Machine Learning 10-701 January 23, 2013
http://alex.smola.org/teaching/cmu2013-10-701/ due on February 6, 2013
Carnegie Mellon University

Homework 1

Instructions

- The homework is due in the lecture on February 6, 2013. Anything that is received after the lecture will not be considered.
- Please submit one set of notes for each of the problems and put them into a separate stack. Don't forget to add your name on each sheet.
- Alternatively, you can e-mail your solution to `10.701.homework@gmail.com`. As before, the cutoff is the end of the lecture.
- If you are on the waitlist, please either e-mail your solution or hand it in with everyone else. However, put it into an envelope and write waitlist on it. While we cannot guarantee that you will definitely get a spot, we will give preference to students who submitted homework.
- If you submit code, it should be sufficiently well documented that the TAs can understand what is happening. Also attach pseudocode if you feel that this makes the result more comprehensible.

1 Basic Statistics

This will help you get better familiarity with probability distributions.

1.1 Probabilities

Assume that $\Pr$ is a probability distribution. Denote by A and B events, and by A^c the complement of A. Prove the following:

1.1.1 $\Pr(B \cap A^c) = \Pr(B) - \Pr(A \cap B)$
1.1.2 $\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$
1.1.3 If $A \subset B$, then $\Pr(A) \leq \Pr(B)$
1.1.4 $(A \cap B)^c = A^c \cup B^c$
1.1.5 $\Pr(A \cap B) \geq \Pr(A) + \Pr(B) - 1$. This is the Bonferroni inequality. When is it useful?

1.2 Random Variables

1.2.1 Show that the following functions are Cumulative Distribution Functions (CDF):

$$F(x) = \frac{1}{2} + \frac{1}{\pi}\arctan x \qquad \text{for } x \in (-\infty, \infty) \tag{1}$$

$$F(x) = \left(1 + e^{-x}\right)^{-1} \qquad \text{for } x \in (-\infty, \infty) \tag{2}$$

$$F(x) = e^{-e^{-x}} \qquad \text{for } x \in (-\infty, \infty) \tag{3}$$

$$F(x) = 1 - e^{-x} \qquad \text{for } x \in [0, \infty) \tag{4}$$

1.2.2 Consider variables $X \sim F_X$ and $Y \sim F_Y$. Assume that F_X is *stochastically greater* than F_Y, that is

$$F_X(t) \leq F_Y(t) \text{ for all } t$$

and moreover there exists some t with $F_X(t) < F_Y(t)$. Show that X tends to be bigger than Y, that is

$$\text{for all } t \text{ we have} \qquad \Pr(X > t) \geq \Pr(Y > t) \tag{5}$$

$$\text{there exists some } t \text{ such that} \qquad \Pr(X > t) > \Pr(Y > t) \tag{6}$$

Homework 1

1.2.3 Show that, for random variables X, functions g_i and constants α_i the following holds:

$$\mathbf{E}\left[\sum_{i=1}^{n} \alpha_i g_i(x) + \alpha_0\right] = \sum_{i=1}^{n} \alpha_i \mathbf{E}\left[g_i(x)\right] + \alpha_0 \tag{7}$$

1.3 Common Distributions

1.3.1 Poisson Distribution

The Poisson distribution is used to model a flow of events given only the average rate at which an event occurs. Examples of its use include the following:

- The number of incoming requests to a server.
- The number of mutations on a strand of DNA.
- The number of cars arriving at a traffic light.
- The number of raindrops in a given area in a given time interval.
- The number of Prussian soldiers who died from horse kicks.[1]

The Poisson distribution has one parameter, the average rate $\lambda > 0$ and has probability mass function

$$p(k;\lambda) = \Pr(X = x) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{8}$$

- Compute mean and variance of the Poisson distribution (hint — Taylor expansion for e^λ).
- With $x_1 \dots x_n$ drawn iid from $\text{Poisson}(\lambda)$, derive a Maximum Likelihood estimator for λ.
- Let $X_1 \sim \text{Poisson}(\lambda_1)$ and $X_2 \sim \text{Poisson}(\lambda_2)$ be independent random variables. Show that the random variable $Z = X_1 + X_2$ is Poisson-distributed and compute its mean.

