CSCI 5521: Pattern Recognition

Spring 2005, Prof. Schrater

Problem set 2:

10/14/09 **Due: 10/29/09 by 12pm**

Please submit all equation derivations electronically in a separate file than your code. You may use .ps or .pdf formats.

Note: Derive below means either to: a) derive a formula (you must show your work) and use it to evaluate the requested quantity (for 100% credit), or b) you can numerically compute the requested quantity (for 80% credit). Helpful matlab functions for this assignment include :

mean()
cov()
normpdf()
normcdf()

And for visualization:

meshgrid()
surf()
surfl()
contour()
scatter()

For numerical computations with probabilities:

repmat()
sum()

1. **Computing decision boundaries (10%):** Your botanist friend found out you are taken Pattern Recognition and recruited your help. She has found two kinds of mushroom in the forest, one type containing mainly a deadly poison (Agaric Moribundus) and the other (Agaric Stimulantus) containing both the poison and more of an amazing drug that can cure Lazius Enebriatus Studientis syndrome that causes afflicted students to spend inordinate amounts of time (and money) in the bars of Minneapolis. The two types of mushroom differ in their chemical composition. Your botanist friend gives you indirect and noisy measurements of the amounts of two chemical tracers: the poison *x* and the active ingredient *y*. From these measurements you were able to derive two class models, $p(x,y \mid \omega_1)$ and $p(x,y \mid \omega_2)$ which are both Gaussian with $\mu_1 = (8,2); \quad \mu_2 = (2,8); \quad C_1 = C_2 = \begin{bmatrix} 4.3 & 0 \\ 0 & 3.1 \end{bmatrix}; \quad p(\omega_1) = 4\,p(\omega_2).$
   a. Derive and plot Bayesian decision boundaries.
   b. Plot posterior probabilities using the surf() command.

2. **Using loss functions (set up same as problem 1)(10%):**

The two of you decide to market the mushrooms in a capsule form under the name Detoxifier. You use your classifier above to separate out the two kinds of mushroom.

However, due to imperfect separation, some of the capsules will have the poisonous mushroom. Given the amount of mushroom in each dose, you determine that ingesting a capsule with the poisonous kind will result in death in 1 out of 1000 instances. Each time someone dies, it costs your company an average of $4,000,000.00. The cost of harvesting the amount of mushrooms equal to one dose is $5.00. The price of the product per dose is $11.50.

- Construct a loss function for selling individual capsules assuming you want to maximize expected profit. Consider the possible actions- 1) sell, 2) discard the dose, and only consider losses and gains on average. Compute the optimal decision boundary with the derived loss function.
- What is the expected profit per dose of the optimal decision? Given a million students users, how many can be expected to die?

3. *Bayesian analysis of count data (15%):* For a binary classification problem, the training labels can be considered as a set of "coin flips" that determined which class was present for a given feature point. The goal is to learn $p(\omega_i \mid D_c)$, where D_c is the set of training class labels $\{l_1,\cdots,l_n\}$. Let the category of a new (test) point be l_{n+1}. The likelihood of the data is given by

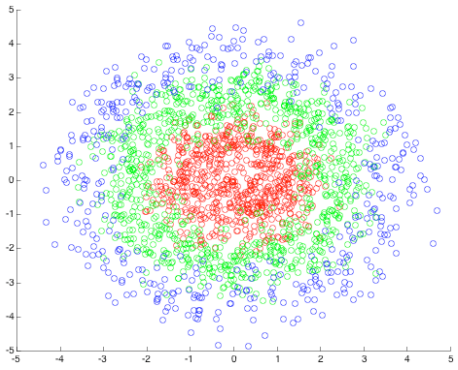
$$p(D_c \mid \omega_i,\theta) = \prod_{i=1}^{n} \theta^{l_i} (1-\theta)^{1-l_i} .$$

Note that with label notation if $l_{n+1} = 1$, $p(\omega_i \mid \theta) = p(l_{n+1} \mid \theta) = \theta^{l_{n+1}} (1-\theta)^{1-l_{n+1}} = \theta$, Additionally, note that the gamma function is directly related to the factorial function: $\Gamma(k+1) = k!$ Derive the posterior distribution $p(\omega_i \mid D_c)$.

