Measure the quality of the resulting system by whatever methods you find reasonable. Report the results briefly (not more than one paragraph of text and 5 numbers total).