10701/15781 Machine Learning, Spring 2005: Homework 1

Due: Monday, February 6, beginning of the class

1 [15 Points] Probability and Regression [Stano][1]

1.1 [10 Points] The Matrix Strikes Back

The Matrix was upgraded to a new, probabilistic version.

After the peace had been reached, things were not as perfect as a lot of people and machines imagined. A large class of people got tired of living a real life and wanted to go back into the simulation. Sentinels got tired of computing chess strategies and wanted a real job again. And, of course, there were those who wanted freedom at all costs, for everyone. In the end, human governments, in cooperation with *Deus ex machina*, devised a scheme, in which people would be automatically classified at childbirth into those allowed in the Matrix and those to be left free. While this rule allowed many people live the lives they wanted (or imagined), it did not meet with comprehension of those who were incorrectly classified and wanted to switch (not to mention those who wanted to switch just to be different from those around them). Thus, despite numerous attempts to improve the classification scheme, the war arose again.

After several unsuccessful trials, a random Neo was generated to save the world and, once again, bring peace to the machines and the humanity. To test Neo's worthiness, Morpheus decided to give him a new task: rather than simply choosing between a red and blue pill, Neo needs to *draw* a pill from one of three identically looking bags with different proportions of red and blue pills. At the beginning, one bag has 7 red and 3 blue pills, another bag has 6 red and 4 blue pills, while the third bag has 5 red and 5 blue pills. Being only at the beginning of his journey, Neo does not know which bag is which and does not have the ability to see through a bag to pick a red pill with certainty. Instead, he must gamble to pick a bag that maximizes his chances of drawing a red pill. In an attempt to help Neo, Morpheus picks a bag uniformly at random and draws a pill from it. It is a red pill, and Morpheus gets to keep it. Neo carefully observes which bag Morpheus picks from and ponders the question "What is the Matrix?", rather than a much more useful "What is a Distribution?" Now it is his turn. Which bag should Neo draw from and what is the probability of drawing a red pill from this bag? Did Morpheus really help Neo? Show your work.

1.2 [5 Points] Regularization

Regression is often prone to overfitting when the feature space is rich. Regularization is a typical strategy to prevent overfitting. Suppose that we choose the feature space to be the polynomials of degree 5, i.e.

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{j=1}^{N} (y_j - \sum_{i=0}^{5} (x_j)^i w_i)^2 + \lambda \sum_{i=0}^{5} w_i^2, \tag{1}$$

where λ is the regularization parameter. Consider the dataset shown in Figure 1. What is the solution (qualitatively) with $\lambda = 0$? What is the solution as $\lambda \to \infty$? What is the solution for a "good" choice of λ. Justify your answer.

[1] Each problem is marked with the name of TA who is responsible for it. If you have any questions about the problem, this TA is your first point of contact. But you can also ask any of us about any problem.

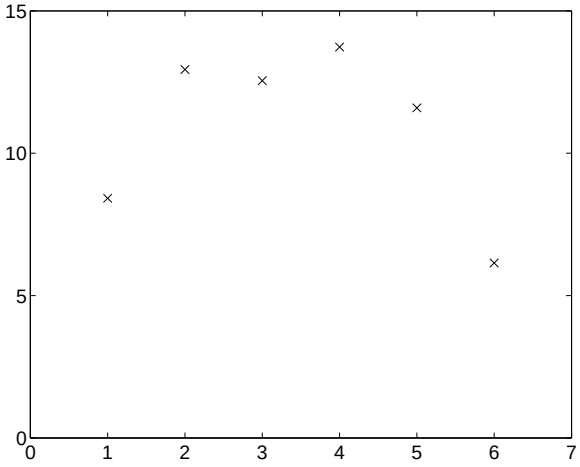

Figure 1: Dataset for Question 1.2

2 [40 Points] Discriminative vs. Generative Classifiers [Jure]

Over the last few years, there has been a growing debate over generative versus discriminative models for classification. This question will explore this issue, both theoretically and practically. We will consider naive Bayes and logistic regression classification algorithms.

To answer this question, you might want to read: *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes*, Andrew Y. Ng and Michael Jordan. In NIPS 14, 2002. `http://www.robotics.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf`

2.1 [15 points] Double Counting the Evidence

1. Show that for both Naive-Bayes and logistic regression, the classification decision rule can be expressed as choosing the class that maximizes a linear sum, each of whose terms refers to at most one of the attributes X_i of X.

2. What function does each approach optimize?

3. Consider the two class problem where class label $y \in \{T, F\}$ and each training example X has 2 binary attributes $X_1, X_2 \in \{T, F\}$.

   Let the class prior be $P(Y = T) = 0.5$ and also let $P(X_1 = T|Y = T) = 0.8$ and $P(X_2 = T|Y = T) = 0.5$, and, similarly for the negative class, $P(X_1 = F|Y = F) = 0.7$ and $P(X_2 = F|Y = F) = 0.9$. So, attribute X_1 provides a slightly stronger evidence about the class label than X_2.

   (a) Assume X_1 and X_2 are truly independent given Y. Write down the Naive Bayes decision rule.

