MACHINE LEARNING COMS 4771, HOMEWORK 3

Assigned October 15, 2014. Due November 6, 2014 before 1:00pm.

1 Problem 1 (10 points)

In this problem we consider tossing a coin with two sides labeled H and T. Assume all tosses are iid with $p(H) = \mu$.

1.1 Discrete possible parameter values

There are 3 possible values for μ. It could be fair $\mu = \frac{1}{2}$, biased tails $\mu = \frac{1}{4}$, or biased heads $\mu = \frac{3}{4}$. Assume our prior is that each of these possibilities is equally likely.

Looking forward, what is the minimum number of tosses we'd need to see in order to conclude that $p(\mu = \frac{1}{2}) > \frac{1}{2}$ (strictly greater)? Give an example of a sequence of tosses with this length which would lead to this conclusion. Prove your results.

Instead, assume no tosses have yet occurred. What is the minimum number of tosses we'd need to see in order to conclude that $p(\mu = \frac{3}{4}) > \frac{1}{2}$ (strictly greater)? Give an example of a sequence of tosses with this length which would lead to this conclusion. Prove your results.

1.2 Continuous possible parameter values

You may find results on the Beta distribution given in lecture 2 useful and may quote them without proof.

Consider two possible prior distributions: (A) $\mu \sim \text{Uniform}[0, 1]$; (B) we have some reason to think the coin is likely to be fair so assume μ has a probability distribution which has the form of a concave parabola with its maximum at $\frac{1}{2}$ and falls to 0 at 0 and 1.

Along with 2 possible realizations: (1) $\mathcal{D}_1 = \{H, T\}$; (2) $\mathcal{D}_2 = \{T, T, T\}$.

For each of the 4 combinations of priors and realizations, derive with proof:

- $p(H)$ given the prior
- the posterior distribution $p(\mu|\mathcal{D})$ [ensure this is properly normalized]
- the maximum likelihood estimate μ_{ML} given the data $\mathcal{D}$
- the MAP estimate μ_{MAP} given the data $\mathcal{D}$
- $p(H|\mathcal{D})$, i.e. the full Bayesian probability that the next toss is H
- the variance of the posterior distribution $p(\mu|\mathcal{D})$

Give one reason why the maximum likelihood estimate might not be good in this context.

2 Problem 2 (5 points): Bayes Rule

Assume we have a good test for swine flu that is highly accurate, and outputs either true(T) or false(F). If a patient has swine flu and takes the test, the test will output true(T)99% of the time. If a patient does not have swine flu, the test will output false (F)98% of the time.

Assume that we know that 1 in 10,000 people have swine flu.

If Joe takes the test and it outputs T, what is the probability that Joe has swine flu?

3 Problem 3 (5 points): Conditional Independence

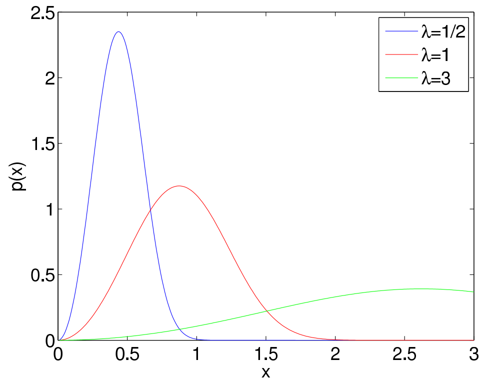
Prove or disprove with a counterexample:

1. For random variables u, v, w and $x : u \perp (v, w)|x \rightarrow u \perp v|x$.

2. For single events A, B and $C : A \perp (B, C) \rightarrow A \perp B$.

4 Problem 4 (10 points): Maximum Likelihood

The Weibull distribution is a probability density over non-negative scalars, $x \geq 0$. One particular form is as follows $p(x|\lambda) = \frac{3}{\lambda}(\frac{x}{\lambda})^2 e^{-(\frac{x}{\lambda})^3}$. The density has a scalar parameter $\lambda > 0$ which adjusts its shape as shown below.



What is the maximum likelihood estimate of λ from a dataset $x_1, \cdots, x_n$ of n iid samples?

5 Problem 5 (20 points): Kernel Logistic Regression

Load dataset 'data1.mat'. Implement kernel logistic regression with an ℓ^2 regularizer using the empirical kernel map. In other words, minimize:

$$J(w) = -\sum_{i=1}^{N} \log(\sigma(y_i w^\top k_i)) + \lambda w^\top w$$

to get w. Here k_i is a column vector such that $k_i = [k(x_i, x_1), \cdots, k(x_i, x_j), \cdots, k(x_i, x_N)]^\top$. Here $y_i \in \{-1, +1\}$ is the given label for each input $x_i \in \mathbb{R}^d$. Note that function σ is defined as $\sigma(v) = 1/(1+e^{-v})$. Use the radial basis function (RBF or Gaussian) kernel $\exp(-\|x_i - x_j\|^2/\kappa^2)$, where the parameter κ is used as the mean of the pairwise distance, i.e., $\kappa^2 = \frac{1}{N^2}\sum_{i,j=1}^{N} \|x_i - x_j\|^2$. After w is obtained, for any test input x, compute $p(y = 1|x) = \sigma(w^\top k_x)$, where $k_x = [k(x, x_1), \cdots, k(x, x_j), \cdots, k(x, x_N)]^\top$. If $p(y = 1|x) > 0.5$, the predicted label $y = 1$, otherwise, $y = -1$. Report the accuracy: the percentage of test data where your predicted label agrees with the given ground truth label. To optimize $J(w)$, use two methods: gradient descent and stochastic gradient descent. For stochastic gradient descent, use p points to estimate the gradient. Experiment with $p = 1$ and $p = 100$. Experiment with the step size and choose the one that works for you. Compare how the value of the cost function decreases with time for different methods. Stop the iterations when the gradient becomes smaller than epsilon (say, $1e - 5$).