MSCBIO 2070/02-710: Computational Genomics, Spring 2014

A4: clustering, classification, spline, time-series data analysis

Due: March 19, 2014 by email
TA in charge: Silvia Liu (silvia.shuchang.liu@gmail.com)

Your goals in this assignment are to

1. Master the common clustering methods and select the suitable one for specific data
2. Master the common classification methods and make the comparison
3. Understand cubic spline interpolation
4. Implement bi-clustering algorithm into time-series dataset

What to hand in. Write a short report addressing each of the questions below (either a hand-written report or a report submitted as a pdf file is acceptable). The report should be self-contained, we should not have to run your code to be convinced that your code is correct. Be sure to comment your code and use any programming language you like (if not specified). Also, include instructions on how to run your code. In your report you can assume that we know the context of the questions, so do not spend time repeating material in the hand-out or in class notes. Email a zip file containing the complete code (if any) and pdf file (if any) to: Silvia Liu, *silvia.shuchang.liu@gmail.com* with the Subject: *2070S14 A4 YourName*.

1 [15 points] Which clustering method should we use?

Which clustering method(s) is/are most likely to produce the following results (figure 1, 2 and 3)? Choose the most likely method(s) and briefly explain why it/they will work better where others will not in at most 3 sentences. Here are the five clustering methods you can choose from:

1. Hierarchical clustering with single linkage
2. Hierarchical clustering with complete linkage
3. Hierarchical clustering with average linkage
4. K-means clustering
5. Gaussian Mixture Model (GMM) clustering

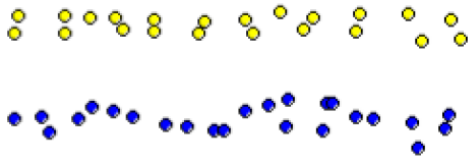

Figure 1: Clustering result for data 1.

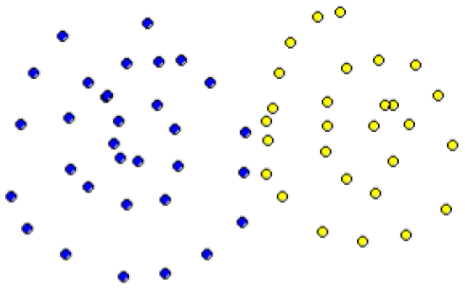

Figure 2: Clustering result for data 2.

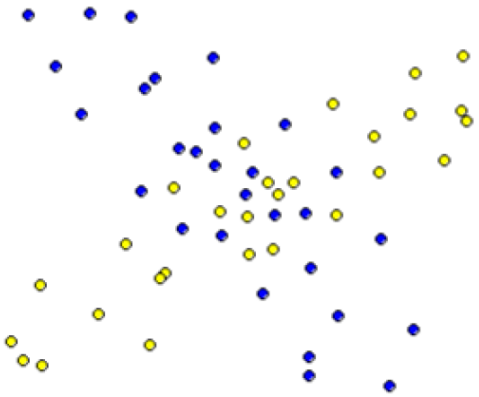

Figure 3: Clustering result for data 3.

- Hierarchical clustering with single link is most likely to well. GMM can also produce a decision boundary that can produce such clustering result, but depending on initialization it might converge to a different set of clusters (left half vs. right half). Other hierarchical clusterings wont really work well because at some point, two intermediate clusters from different true cluster will have shorter cluster distance than two from the same true cluster.
- K-means or GMM is most likely. Hierarchical clustering would not work since the early few steps will group instances near the decision boundary.
- Among the five methods, only GMM has the capability of handling overlapping clusters. So GMM is the only method that would result in such clusters.

2 [25 points] Discriminative VS Generative Classifiers

A common debate in machine learning has been over generative versus discriminative models for classification. In this question we will explore this issue, both theoretically and practically. We will consider Naive Bayes and logistic regression classification algorithms.

Your tasks:

1. (4 points) Briefly describe the functions Naive Bayes and Logistic-regression optimize.

   Naive Bayes

   $$f_{NB}(x) = \arg\max_y P(x_1, x_2, \ldots, x_d|y)P(y) = \arg\max_y \prod_{i=1}^{d} P(x_i|y)P(y) \quad (1)$$

   Logistic regression

   $$P(y=1) = \frac{e^{\alpha+\beta_1 x_1+\ldots\beta_k x_k}}{1+e^{\alpha+\beta_1 x_1+\ldots\beta_k x_k}} \quad (2)$$

   Naive-Bayes optimizes joint data likelihood $P(XX, Y)$ with respect to the conditional independence assumptions, while Logistic Regression directly optimizes conditional data likelihood $P(Y|XX)$.

