COS424
Homework #3
Due Tuesday, April 3, 2007

See the course website for important information about collaboration, late policies, grading policies, as well as where and when to turn in assignments.

Problem 1

Consider the Gaussian kernel (a.k.a. Gaussian radial basis function kernel) given by

$$K_c(\mathbf{x}, \mathbf{z}) = \exp(-c\,||\mathbf{x} - \mathbf{z}||)$$

where $\mathbf{x}$ and $\mathbf{z}$ are points in $\mathbb{R}^n$, and c is a positive constant. This problem considers how SVM's behave with this kernel as c becomes large.

(Note: There is an important difference between the way that SVM's were presented in class, and the way they are presented in the reading that was posted on-line. In class, we assumed that the hyperplane we are seeking must pass through the origin. In the reading, on the other hand, the hyperplane is not required to pass through the origin. This makes the math slightly more complicated. For this problem, you should follow the development that was given in class.)

a. For fixed $\mathbf{x}$ and $\mathbf{z}$, let us define

$$K_\infty(\mathbf{x}, \mathbf{z}) = \lim_{c \to \infty} K_c(\mathbf{x}, \mathbf{z})$$

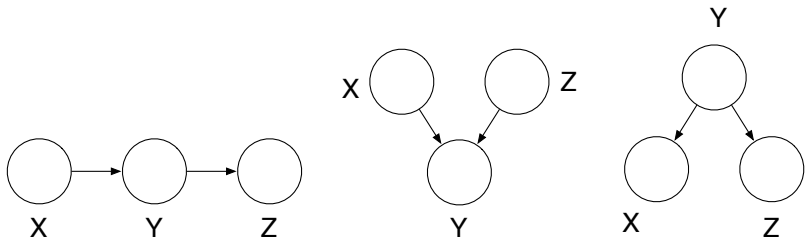
to be the limit of $K_c(\mathbf{x}, \mathbf{z})$ as c becomes very large. As a function of $\mathbf{x}$ and $\mathbf{z}$, what is the value of $K_\infty(\mathbf{x}, \mathbf{z})$?

b. Consider a fixed training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$. You can assume that all of the $\mathbf{x}_i$'s are distinct. In the limit as c becomes extremely large, what happens to the Lagrange multipliers α_i computed by SVM's? Although there are some technical issues that we are ignoring, you can assume that it is okay to answer this question by computing the Lagrange multipliers that would be found if we simply used the kernel K_∞ defined in the last part.

c. Suppose that c is finite but sufficiently large that we are close to the limiting behavior determined in the last parts of this problem. How will the classifier found by SVM's using this Gaussian kernel classify a new test point $\mathbf{x}$? How do these predictions relate to the nearest neighbor algorithm?

Problem 2 Prove that the k-means algorithm will terminate after a finite number of steps. (Hint: recall that k-means is a coordinate descent algorithm.)

Problem 3



For the three graphical models above, determine whether the following independence statements necessarily hold and prove your answers.

a. $X \perp\!\!\!\perp Z$

b. $X \perp\!\!\!\perp Z \,|\, Y$

In the R problems for this assignment, you will need to install the `cluster` *and* `pixmap` *packages. Additional packages can easily be installed with the* `install.packages` *command. Once a package is installed, use the* `library` *command to load it into memory. You will also need the files* `stubs.R`, `helper.R`, *and the data. These can be found on the assignments page,* `http://www.cs.princeton.edu/courses/archive/spr07/cos424/assignments.html`.

In all of these problems, please hand in the code that was used to generate the results. In addition, please hand in PDF files of all plots. Hand in files using the naming convention `problem-4b.pdf` *and* `problem-4b.R`.

Problem 4 Implement the k-means and k-medoids algorithms in R by filling in the functions in `stubs.R`. Initialize the centers/medoids by choosing k data points at random, and assigning the k means/medoids to those points.

