

CS294 - HOMEWORK 1

The purpose of this homework is for you to get exposure to statistical tools which will be used later in the course, as well as give you a chance to work with some real data and apply basic concepts taught in lecture. The purpose is not for you to spend a lot of time creating a fancy write-up; becoming familiar with the tools and learning through experimentation about the various learning algorithms is much more valuable. Thus, the homework will be graded on a 0-2 pt scale. 2 points will be given for (almost) complete assignments, 1 point for roughly half-completed assignments.

The homework is due Tuesday, Feb 19th in class. If you have any questions, contact Alex Simma (asimma at eecs), Simon Lacoste-Julien (slacoste at eecs) or Romain Thibaux(thibaux at eecs).

1. TUTORIAL PROBLEMS

Turning these problems in is not mandatory, but it is suggested that you do them to see whether you are comfortable with the background material covered at the tutorial.

1.1. Binomial Statistics. The binomial distribution models the outcome of n independent events, where each event has a “success” with a probability p . For example, if a coin is tossed n times, m is the number of times the coin came up ‘heads’ and p is the probability of the coin toss coming up heads (0.5 for a fair coin). The distribution takes of the form $P(m \text{ out of } n) = \binom{n}{m} p^m (1 - p)^{n-m}$

1.1.1. *Deriving the MLE estimator.* Show that $\hat{p}_{MLE} = \frac{m}{n}$ is the Maximum Likelihood Estimator for p .

Hint: This is done by finding p which maximizes $\log P(m \text{ out of } n)$ and thus $P(m \text{ out of } n)$. Maximization is done by taking the derivative and setting it to zero.

1.1.2. *Multiple Samples.* Suppose you have k people. The i^{th} person tosses the coin n_i times and it comes up heads m_i times. The coin is the same for all the people and we assume there is no difference in tossing technique, etc.

We have two methods for estimating p . The first is to aggregate all the tosses, and maximize $P(\sum_i m_i \text{ out of } \sum_i n_i)$ and the other is to directly maximize the joint probability $\prod_i P(m_i \text{ out of } n_i)$. Show that both of these methods give the same estimator.

1.1.3. *Bayesian Inference.* The conjugate distribution for the binomial distribution is the Beta distribution. This is a two-parameter distribution $\text{Beta}(r, s)$ The distribution takes a fairly complex form, but the posterior after observing m successes from n trials is $\text{Beta}(r + m, s + n - m)$. The maximum a posteriori (MAP) estimate of p from a Beta distribution is $\frac{r}{r+s}$ MAP is, crudely, one version of Maximum Likelihood in the Bayesian setting.

Suppose your prior is $p \sim \text{Beta}(r_0, s_0)$. You have k people toss the coin, each getting m_i heads out of n_i tosses. Compute the posterior and show that it doesn’t depend on the way in which you order the people. Also, note how r_0 and s_0 in the prior act as made-up outcomes of made-up coin tosses. You can use this interpretation to determine how strong of a prior you should select.

2. PREREQUISITES FOR R

Parts of this homework assignment will be done using a statistical programming language called R. The software is freely available at <http://www.r-project.org/> and is available in Windows, OS X and Linux versions. Knowledge of R will be very useful for future work with statistical models, and furthermore, R has a very good SVM package which can be used on large-scale real-life problems. This section will have you install R; no deliverables are required.

2.1. Installation. You will need to install R and a package called e1071. Once R is installed, run it. So, in order to install e1071, type in

```
install.packages("e1071")
```

Note. On Unix machines, the installer will attempt to install the libraries to a global location. If you do not have permissions to do so (or do not want an unknown program to execute as root), you can set the environment variable `R_LIBS` to point to a writable directory and all package installation will occur there.

2.2. Basic Usage. R is a very powerful language which has many useful features. It is suggested that you take a look at a tutorial to get familiar with basic functionality.

In order to use a library, execute

```
library("libname")
```

3. TOY PROBLEM

See file `toy.R` for code and assignment. Perform the experiments required in the assignment and report on your findings. Submitting code is not required.

4. YALE AND SPAM FILTERING

YALE (Yet Another Learning Environment), which is also known as RapidMiner, is a Java library and GUI implementing a large number of ML functions. Download and install it from <http://rapid-i.com>. It is suggested that you go through the tutorial to familiarize yourself with its features. If you wish to experiment with the data for problem 3), load the experiment `toy-yale.xml`. Remember to turn on the expert mode in YALE.

This time, our dataset will be the real-life spam/ham (not spam) dataset used to train SpamAssassin. Originally, the training set is in the form of email messages. In order to perform classification, it is necessary to convert the email messages into feature vectors; in this case, we use a representation called the “bag of words.” That means we model a document as a collection of word frequency counts – each dimension in the feature vector is the number of times that the corresponding word appeared in the document. As part of preprocessing, the headers were removed, the HTML tags were stripped, and the 500 most relevant (you’ll learn about this later) words were left.

4.1. Basic Model. Load `spam-basic.xml` to have a basic experiment loaded. This loads the data, then cross-validates a learner. Run the model and observe the performance. The model currently used is a decision tree. Look at the accuracy statistics. There is something profoundly wrong. What’s the problem? Try using other learners (except for SVM). Report what works. Consider using preprocessing filters such as `Preprocessing->Attributes->Filter->Numeric2Binary` to transform the data before classifying it. `Numeric2Binary` replaces the word counts by indicators of whether the word was present or not. Why can it help accuracy?

4.2. Choosing Parameters. Load `spam-grid.xml` This experiment uses the SVM as a classifier, and the SVM has several parameters. All kernel types have the `C` parameter, polynomial kernels have the degree, gamma and `coef0` parameters, rbf kernels have gamma. These can be chosen by grid search, with cross-validation used to estimate performance with a certain set of parameters.

Currently, the experiment searches for the best `C` and degree for the polynomial kernel. Modify it to find the best `coef0` as well; also try changing the kernel to rbf and selecting the gamma (but don’t sweep across `coef0` and degree when using the rbf; that’ll just waste cycles).

Look at the log file to see what the accuracy is with each set of parameters. Try to understand and explain why changing the `C` has an effect on the accuracy and why having a too high `C` is not optimal (hint, higher `C` penalizes the SVM more for selecting a function which makes mistakes or comes close to making mistakes on some data).

4.3. Training, validation, testing. In 4.1 and 4.2, we have been abusing our data. Since we ran crossvalidation on the whole dataset, the resulting choice of model parameters may be overfit to the data. Explain what the problem is, since we never train on the data we test on. Why is using the same data set for selecting parameters and estimating final performance a bad idea?

If you are interested, spam-fancy.xml sets up the proper experiment. However, due to some peculiarities of YALE, it is fairly complex and it is not trivial to change learners.

5. REGRESSION

Follow the instructions in regression.R