2. (3 points) Consider the data containing binary class label $y \in \{T, F\}$ and training example X with 2 binary attributes $X_1, X_2 \in \{T, F\}$. What is the minimum number of parameters that you need to know/evaluate if you are to classify an example using the Naive Bayes classifier? Explain why.

   The Naive Bayes classifier learns the the conditional probabilities $P(X_1|Y)$ and $P(X_2|Y)$ as well as the class probability $P(Y)$. To represent $P(Y)$ we need only one parameter π because $P(Y=T) + P(Y=F) = 1$. The $P(X_i|Y)$ can be represented using only $P(X_i = T|Y = T) = \theta_{i1}$ and $P(X_i = T|Y = F) = \theta_{i0}$. We do not need additional parameters to represent $P(X_i = F|Y = F) = 1 - \theta_{i0}$ or $P(X_i = F|Y = T) = 1 - \theta_{i1}$. Therefore we need 1 parameter for $P(Y)$, 2 parameters for $P(X_1|Y)$, and 2 parameters for $P(X_2|Y)$ resulting in a total of 5 parameters.

3. Let the class prior be $P(Y = T) = 0.6$ and also let $P(X_1 = T|Y = T) = 0.7, P(X_1 = F|Y = F) = 0.8, P(X_2 = T|Y = T) = 0.4, P(X_2 = F|Y = F) = 0.5$. So attribute X_1 provides slightly stronger evidence about the class label than X_2. For this problem, you should assume that the true distirubtion of X_1, X_2 and Y satisfies the Naive Bayes assumption of conditional independence with the above parameters.

   (a) (3 points) Assume X_1 and X_2 are truly independent given Y. Write down the Naive Bayes decision rule given $X_1 = x_1$ and $X_2 = x_2$.

   The mathematical form of the decision rule for the Naive Bayes classifier is given where $Ind(\cdot)$ is the indicator function. Notice that in the sum of log ratios is used rather than a direct comparison of $P(Y = T|XX) > P(Y = F|XX)$. While both are equivalent the sum of logs form is often more numerically stable and easier to work with.

   $$\hat{Y} = Ind\left[\ln\left(\frac{P(Y=1)}{P(Y=0)}\right) + \sum_{i=1}^{n} \ln\left(\frac{P(X_i|Y=1)}{P(X_i|Y=0)}\right) > 0\right]$$

(b) (6 points) Show that if Naive Bayes uses both attributes, X_1 and X_2, the error rate is 0.26. Also calculate the error rates for using only each one of the attributes and compare the three results. The error rate is defined as the probability that each class generates an observation where the decision rule is incorrect.

The joint probability is give by

$$P(X_1 = x_1, \ldots, X_n = x_n, Y = y) = P(Y = y) \prod_{i=1}^{n} P(X_i = x_i | Y = y)$$

We can get the following decision table if both attributes are used. In the table, joint probabilities corresponding to incorrect predictions is colored yellow. Based on the table, we can calculate the error rate as $0.04 + 0.04 + 0.072 + 0.108 = 0.26$.

X_1	X_2	Y	$P(X_1, X_2, Y)$	$\hat{Y}$
1	1	1	0.168	1
1	1	0	0.04	1
1	0	1	0.252	1
1	0	0	0.04	1
0	1	1	0.072	0
0	1	0	0.16	0
0	0	1	0.108	0
0	0	0	0.16	0

If we only consider X_1 we can get the following decision table. The error rate is $0.08 + 0.18 = 0.26$.

X_1	Y	$P(X_1, Y)$	$\hat{Y}$
1	1	0.42	1
1	0	0.08	1
0	1	0.18	0
0	0	0.32	0

If we only consider X_2 we can get the following decision table. The error rate is $0.2 + 0.2 = 0.4$.

X_2	Y	$P(X_2, Y)$	$\hat{Y}$
1	1	0.24	1
1	0	0.2	1
0	1	0.36	1
0	0	0.2	1

We see that a Naive Bayes classifier that uses both variables performs the same or better than a Naive Bayes classifier that uses only one of the two. This implies that there is unique (uncorrelated) information in each variable.

