CS340 Winter 2010: HW4
Out Wednesday 2nd March, due Wednesday 16th March

1 Naive Bayes Classifier Implementation

The Nursery database records a series of admission decisions to a nursery in Ljubljana, Slovenia. These data were downloaded from http://archive.ics.uci.edu/ml/datasets/Nursery. The database contains one tuple for each admission decision. The features or attributes include financial status of the parents, the number of other children in the house, etc. See http://archive.ics.uci.edu/ml/machine-learning-databases/nursery/nursery.names for more information about these features. The first three tuples in the dataset are as follows:

```
usual,proper,complete,1,convenient,convenient,nonprob,recommended,recommend
usual,proper,complete,1,convenient,convenient,nonprob,priority,priority
usual,proper,complete,1,convenient,convenient,nonprob,not_recom,not_recom
```

where the first 8 values are features or attributes and the 9th value is the class assigned (i.e., the admission decision recommendation).

Your job is to build a Naive Bayes classifier that will make admission recommendations. The data have been prepared for you by Marcos and can be downloaded from the course Webpage. All of the symbols have been replaced with identifying integers.

The first three rows of this matrix are:

```
>> load nursery.mat;
>> nursery(1:3,:)
ans =
1 1 1 1 1 1 1 1 2
1 1 1 1 1 1 1 2 4
```

```
1 1 1 1 1 1 1 3 1
```

You should divide this data into equal-sized training and testing data sets as follows (the reset ensures that we will ll all use the same training/test split).

```
load nursery.mat;
reset(RandStream.getDefaultStream)
nursery = nursery(randperm(size(nursery,1)),:);
train = nursery(1:size(nursery,1)/2,:);
test = nursery(size(nursery,1)/2+1:end,:);
```

1. Estimate a Naive Bayes model using Maximum Likelihood (ML) on `train` and use it to predict the y labels of the `test` data. To model the class conditional distribution of each feature x_i $(i = 1, ..., 8)$ and the class distribution, use multinomial/multinoulli distributions. Report the accuracy of your classifier, and submit your code.

2. Modify the Nursery data by duplicating the last attribute 20 more times. You can do this by executing

   ```
   nursery = [nursery(:,1:end-1),repmat(nursery(:,end-1),1,20),nursery(:,end)];
   ```

   before splitting into `train` and `test`. Then run your ML Naive Bayes estimator on the new training data, and evaluate it on the new testing data. Explain in words why you see the change in accuracy that you observe.

3. Using the original data and the model you estimated in part 1, calculate the (joint) log-likelihood $\sum_i \log P(\mathbf{x}^i, y^i)$ of the training data. Using this model, is it possible to calculate the (joint) log likelihood of the test data? Explain your answer.

4. Now use Bayesian MAP estimators using Dirichlet priors having all their hyperparameters set to $\alpha_k = 2$ to estimate all of the multinomial/multinoulli distributions parameters in your Naive Bayes model from the original training data. What accuracy do you obtain on the test data using this model? Calculate the (joint) likelihood of the training data under the MAP model. Is it higher or lower than the likelihood of the training data under the ML model? Explain your answer. Now calculate the likelihood of the test data under the MAP model. Explain why you don't run into the same problems.

5. Using Dirichlet priors having all their hyperparameters set to $\alpha_k = 2$, you can also compute $P(y|\mathbf{x}, D)$ where D are the training data by integrating out the parameters of the model following the approach discussed in Section 4.5.3 of the textbook (and in the slides of the Bayesian statistics lecture (page 24)). What accuracy do you obtain on the test data using this approach?

2 Linear Regression

We study here different approaches to linear regression using a one dimensional dataset collected from a simulated motorcycle accident. The input variable, x, is the time in milliseconds since impact. The output variable, y, is the recorded head acceleration. The dataset `motor.mat` is available on the webpage. We have divided the full dataset into 40 training examples (variables Xtrain and Ytrain), and 53 test examples (variables Xtest and Ytest).