1.3.2 Cauchy Distribution

The Cauchy distribution is given by

$$p(x;\gamma) = \frac{\gamma}{\pi(x^2 + \gamma^2)} \tag{9}$$

It was established in class that the mean of a Cauchy distribution does not exist. Show that its second moment is infinite — therefore the variance is not defined either.

1.3.3 Mean and Variance

Let $X_1, \dots, X_n$ be iid random variables with mean μ and variance σ^2.

(a) Prove that

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \text{ and } S_n = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

are unbiased estimators of the mean and variance, that is, $E[\bar{X}] = \mu$ and $E[S_n] = \sigma^2$.

(b) For a distribution of your choice (Gauss, binary, uniform, etc.) implement these estimators and empirically show on a plot that these estimators indeed converge to the true quantities μ and σ^2 as we increase the sample size n. Based on your plots guess the convergence rates of the estimators.

[1] http://books.google.com/books?id=o_k3AAAAMAAJ&pg=PA23

Homework 1

1.4 Conditional Independence

The notation $X \perp Y$ indicates that random variables X and Y are independent. Similarly, $X \perp Y|Z$ means that X and Y are conditionally independent given Z, that is $p(x, y|z) = p(x|z)p(y|z)$.

1.4.1 Prove (or disprove via a counterexample) the following statements about conditional independence

$$\begin{aligned} (X \perp (Y, W)|Z) \text{ implies } & (X \perp Y|Z) \\ (X \perp Y|Z) \text{ and } (X, Y \perp W|Z) \text{ implies } & (X \perp W|Z) \\ (X \perp (Y, W)|Z) \text{ and } (Y \perp W|Z) \text{ implies } & (X, W \perp Y|Z) \end{aligned}$$

1.4.2 Give examples of distributions over random variables X, Y, Z that satisfy

$$\begin{aligned} X \not\perp Y \text{ but } X \perp Y|Z \\ X \perp Y \text{ but } X \not\perp Y|Z \\ X \perp Y \text{ and } X \perp Z \text{ and } Y \perp Z \text{ but } (X, Y) \not\perp Z \end{aligned}$$

2 Linear Regression

Suppose that $X \in \mathbb{R}^d$ is a random variable with $X \sim D$. Moreover, assume that $b \in \mathbb{R}$ and $w \in \mathbb{R}^d$ are parameters and let

$$Y = \langle X, w \rangle + b + \epsilon \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2). \tag{10}$$

That is, Y is a linear function of X with some Gaussian noise added.

2.1 Maximum Likelihood Estimation

2.1.1 Give an explicit expression for the conditional distribution of $Y|X, w, b$.
2.1.2 Assume that we draw n pairs (x_i, y_i) from the above model, i.e. we draw x_i from D and the y_i according to (10). What is the distribution of $(y_1, \ldots, y_n)|(x_1, \ldots, x_n), w, b$? Write out the associated the log-likelihood.
2.1.3 Derive the parameters w, b that maximize the log-likelihood of the conditional model.

2.2 Maximum-a-Posteriori Estimation

Now assume that the parameters w, b are drawn from a Normal distribution with $w \sim \mathcal{N}(0, \lambda^2 \mathbf{1})$ and $b \sim \mathcal{N}(0, \lambda^2)$.

2.2.1 Give an explicit expression for the conditional distribution of $Y|X$. Note that w, b are not specified.
2.2.2 After drawing n pairs as above, what is the distribution of $(y_1, \ldots, y_n)|(x_1, \ldots, x_n)$?.
2.2.3 What is the log-posterior $w, b|((x_1, y_1), \ldots, (x_n, y_n))$.
2.2.4 Find the mode of it with respect to w, b.
2.2.5 Do we need to know w, b in order to estimate $y_{n+1}|(x_1, \ldots, x_n), (y_1, \ldots, y_n), x_{n+1}$? Describe an alternative.

2.3 Sparsity Penalty

Now assume that instead of a log-prior on w, b we use an ℓ_1 penalty ($\|z\|_1 := \sum_i |z_i|$ is the 1-norm of z). That is, we minimize

$$-\log p(y_1, \ldots, y_n | x_1, \ldots, x_n, w, b) + \lambda \left[\|w\|_1 + |b|\right]. \tag{11}$$

Give a sufficient condition under which (11) has a closed form solution in terms of w, b (hint — consider large λ). Why does it not have such a solution in general.