(c) (6 points) Now suppose that we create a 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, $X_2 = X_3$. Obviously, X_2 and X_3 are not conditionally independent. But if we still assume that the independence assumption holds, what error rate will we get? Compare the new error rate with the original one (only using attributes X_1 and X_2) and explain what is happening with Naive Bayes?

We can get the following decision table by three attributes. The error rate is $0.02 + 0.02 + 0.02 + 0.02 + 0.0288 + 0.0432 + 0.0432 + 0.0648 = 0.26$.

X_1	X_2	X_3	Y	$P(X_1, X_2, X_3, Y)$	$\hat{Y}$
1	1	1	1	0.0672	1
1	1	1	0	0.02	1
1	1	0	1	0.1008	1
1	1	0	0	0.02	1
1	0	1	1	0.1008	1
1	0	1	0	0.02	1
1	0	0	1	0.1512	1
1	0	0	0	0.0002	1
0	1	1	1	0.0288	0
0	1	1	0	0.08	0
0	1	0	1	0.0432	0
0	1	0	0	0.08	0
0	0	1	1	0.0432	0
0	0	1	0	0.08	0
0	0	0	1	0.0648	0
0	0	0	0	0.08	0

The Naive Bayes is based on the independent assumption. But actually X_2 and X_3 are not independent to each other. The new Naive Bayes classifier with three attributes may potentially overfit the data.

(d) (3 points) Does Logistic Regression suffer from the same problem? Explain why.

Logistic Regression model is based on the function (2). Here each attribute is not required to be independent.

If there are only two attribute,

$$P(y=1) = \frac{e^{\alpha+\beta_1 x_1+\beta_2 x_2}}{1+e^{\alpha+\beta_1 x_1+\beta_2 x_2}}$$

But when adding another attribute X_3 (the exact copy of X_2), we have,

$$P(y=1) = \frac{e^{\alpha+\beta_1 x_1+\frac{1}{2}\beta_2 x_2+\frac{1}{2}\beta_2 x_3}}{1+e^{\alpha+\beta_1 x_1+\frac{1}{2}\beta_2 x_2+\frac{1}{2}\beta_2 x_3}}$$

So there will be no influence on the prediction error rate.

3 [15 points] Cubic Spline

Cubic Spline Method

Given a set of n control points, cubic spline constructs $n-1$ piecewise third-order polynomials between the points. The splines need to satisfy the following properties:

- Each spline need to pass through its left-handed and right-handed control points.
- The splines located on the left and right hand of a control point should be continuous and have the equal first derivative value at that point.
- The splines located on the left and right hand of a control point should have the equal second derivative value at that point.
- The second derivatives at the endpoints (first and last control points) should be zero.

Your tasks:

1. (5 points) Assume we have a value of v for point i. Let $S_1 = ax^3 + bx^2 + cx + d$ be the spline to the left of this point, and $S_2 = ex^3 + fx^2 + gx + h$ be the spline to its right. You may parameterize both splines as $x \in [0, 1]$. How many equations are defined by point i? Write all these equations and simplify each as much as you can. *[Hint: You may need to implement the properties listed above.]*

   There are four equations defined by each internal point (and one by each of the two endpoints). The four equations are:

   1. Continuous value at end point

   $$a1^3 + b1^2 + c1 + d = v \quad \Rightarrow \quad a + b + c + d = v$$

   2. Continuous value at start point

   $$e0^3 + f0^2 + g0 + h = v \quad \Rightarrow \quad h = v$$

   3. Equality of first derivative

   $$3a1^2 + 2b1 + c = 3e0^2 + 2f0 + g \quad \Rightarrow \quad 3a + 2b + c = g$$

   4. Equality of second derivative

   $$6a1 + 2b = 6e0 + 2f \quad \Rightarrow \quad 3a + b = f$$

2. (5 points) Assume we have a value of u for the first control point. Let $S = \alpha x^3 + \beta x^2 + \gamma x + \theta$ be the spline located on its right. Also parameterize the spline as $x \in [0, 1]$. How many equations can you write for this point? Try to simplify each equation. In order to get all the splines, how many equations for n control points do we need?

   There are two equations defined by the first point.