When fitting polynomial functions, as explored below, numerical problems can arise when the input variables take even moderate values. To minimize these, the input variables should be scaled to lie in the interval [-1,+1] before fitting. Use

$$x \leftarrow 2\left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}\right) - 1$$

where $x_{\min}$ and $x_{\max}$ are respectively the minimum and maximum values of the inputs $\{x^i\}$ over the training data set. Note that the same scaling must then be applied to the test data.

Submit your Matlab code for all the questions.

1. Consider a polynomial basis, with functions $\phi_j(x) = x^j$. Write a matlab function which evaluates these polynomial functions at a vector of points $x_i \in \mathbb{R}$. In a single figure, plot $\phi_j(x)$ for $-1 \leq x \leq 1$, and $j = 0, 1, 2, ..., 19$.

   Hint: To create a dense regular grid of points at which to evaluate and plot these functions, use the `linspace` command.

2. Consider the standard linear regression model, in which observations y^i follow a Gaussian distribution of mean $\mathbf{w}^{\mathrm{T}}\Phi(x^i)$ and variance σ^2. Define a family of regression models, each of which contains all polynomials

$\phi_j(x)$ of order $j \leq M$, that is $\Phi(x) = [\phi_0(x)\ \phi_1(x)\ \cdots\ \phi_M(x)]^{\mathrm{T}} = \left[1\ x\ \cdots\ x^M\right]^{\mathrm{T}}$ and $\mathbf{w} = [w_0\ \cdots\ w_M]^{\mathrm{T}}$ where M is a parameter controlling model complexity. Using the training data, compute the maximum likelihood (ML) estimate $\widehat{\mathbf{w}}$ of the regression parameters for models of order $M = 0, 1, 2, ..., 19$. Plot, as a function of x, the mean prediction $\widehat{\mathbf{w}}^{\mathrm{T}}\Phi(x)$ for models of order $M = 0, 1, 3, 5, 19$.

Hint: To compute $u = A^{-1}v$ in Matlab, rather than explicitly calling the `inv` command, use the following command to improve numerical stability:

>> `x = A \ b;`

3. Consider the following so-called RMSE (root mean square error) which is defined for any set of N points (x^i, y^i) by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y^i - \widehat{\mathbf{w}}^{\mathrm{T}}\Phi(x^i)\right)^2}.$$

What is the relationship between $RMSE$ and the ML estimate of σ^2? Evaluate and plot $RMSE$ as a function of the model order M for the training examples. Also do this for the test examples. Which model has the smallest training error, and which has the smallest test error? Together with the test data, plot the mean prediction $\widehat{\mathbf{w}}^{\mathrm{T}}\Phi(x)$ for both of these models.

4. We now consider radial basis function basis of the form

$$\Phi_j(x) = \exp\left(-\frac{\left(x - \mu_j\right)^2}{2\sigma^2}\right)$$

for $j = 1, ..., M$. For any model order M, we space the basis function centers μ_j evenly between -1 and 1. In Matlab, this can be done with the following command:

>> `mu = linspace(-1,1,M);`

We then set the bandwidth to triple the distance between basis centers, $\sigma = 3(\mu_2 - \mu_1)$. Finally, for any M, we also include a constant bias term $\Phi_0(x) = 1$. In three figures, plot these basis functions for $-1 \leq x \leq 1$ for models of order $M = 5, 10, 15$.

5. Repeat part 2 and part 3 for the radial basis function basis of part 4, and models of order $M = 5, 10, 15, 20, 25, 30$. Which basis performs better for this data?

In the previous question, you may have noticed that the ML estimates can become unstable for large model orders, M. We now consider an alternative, Bayesian approach in which the regression coefficients are assigned Gaussian priors

$$p(\mathbf{w}|\,\alpha) = \prod_{j=0}^{M} p\,(w_j|\,\alpha)$$

where w_j follows a Gaussian distribution of mean 0 and variance α^{-1}. [For sake of simplicity, I have here also introduced a prior on w_0].