(Note: R implements the k-means algorithm in `kmeans` and the k-medoids algorithm in `pam`, both as part of the `cluster` package. You *may not* use either of these functions in this problem. You may use them in later problems.)

a. Run your algorithms with 10 random restarts using $k = 3$ on the data in `fake-data.dat`, which is a 21×2 matrix that represents 21 2d data points. For each algorithm, plot the points color coded by cluster assignment, the cluster centers or medoids, and report the final objective value.

b. One can use the *silhouette statistic* to choose k in the k-means algorithm. For the ith data point, the silhouette statistic is

$$\frac{b(i) - a(i)}{\max(b(i), a(i))},$$

where $a(i)$ is the average distance to other points in the cluster assigned to the ith data point, and $b(i)$ is the average distance to points in the cluster closest to the cluster assigned to the ith data point. One can choose an "optimal" k by maximizing the per-data average silhouette statistic across various values of k. (This statistic is efficiently implemented in the function `silhouette`.)

Learn how to use the R function `silhouette` by reading the help file. Apply the silhouette statistic to your clusterings from the previous question. What is the natural value of k? In general, why does the silhouette statistic make sense?

In the following problems, you will be clustering real data sets. You may use ***`kmeans`****, which is likely to be faster than your implementation because it calls fast Fortran and C code. This function has several parameters to play with; type* ***`help(kmeans)`*** *to learn how to use it. We suggest changing* ***`iter.max`*** *to 100 or more.*

Problem 5 The file `faces.dat` is a 471 x 361 matrix, where each row is a grey scale image of a face. Load it using `data <- matrix(scan("faces.dat", nrow=471, ncol=361)` You can use `plot.face(x)` to plot one of the rows of the matrix.

Cluster the faces by running k-means for $k = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$, each time choosing the clustering that gives the best objective of 10 random restarts. Note that this is *not* vector quantization. You are clustering a collection of images, rather than the pixels of a single image.

a. Plot the objective as a function of iteration for different values of k (on the same plot). What can be gleaned from this plot?

b. Plot the converged objective as a function of k.

c. Plot the k means for $k = \{4, 8, 16\}$, each on one plot. Compare the different clusterings. What has been captured in each?

d. For your choice of k, compare the k means and k medoids clustering of the faces. Is there a difference in this case?

Problem 6 The function `load.album.cover(filename)` returns a 57600×3 matrix, where each row is a pixel in an image of an album cover downloaded from Amazon. Load the PNM files for the Beatles albums "Let It Be," "Sergeant Pepper's Lonely Hearts Club Band," and "The White Album." (All three of these albums are great.)

Run the k-means algorithm (i.e., vector quantization) on each of these matrices for different values of k. Make the following plots.

a. Plot the final objective as a function of k for each album cover (on the same plot). Comment on what this plot demonstrates.

b. For fixed k, plot the objective as a function of iteration for each album cover (on the same plot). Comment on what this plot demonstrates.

c. For "Let It Be," plot the vector quantized image for 9 different values of k (on one plot). Use the function `plot.album.cover(data)`.

Problem 7 In this problem, you will analyze all the articles from the journal *Science* in the year 2000. (Thanks to JSTOR for providing the data.) Many of the parameters of this analysis will be left for you to decide.

For these files you will need to use `science2k-vocab.dat` and `science2k-titles.dat`, which are vectors of terms and titles respectively. (You can load these with the R function `readLines`.)

a. The file `science2k-doc-word.dat` contains a 1373×5476 matrix, where each row is an article in *Science* described by 5476 word features. The articles and words are in the same order as in the vocabulary and titles files above.

   To obtain the features, we performed the following transformation. First, we computed per-document smoothed word frequencies. Second, we took the log of those frequencies. Finally, we centered the per-document log frequencies to have zero mean.

   Cluster the documents using k-means and various values of k (go up to at least $k = 20$). Select a value of k.

   For that value, report the top 10 words of each cluster in order of the largest positive distance from the average value across all data. More specifically, if $\bar{\mathbf{x}}$ is the 5476-vector of average values across documents and $\mathbf{m}_i$ is the ith mean, report the words associated with the top components in $\mathbf{m}_i - \bar{\mathbf{x}}$. Report the top ten documents that fall closest to each cluster center.

   Comment on these results. What has the algorithm captured? How might such an algorithm be useful?

b. The file `science2k-word-doc.dat` contains a 5476×1373 matrix, where each row is a *term* in *Science* described by 1373 "document" features. These are transformed document frequencies (as above).

   Repeat the analysis above, but cluster terms instead of documents. Comment on these results. How might such an algorithm be useful? What is different about clustering terms from clustering documents?