3 Naive Bayes and k-Nearest Neighbors

In this problem you will implement Naive Bayes and k-NN and test them on both synthetic data and real data. For the synthetic dataset you can download the code generator here[2]. You can check out what it does by running this script[3]. Moreover, choose one dataset of your choice from the UCI repository[4] that can be viewed as a *binary* classification problem and that contains at least 2,000 instances and at least 10 dimensions.

3.1 Data Preprocessing

Download the dataset. Describe the transformations (covariates x, labels y) you applied to convert it into a binary classification problem. Attach the preprocessing code.

3.2 Naive Bayes

Implement a Naive Bayes algorithm. Apply binning for the discrete components and use a Gaussian model for the continuous components of the data (e.g. the synthetic dataset has only continuous components). Attach the code. It should have the following signature:[5]

```
model = nb_train(Xtrain, Ytrain, attributes)
Ytest     = nb_test(model, Xtest)
```

Here `X`, `Y` are sets of observations and labels respectively. `attributes` is a vector of attribute indicators, i.e. whether they are discrete or continuous respectively. `model` is a container for all the model parameters you generate.

3.3 k-Nearest Neighbor

Implement a k-Nearest Neighbor algorithm. Discuss your choice of distance function for discrete random variables. Attach the code. The call signature should take the following form:

```
Ytest = knn(Xtrain, Ytrain, Xtest, attributes, k)
```

The meaning of the variables is as above. `k` obviously denotes the k in k-Nearest Neighbors. It is OK to use a memory-inefficient implementation. Moreover, it is OK for the purpose of this assignment to use Quicksort (i.e. Matlab's sort function) to find the nearest neighbor.

Why should you use odd k?

[2] http://alex.smola.org/teaching/cmu2013-10-701/assignments/gen_synthetic.m
[3] http://alex.smola.org/teaching/cmu2013-10-701/assignments/draw_synthetic.m
[4] http://archive.ics.uci.edu/ml/datasets.html
[5] Feel free to use any other language than MATLAB/Octave, as long as you are sure that the TAs are able to read it. For C/C++ the eigen and flens libraries are quite useful.

3.4 Comparison on synthetic data

Your task is to compare k-NN and Naive Bayes on the synthetic datasets. You need to implement N-fold crossvalidation for both codes. We set $N = 10$ in the experiments below. Keep track of training and validation error. The function `gen_synthetic.m` generates (as expected) synthetic data. Its signature is as follows:

```
function [Y,X,w] = gen_synthetic(n,p,rho,sigma)
```

Here `n` controls the number of samples, `p` controls the dimensionality, `rho` measures correlation between dimensions and `sigma` determines the amount of additive noise in the associated linear model. Moreover, `X`, `Y` correspond to data and labels respectively. Finally, `w` is the parameter vector used by the model to generate the data.

3.4.1 Generate a set of points with `n=1000`, `p=10`, `rho=0` and `sigma=0.001`. For the values of `k=1:2:49` run the k-NN algorithm and plot the change of the mean cross-validation test and train error (submit the plot). What is a value of `k` after which the test error becomes rather stable? What can you say about the shape of the training error for small values of `k`?

3.4.2 Generate a set of points with `n=1000`, `p=4`, and `sigma=0.001`, but now for `rho=0.1:0.1:0.8`. Run Naive Bayes and k-NN with `k=9`. Plot the change in cross-validation test error as a function of `rho`. How does the performance of these two classifiers compare when the variables become more dependent? Explain the differences between the performance of the two algorithms.

3.4.3 Generate a set of points with `n=400`, `rho=0.1`, and `sigma=0.001`, but now for `p = 2:2:40`. Run both classifiers (with `k=9` and plot the change in cross-validation test error as a function of the dimensionality `p`. How does the performance of these two classifiers compare as the dimensionality of the data increases? Explain the differences between the performance of the two algorithms.

3.5 Comparison on real data

Compare k-NN and Naive Bayes on several fractions of your data. For this purpose, pick 1%, 2%, 5%, 10%, 20% and 100% of the data and compare the performance of naive Bayes and k-NN respectively (for `k=1` and `k=9`). For the full dataset, plot the k-NN crossvalidation error for `k=1:2:49`.