   1. Continuous value

   $$\alpha 0^3 + \beta 0^2 + \gamma 0 + \theta = u \quad \Rightarrow \quad \theta = u$$

   2. Second derivative equals to zero

   $$6\alpha 0 + 2\beta = 0 \quad \Rightarrow \quad \beta = 0$$

   For n points, we need $n-1$ splines. Each spline has 4 parameters to estimate. So totally we need $4(n-1)$ equations. Each internal point can define 4 equations and each end point can define 2 equations. Totally there are $4(n-2) + 2 \times 2 = 4(n-1)$ equations. These are exactly what we need to estimate all the parameters.

3. (5 points) In class we actually discussed approximating splines, that is splines that contain less control points than the number of actual measured points. One important issue is how to choose the number of control points to assign. If we assign too many, we will overfit the data. But if we assign too few, we might not be able to accurately reconstruct the underlying expression curve. Assume control points are uniformly spaced. Suggest a method for determining the number of control points we should use.

   We can use cross validation for this task. Start with interpolating splines (the largest number of control points) and hide a complete column (one experiments), say out of 10 time points we are hiding time point 5. Following spline assignment we can compute the predicted values for TP 5 (since we have a continuous curve). Next, we test how far are the values predicted from the values we have hidden. This is repeated for other internal TPs and an average loss is computed. Next, we reduce the number of control points by one and repeat this process until we end with 4 control points (the lowest number possible). Then we pick the number of control points that yielded the lowest average error between the predicted and hidden values and use these.

4 [45 points] Biclustering: Application in Time-series Data

In this problem you will develop and implement a bi-clustering algorithm. A bi-cluster is a cluster containing a subset of the experiments and a subset of the genes. In this problem we will not allow overlap between the bi-clusters, though other methods allow such overlap. By answering the questions below you will develop (and implement) a method that uses bipartite graphs for bi-clustering.

Data Description

We will implement the bi-clustering algorithm into a time-series data set. The data file named *alphaCycle.txt* can be downloaded online from our website. In the data matrix, each row represents a gene and each column corresponds to a time point (experiment). So each value in row i and column j corresponds to the expression level (log ratio) of gene i in time point j. File *alphaGenes.txt* contains the corresponding gene names.

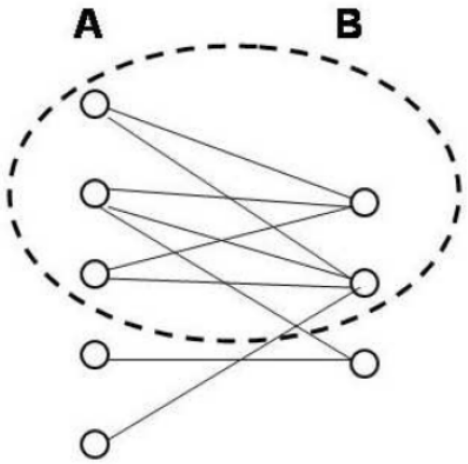

Figure 4: A bipartite graph. The dashed circle contains a complete subgraph in this graph, serving as a good candidate for a bi-cluster.

Your task:

1. (3 points) How could we use an undirected bipartite graph to represent the gene expression data matrix? What do the nodes and edges represent?

   We can have the left nodes represent genes and the right nodes represent time points. An edge in the graph means that the gene is either expressed or repressed in that time point.

2. (5 points) Since gene expression data contains non-discrete values, we will first discretized the data. Every value above 0.9 will be set to 1 and every value below -0.9 will be set to -1. Values between -0.9 and 0.9 will be set to 0. Discretize your data matrix by this method and show how many values are assigned to 1 and -1 respectively.

   Totally 520 values are assigned to 1 and 416 values are assigned to -1.

3. (3 points) Using one unweighted bipartite graph (that is, all edges have the same weight of 1) as you have described above, how can you represent both activation (1) and repression (-1)? (Remember, we would like to cluster activated genes in a different cluster than the repressed ones.)

We will use two nodes for each time point. One will be connected to all over-expressed genes and the other to the under-expressed ones. Thus, we will have twice the number of nodes than the number of time points.

Alternatively, we can use both two copies of gene nodes and two copies of time nodes. The edges between first copy of gene nodes and time nodes will represent the higher expression, and the edges connecting second copy of gene and time nodes will represent the lower expression.

4. (3 points) Assume the graph has a bounded out degree on the left (that is, no node on the left side has more than d outgoing edges). Also, assume that we are looking for complete subgraphs (figure 4). That is a subset of the nodes on the left ($l \subseteq A$) and a subset of the nodes on the right ($r \subseteq B$), where each node in l is connected to all nodes in r and vice versa. What is the largest possible size of r?