4. *Noisy Labels (10%):*

- Problem 4.16 in Bishop's Pattern Recognition and Machine Learning

5. Generative approach to Classifiers (55%):



The goal of this problem is two-fold: Demonstrate what happens when you use the wrong model for data; show how transforming data can make a problem tractable. Load the data set bullseye.mat found in the directory with this file. The variable bullseyetrain is an array of structures that contains 20 sets of training data. Each set consists of 3 classes of 2-D features. Access the first set for the first class by pts=bullseyetrain.cls(1).pts; When plotted, the three classes should look similar to the above plot (with less data points).

a. *Using Gaussians when it is clearly the wrong model.* Assume each class is Gaussian distributed $p(\omega_i, x \mid D) \propto p(\omega_i \mid D) N(\hat{\mu}_i, \hat{\mathbf{C}}_i)$ with unknown

mean vector $\vec{\mu}_i$ and covariance $\mathbf{C}_i$. Compute Maximum likelihood estimates of the Gaussian parameters for each class using only the first set of training data (there are 20 sets). Also compute estimates of the class probabilities from the training data, and use as the prior class probabilities. How many total free parameters are there in this model?

b. Compute the Bayesian decisions via the posterior probabilities $p(\omega_i \mid x,D) \propto p(\omega_i \mid D)p(x \mid \hat{\mu}_i,\hat{\mathbf{C}}_i)$ (i.e. Use plug in the ML parameter estimates into the Gaussians. Classify the test data and compute the test error using maximum likelihood (e.g. count the number of misclassified points). Compute the conditional performance in the form of a 3x3 table: $p(\hat{\omega} = c_i \mid \omega = c_j)$ where $\hat{\omega}$ is the class estimate and ω is the true class for each data point. You can also derive the total error from this table. This table is sometimes called a confusion matrix.

c. Use the provided function GaussianRegionsDisplayUtility(meanstruct,Cstruct,classpriors,data) to help visualize the boundaries you have constructed. Note that the boundaries are not optimal. This occurred because Gaussians are not the right model for this data! However, performance can be improved. Change by hand the class probabilities to minimize training error. How does it perform on the test data?

d. *The less parameters, the better.* Assume that all three class means are [0,0] and all the covariances have the form: $\sigma_i^2\mathbf{I}$. Compute Maximum likelihood estimates of the Gaussian parameters for each class using only the first set of training data. Which has the higher error rate, part a) or part d)? By how much?

e. Repeat b) & d) for each of the 20 training sets. You will then have twenty maximum likelihood estimates of each set of parameters. Compute the variance across the mean estimates. How does the number of parameters impact the variability of the estimates?

f. ***Bayesian estimates:*** The data has a simpler description in polar coordinates. Use cart2pol() to transform all the data points to polar coordinates. Use scatter() to plot the transformed points. What you should see is that the resulting data looks Gaussian on *r* and uniform on theta. In other words, the two features provided to you can be expressed as one relevant feature and one irrelevant (theta). Ignore theta, and classify solely on *r*. Now the problem is 1-D and if you inspect the data, a Gaussian looks like a better model for all three classes. Assume each class is distributed:

$$p(r \mid \omega_i) = N(\mu_i,\sigma^2),$$

and that all have the same σ^2=1/4.

Given a prior for all three classes:

$$p(\mu) = N(\mu_0 = 0,\sigma^2 = 100),$$

compute the posterior distribution for the mean of each of the three classes. Next compute:

$$p(x \mid \omega_i, D) = \int p(x \mid \mu_i)\, p(\mu_i \mid D) d\mu_i .$$

Use this density estimate to classify test points as before. Compute test performance.