   (b) What is the expected error rate of Naive Bayes if, it uses only the attribute X_1? What if it uses only X_2?

   The expected error rate is the probability that each class generates an observation where the decision rule is incorrect: if Y is the true label, let $\tilde{Y}(X_1, X_2)$ be the result of classification (predicted class label), then the expected error rate is

   $$P(X_1, X_2, Y|Y \neq \tilde{Y}(X_1, X_2))$$

   (c) Show that if Naive Bayes uses both attributes, X_1 and X_2, the error rate is 0.235, which is better than if using only a single attribute (X_1 or X_2).

(d) Now, suppose that we create new attribute X_3, which is an exact copy of X_2. So, for every training example, attributes X_2 and X_3 have the same value, $X_2 = X_3$. What is the expected error of Naive Bayes now?

(e) Explain what is happening with Naive Bayes? Does Logistic Regression suffer from the same problem? Explain why?

2.2 [25 points] Learning Curves of Naive Bayes and Logistic Regression

Compare the two approaches on the Chess dataset you can download from course webpage. Obtain the learning curves similar to Figure 1 in the paper.

Implement a Naive Bayes classifier and a logistic regression classifier with the assumption that each attribute value for a particular record is independently generated.

For the NB classifier, assume that $P(x_i|y)$, where x_i is a feature in the chess data (that is, i is the number of column in the data file) and y is the label, of the following multinomial distribution form:

for $x_i \in \{v_1, v_2, ..., v_n\}$,

$$p(x_i = v_k|y = j) = \theta^i_{jk} \ s.t. \ \forall i, j : \sum_{k=1}^{n} \theta^i_{jk} = 1$$

where $0 \leq \theta_{jk} \leq 1$ and $I(z) = 1$ iff the condition z is true (else $I(z) = 0$). It may be easier to think of this as a normalized histogram or as a multi-value extension of the Bernoulli.

Use 2/3 of the examples for training and the remaining 1/3 for testing. Be sure to use 2/3 of each class, not just the first 2/3 of data points.

For each algorithm:

1. Briefly describe how you implement it by giving the pseudocode. The pseudocode must include equations for estimating the classification parameters and for classifying a new example. Remember, this should not be a printout of your code, but a high-level outline.

   You should submit the code itself electronically to

   `/afs/andrew.cmu.edu/course/10/701/Submit/your_andrew_id/HW1/`

2. Plot a learning curve: the accuracy vs. the size of the training data. Generate six points on the curve, using 1/6, 2/6, ..., 6/6 of your training set and testing on the full test set each time. Average your results over 10 random splits of the data into a training and test set (always keep 2/3 of the data for training and 1/3 for testing, but randomize over which points go to training set and which to testing). This averaging will make your results less dependent on the order of records in the file. Plot both the Naive Bayes and Logistic Regression, learning curves on the same plot.

   Specify your choice of prior/regularization parameters and keep those parameters constant for these tests. A typical choice of constants would be to add 1 to each bin before normalizing (for NB) and $\lambda = 0$ (for LR).

3. Plot the accuracy using the full training set as you vary the prior/regularization parameters. Generate one plot with priors of varying strengths for the naive-Bayes classifier. Also create one plot for the logistic regression classifier, varying the λ regularization parameter for regression.

   The MATLAB functions `fminsearch` and `glmfit` may be useful for this part. Try to choose a range of parameters such that a peak is visible in the curve.

4. What conclusions can you draw from your experiments? Specifically, what can you say about speed of convergence of the classifiers? Do Naive Bayes assumptions hold for this dataset?

3 [30] Decision Trees and Overfitting [Andreas]

After having explained Maximum Likelihood Estimation to the friendly billionaire, he is impressed and invites you to a little gamble. He flips his thumbtack X, and you have to tell him whether it has landed upside down ($X = -1$) or heads up ($X = +1$). *Hint: For answering the later parts of this problem, the functions* `normcdf` *and* `norminv` *in the Matlab Statistics Toolbox might come in handy.*

1. [2 points] From your earlier experiments, you know that the thumbtack lands heads up with probability $\theta = 4/5$. You get one guess; if you guess correctly, the billionaire gives you one million, if you are wrong, you get nothing. Based on your earlier estimate, what should you predict? How much do you win in expectation?

2. [5 points] The billionaire is even more generous – you do not even have to guess! Of course he does not let you see his throw, but he lets his parrot see it, who then croaks the answer into a microphone. A speech recognition software then outputs a real number Y which you observe. If the result of the thumbtack throw is $x \in \{-1, +1\}$, then Y is normally distributed with mean x and variance 1. For example, if the thumbtack lands upside down, $Y \sim \mathcal{N}(-1, 1)$, i.e. Y is distributed according to a Gaussian with mean -1 and variance 1.