6. What is the MAP estimate of $\mathbf{w}$ under the Gaussian prior above, and observation model of part 2? Consider polynomial and radial basis functions of order $M = 100$. For each of these two models, determine the MAP estimate $\widehat{\mathbf{w}}$ assuming hyperparameter values of $\sigma^2 = 400$ and $\alpha = 0.01$. Plot the mean prediction $\widehat{\mathbf{w}}^{\mathrm{T}}\Phi\,(x)$ for both of these models. Would it be possible to compute ML estimates for models of order $M = 100$?

7. Fix $\sigma^2 = 400$, and consider 100 candidate values for the regularization parameter α, logarithmically spaced between 10^{-8} and $10^{0} = 1.0$:

   `>> alpha = logspace(-8,0,100);`

   Using the `semilogx` command, plot the $RMSE$ defined in part 3 versus α for both the training and test datasets, and both basis families. Then plot the mean prediction $\widehat{\mathbf{w}}^{\mathrm{T}}\Phi\,(x)$ for the models which minimize the training and test error, for each basis family.

In the previous questions, we compared the accuracy of various models on test data, but did not provide a mechanism for choosing among models

given solely training data. Cross validation methods provide one popular, but computationally intensive, solution to this problem. The following question instead explores a Bayesian approach to model selection.

8. The marginal likelihood of the training data is given by

$$p\left(\left\{y^i\right\}_{i=1}^N \middle| \left\{x^i\right\}_{i=1}^N, \sigma^2, \alpha\right) = \int p\left(\left\{y^i\right\}_{i=1}^N \middle| \left\{x^i\right\}_{i=1}^N, \mathbf{w}, \sigma^2\right) p\left(\mathbf{w}\middle|\,\alpha\right) d\mathbf{w}$$

It can be shown that the marginal log-likelihood of the training data is equal to

$$\begin{aligned}\log\ p\left(\left\{y^i\right\}_{i=1}^N \middle| \left\{x^i\right\}_{i=1}^N, \sigma^2, \alpha\right) &= \frac{M+1}{2}\log\alpha - \frac{N}{2}\log\left(2\pi\sigma^2\right) \\ &\quad + \frac{1}{2}\log|\Sigma| - \frac{\alpha}{2}\mathbf{m}^{\mathrm{T}}\mathbf{m} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y^i - \widehat{\mathbf{w}}^{\mathrm{T}}\Phi\left(x^i\right)\right)^2\end{aligned}$$

where $\mathbf{m}$ and Σ are given by

$$\begin{aligned}\Sigma^{-1} &= \sigma^{-2}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \alpha\ I_{M+1} \\ \mathbf{m} &= \sigma^{-2}\Sigma\ \mathbf{\Phi}^{\mathrm{T}}\mathbf{y}\end{aligned}$$

where $\mathbf{\Phi}$ is the $N \times (M+1)$ matrix associated to the training data where

$$[\mathbf{\Phi}]_{i,j} = \Phi_j\left(x^i\right),$$

$\mathbf{y} = \begin{pmatrix} y^1 & \cdots & y^N \end{pmatrix}^{\mathrm{T}}$ and I_{M+1} is the $(M+1) \times (M+1)$ identity matrix of dimension.

Plot this quantity as a function of α, for the pair of basis families and range of hyperparameter values considered in part 7.

Hint: To avoid numerical underflow when computing the marginal log-likelihood above, you can exploit the following identity:

$$\log|\Sigma| = \sum_{j=0}^{M+1}\log\lambda_j$$

where λ_j are the eigenvalues of Σ.

9. For each model family (polynomial and radial basis function), what is the test accuracy for the hyperparameters which maximize the marginal likelihood of the training data? How do these compare to the models which actually performed best in part 7?