CMPS 142 Third Homework, Winter 2016

4 Problems, 16 pts, due start of class Wednesday 2/17

This homework is to be done in groups of 2 or 3. Each group members should completely understand the group's solutions and *must* acknowledge all sources of inspiration, techniques, and/or helpful ideas (web, people, books, etc.) other than the instructor, TA, and class text. Each group should submit a single set of solutions containing the names and e-mail addresses of all group members. Although there are no points for "neatness", the TA may deduct points for illegible or poorly organized solutions.

1. (2 pts) Suppose that we have the following training set (where the instances have two features):

x_1	x_2	label
1	1	+1
1	2	+1
2	1	+1
0	0	-1
1	0	-1
0	1	-1

   Plot them and find the support vectors (by eye). Using the support vectors, find the equation for the maximum margin separating plane, and determine the geometric margin.

2. (5 pts) Naive Bayes.

   Consider using Naive Bayes to estimate if a student will be an honor student (**H**) or normal student (**N**) in college based on their high school performance. Each instance has two measurements: the student's high school GPA (a real number) and whether or not the student took any AP courses (a boolean value, yes=1, no=0). Based on the following training data, create (by hand and/or calculator) a Naive Bayes prediction rule using gaussians to estimate the conditional probability density of a high school GPAs given the class (**H** or **N**) and a Bernoulli distribution for the AP probability . (I know that Gaussians may not fit this problem well, but use them anyway).

   Recall that Naive Bayes makes the simplifying assumption that the features are conditionally independent given the class (Although in the Naive Bayes chapter Andrew Ng emphasizes fitting discrete distributions to the features, one can also fit a continuous density to the features an use the density at a feature value just like the probability under a discrete distribution), so (for example)

$$\mathbb{P}\left[\text{GPA=3.2, AP=yes} \mid \text{type=H}\right] = \mathbb{P}\left[\text{GPA=3.2} \mid \text{type=H}\right] \mathbb{P}\left[\text{AP=yes} \mid \text{type=H}\right].$$

label	AP	GPA
H	yes	4.0
H	yes	3.7
H	no	2.5
N	no	3.8
N	yes	3.3
N	yes	3.0
N	no	3.0
N	no	2.7
N	no	2.2

Use maximum likelihood estimation (*not* the unbiased or Laplace estimates) for the distributions of the two features conditioned on the two classes. Give the mean and variance of the gaussians you found for the GPA.

Describe your prediction rule in the following form:

> If AP courses are taken, predict **H** if the GPA is between ..., and
> if AP courses are not taken, predict **H** if the GPA is between ...

(It is probably easier to get this description if you take logarithms, 3 digits of precision should sufficient. Also, the logarithm of the Gaussian densities are quadratic, so it is possible that two different GPA values v could both have

$$\mathbb{P}\left[\text{GPA}{=}v, \text{AP}{=}\text{yes} \mid \text{type}{=}\text{H}\right] = \mathbb{P}\left[\text{GPA}{=}v, \text{AP}{=}\text{yes} \mid \text{type}{=}\text{N}\right].$$

so the prediction rules can be finite intervals of GPA values rather than a simple threshold.)

3. (5 pts) Gaussian discriminant analysis.

   Consider using Gaussian Discriminant Analysis to learn a two-class problem where the labels are either 0 or 1 and the instances (feature vectors) are vectors of n real numbers. The learned model will be of the form:

$$\begin{aligned} p(y) &= \phi^y (1-\phi)^{1-y} \\ p(\mathbf{x} \mid y=0) &= \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_0)^\mathsf{T}\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_0)\right) \\ p(\mathbf{x} \mid y=1) &= \frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_1)^\mathsf{T}\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_1)\right) \end{aligned}$$

   Note that although μ_0 and μ_1 are different, the covariance matrix $\mathbf{\Sigma}$ is shared between the two classes.

Recall that $\mathbf{\Sigma}$ and $\mathbf{\Sigma}^{-1}$ are symmetric, and so equal their transposes. Also, since $\mathbf{v}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{w}$ is a scalar, it is equal to its transpose. Therefore:

$$\mathbf{v}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{w} = (\mathbf{v}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{w})^\mathsf{T} = \mathbf{w}^\mathsf{T}(\mathbf{v}^\mathsf{T}\mathbf{\Sigma}^{-1})^\mathsf{T} = \mathbf{w}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{v}$$

First show that the ratio $p(\mathbf{x} \mid y = 1)/p(\mathbf{x} \mid y = 0)$ can be written as $\exp(\theta^\mathsf{T}\mathbf{x} + c)$ for vector θ and scalar c that are functions of $\mathbf{\Sigma}^{-1}$, μ_0, and μ_1.

Then show that, like logistic regression, Gaussian Discriminant Analysis's model has the relationship

$$p(y = 1 \mid \mathbf{x})/p(y = 0 \mid \mathbf{x}) = \frac{1}{1 + \exp(\mathbf{w}^\mathsf{T}\tilde{\mathbf{x}})}$$

for some $\mathbf{w}$ (here $\tilde{\mathbf{x}}$ is $\mathbf{x}$ augmented by the add-a-dimension trick).

4. (4 pts) Support Vector Machine Kernels

We'll use LibSVM to explore the effects of kernels on SVM performance. You can Install LibSVM through the Weka package manager, after which it should be available in `functions` → `LibSVM`. You may use LibSVM another way if you prefer.

There is a guide to using SVMs for classification at `http://www.csie.ntu.edu.tw/ cjlin/papers/guide/guide.pdf`. Begin by reading this guide. They recommend beginners try the following first:

(a) Transform data to the format of an SVM package

(b) Conduct simple scaling on the data

(c) Consider the RBF kernel $K(\mathbf{x}; \mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|^2}$.

(d) Use cross-validation to find the best parameter C and γ.

(e) Use the best parameter C and γ to train the whole training set.

(f) Test

In this exercise, we'll concentrate on steps (a) through (d) working with the UCI dataset spambase. This dataset can be obtained from `https://archive.ics.uci.edu/ml/datasets/Spambase`. Open the file spambase.arff (there are several versions on the web, we will put one in the resources shortly). You may have to run the filter supervised → attribute → NominalToBinary to change the categorical attributes to numeric (binary) attributes. You will also want to normalize the data so that each feature lies in a small range..

Use cross validation (if 10-fold cross validation takes too long, use fewer folds) to find good values for the C and γ parameters. The C (classifier.cost) parameter controls the penalty for failing to achieve the correct margin: a higher value of C increases the penalty and encourages the algorithm to separate all of the data. A lower value of C

makes the algorithm more willing to give up on a few examples in order to better fit the bulk of the data. Try C values ranging from around 1/10 to 1000.

The γ (classifier.gamma) parameter controls the width of the gaussian. Try γ values between 1/100 to 10. Report the number of folds you use, the best C and γ pair, and the average cross validation accuracy you get.

Compare against a polynomial kernel of degree 2 in LibSVM. The kernal used here is $K(\mathbf{x}, \mathbf{x}') = (\gamma \mathbf{x} \cdot \mathbf{x}' + \text{coef0})^2$. Make a small attempt to optimize the C, γ, and "coef0" parameters. How does its accuracy compare with the RBF kernel?

Compare against the default Linear SVM, which uses a linear kernel (just the standard dot product. Make an attempt to approximately optimize the C (cost) parameter. How did it compare?