   Knowing that the thumbtack lands heads up with probability 4/5, which decision rule of the form "Predict $X = 1$ if $Y \geq q$, and $X = -1$ otherwise" should you choose? Show that setting $q = -\frac{1}{2} \log \frac{\theta}{1-\theta} \approx -.69$ maximizes your chance of winning. How much do you win in expectation? *Hint: For fixed q, express the probability of winning using the cumulative distribution function of a normal distribution.*

3. [3 points] Realizing that you are winning too much, the billionaire makes the gamble a little harder for you. Now the parrot is sometimes sitting upright on a branch, and sometimes hanging upside down on it. Hence the parrot might misinterpret the flip of the thumbtack. Before the gamble starts, the parrot chooses a position of its liking, and will subsequently remain in this position.

   More formally, your goal is again to predict the outcome X of the thumbtack toss (+1 or -1). You will again see the interpretation Y of the parrot's assessment by the speech recognizer: However, unlike in part b), you only know that *either* $Y \mid X = x \sim \mathcal{N}(x, 1)$ (if the parrot is sitting heads up) or $Y \mid X = x \sim \mathcal{N}(-x, 1)$ (if the parrot is hanging upside down). Hence Y is either a noisy copy of X or of $-X$, with standard normal noise.

   Since you do not have any prior information about parrots, you do not know which of the hypothesis (parrot is sitting or hanging) is true. Does the prior information $\theta = 4/5$ help you in interpreting the parrot's croaking? What is your best bet for winning, and how much do you win in expectation?

4. [5 points] Realizing that he is being unfair, the billionaire allows you to observe n training examples (x_i, y_i), $1 \leq i \leq n$ before you make your classifications. After you have seen your training examples, you can either choose to

   (a) Always predict +1 (ignore the parrot) or

   (b) Use the decision rule from part b) if $\sum_i x_i y_i \geq 0$, and the inverted rule (predicting -1 instead of +1 and vice versa) if $\sum_i x_i y_i < 0$. Convince yourself that this decision rule makes sense, by observing that the products $x_i y_i$ are distributed according to a Gaussian with mean 1 if the parrot is sitting and mean -1 if the parrot is hanging.

   Option (a) corresponds to an empty decision tree, resulting in a baseline classifier which has a test set error of 1/5. Option (b) corresponds to a decision tree with one node (one split at $Y \geq q$).

   Assuming you get to see only a single training example ($n = 1$), what is the chance that you learn the correct decision rule for Option (b)? *Hint: You will need to compute tail bounds of a normal distribution.*

5. [5 points] How many training examples do you need such that you will learn the correct classification rule with probability at least 0.95? How many for probability 0.99? Express your answer in terms of the inverse c.d.f. Φ^{-1} of the standard normal distribution:

$$\Phi^{-1}(a) = z \quad s.t. \quad a = P(Z \leq z) \triangleq \Phi(z),$$

where $Z \sim \mathcal{N}(0, 1)$.

6. [5 points] How many training examples do you need such that Option (b) is preferable over Option (a)? Please give a general formula.

7. [5 points] What do you learn from this analysis about overfitting of decision trees?

4 [15 points] Decision Trees vs. Linear Classifiers [Anton]

Your friendly billionaire is going to hire a new team to work on a new search engine called Shmoogle. He has a bunch of applicants and wants to classify these people into good programmers (worth hiring) and bad programmers based on their GPA and top score in Minesweeper. In order to do that, he wants to use a classifier, but cannot choose which one, so he needs your help.

4.1 [3 points] Single-node Decision Tree As Linear Classifier

When the features are continuous, a decision tree with one node (a depth 1 decision tree) can be viewed as a linear classifier (such degenerate trees, consisting of only one node and therefore using only one variable to split the data, are also called *decision stumps*). What is the difference between this classifier and other linear classifiers you have learned in class, such as logistic regression (LR)?

Provide a small dataset (no more than 10 2-dimensional points), for which a perfect classifier can be trained using LR, but no one-node decision tree is a perfect classifier. Qualitatively show the decision boundary of the classifier learned using LR and one-node decision tree with ID3 (assume that ID3 stops after generating the root node). Justify your answer.

4.2 [5 points] General Decision Trees vs. Linear Classifiers

If we allow decision trees to have depth more than 1, they can capture more complex separation surfaces. Provide a small 2D dataset for which there exists a decision tree that is a perfect classifier, but no perfect linear classifier exists. Show (again qualitatively) the resulting decision tree learned by ID3 without pruning, and linear classifier learned by LR.

4.3 [7 points] Using Linear Classifiers In Decision Trees

Now that we saw that there are cases in which decision trees are preferable, and there are cases in which linear classifiers are preferable, let us combine their advantages. Your goal is to sketch an algorithm that would train a linear-classifier-augmented decision tree: at each node, instead of testing some particular feature to split the decision tree, we now use the output of some linear classifier. Positively classified examples will go to one branch of the tree, while negatively classified ones will follow the other branch.

Assume that you have a "black box" that takes a training set as input and gives you an optimal (but not necessarily perfect!) linear classifier as the output. Do not worry about the stopping criteria - grow the tree until the classifier is perfect for the training set.

Provide a dataset, for which no perfect linear classifier exist, and your algorithm outputs a decision tree with depth smaller than the depth of an ID3 decision tree (again assume that no pruning takes place for ID3). Qualitatively show both resulting trees, and justify your answer.