   Since each gene node (left node) is connected to at most d experiment nodes (right nodes), we can only have at most d nodes from B in a complete subgraph.

5. (5 points) Let n be the number of genes. You are asked to develop a $O(n2^d)$ algorithm for finding the maximal complete subgraph (where maximal means that it has the most number of edges). Explain your algorithm and its complexity.
   [Hint: Here is a nice paper FYI, `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.8302&rep=rep1&type=pdf`*.]*

   Since each node in A is connected to at most d nodes in B, the total number of subsets of the d nodes is 2^d. For each of these subsets we need to determine a score. This can be done by starting with the singleton sets and increasing them by one node at a time. Each time we add a node we need to perform a merging step to merge the set of nodes in A connected to the subset we have with the nodes connected to the new node we are adding. While this can take a time linear in the number of nodes connected to the new node we had, this cost can be amortized over all nodes by not considering a subset more than once. Thus, the total running time will be $O(n2^d)$.

6. (10 points) Implement the algorithm in the discretized data matrix to find the bi-cluster. After finding one maximal complete subgraph, you can set the edges in that subgraph to 0 and repeat the method to find the next bi-cluster. Draw a table to show out the dimensions of the top 5 bi-clusters you detected.

	# gens	# time points	# edges	activated / repressed
1	99	1	99	repressed
2	66	1	66	activated
3	58	1	58	activated
4	55	1	55	activated
5	51	1	51	activated

7. (10 points) Did you find any problem with this method? Try to come up with a solution to the problem and implement revised algorithm into the discretized data. Again, draw a table to show out the dimensions of the top 5 bi-clusters.

   When checking the bi-cluster dimensions in the above table, we find that all the top bi-clusters only have one time point. Biologically, we are more interested in the genes that have high/low expression levels over multiple time points. Here I change the code slightly, only recording the bi-clusters that have more than 3 time points. The dimensions of the new top 5 bi-clusters are shown in the following table. *This is an open question. Any reasonable answers will be accepted.*

	# gens	# time points	# edges	activated / repressed
1	18	3	54	repressed
2	10	3	30	repressed
3	7	4	28	activated
4	9	3	27	activated
5	8	3	24	activated

8. (6 points) After detecting the bi-clusters, we need to know whether the genes detected are meaningful (or what we could learn form the bi-clusters). GO enrichment analysis is one of the popular methods. For each of the top bi-cluster detected in step 7,

   (a) Go to the FuncAssociate website, http://llama.mshri.on.ca/funcassociate/.

   (b) In step 1, select *species* as *Saccharomyces cerevisiae*.

   (c) In step 2, select *namespace* as *sgd_systematic*.

   (d) In step 3, paste the names of the genes in the bi-cluster in *Query List*.

   (e) Click *Functionate!* button.

   (f) Sort the results by p-values in ascending order. Draw a table to list the top 3 GO categories, showing the Gene-Ontology-ID, Gene-Ontology-Attribute and p-value in each column.

   Briefly explain the GO analysis and what we could learn form the table?

   First bi-cluster:

	Gene-Ontology-ID	Gene-Ontology-Attribute	p-value
1	0000788	nuclear nucleosome	1.138×10^{-19}
2	0000786	nucleosome	8.853×10^{-19}
3	1990104	DNA bending complex	8.853×10^{-19}

   Second bi-cluster:

	Gene-Ontology-ID	Gene-Ontology-Attribute	p-value
1	0005576	extracellular region	8.448×10^{-14}
2	0009277	fungal-type cell wall	1.998×10^{-11}
3	0005618	cell wall	3.060×10^{-11}

   Third bi-cluster:

	Gene-Ontology-ID	Gene-Ontology-Attribute	p-value
1	0000788	nuclear nucleosome	4.125×10^{-21}
2	0000786	nucleosome	2.145×10^{-20}
3	1990104	DNA bending complex	2.145×10^{-20}

   Fourth bi-cluster:

	Gene-Ontology-ID	Gene-Ontology-Attribute	p-value
1	0005576	extracellular region	8.270×10^{-8}
2	0009277	fungal-type cell wall	9.677×10^{-8}
3	0005618	cell wall	1.306×10^{-7}

   Fifth bi-cluster: no result.

   GO analysis is to check whether the selected genes are enriched in some important pathways. From the table we could see that most of the pathways are related to cell cycle (for instance, related to nucleosome or cell wall pathways).