10-701/15-781, Machine Learning: Homework 5

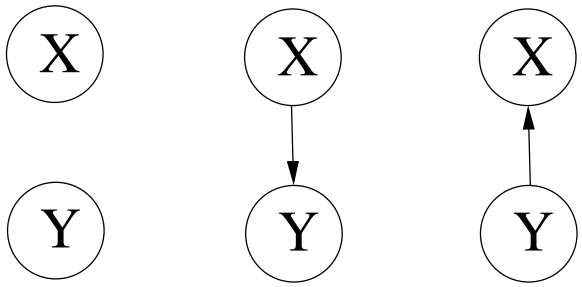
Aarti Singh
Carnegie Mellon University

- The assignment is due at 10:30 am (beginning of class) on **Tues, Dec 7, 2010**.
- Separate you answers into five parts, one for each TA, and put them into 5 piles at the table in front of the class. Don't forget to put both your name and a TA's name on each part.
- If you have question about any part, please direct your question to the respective TA who designed the part (however send your email to 10701-instructors@cs list).

1 Bayes Net Structure Learning [Rob Hall, 20 points]

In this section we will consider learning the structure of a bayesian network for two binary random variables, $x, y \in \{0, 1\}$. Although the proceedings will be extremely simple, since we eschew issues that may arise in higher dimensional problems, we will hopefully gain some intuition about some of the caveats of structure learning.

For the two variables there are evidently three possible structures:



We will call these structures a,b, and c in order from left to right in the picture.

1.1 Parameterization – 2 points

Give the number of parameters required by each structure, if we do not restrict the class of distributions to any particular parametric family. Explain your answer.

1.2 Max Likelihood Inference – 10 points

Suppose we obtain a set of data $D = \{(x_i, y_i)\}_{i=1}^n$. We maximize the likelihood of each model individually on the data. We will examine the maxima attained by each model and the relationship between them.

We denote by:

$$\ell_a^\star(D) = \max_{\theta_a} \ell_a(D|\theta_a)$$

The maximum value of the likelihood for model a, where the parameters are denoted as θ_a. Likewise we may define $\ell_b^\star, \ell_c^\star$ for the other models.

1. **[5 points]** Which of the following is true, prove your answer.
   - $\ell_a^\star(D) \leq \ell_b^\star(D)\ \forall D$.
   - $\ell_a^\star(D) = \ell_b^\star(D)\ \forall D$.
   - $\ell_a^\star(D) \geq \ell_b^\star(D)\ \forall D$.
   - None of the above.
2. **[5 points]** Which of the following is true, prove your answer.
   - $\ell_b^\star(D) \leq \ell_c^\star(D)\ \forall D$.
   - $\ell_b^\star(D) = \ell_c^\star(D)\ \forall D$.
   - $\ell_b^\star(D) \geq \ell_c^\star(D)\ \forall D$.
   - None of the above.

1.3 Model Selection [8 points]

Consider using AIC (Akaike's information criterion) to choose between models, rather than likelihood. AIC for model is defined by:

$$AIC = -2\ell^\star + 2p$$

Where p is the number of parameters in the model. The best model is the one with the minimal AIC.

1. **[2 points]** For the above three models, is there always a unique minimizer of the AIC? Explain yourself.
2. **[6 points]** For some fixed dataset D, denote:

$$\#(x,y) = \sum_{i=1}^{n} 1\{x_i = x \text{ and } y_i = y\}$$

$$\#(x,\cdot) = \sum_{y\in\{0,1\}} \#(x,y)$$

$$\#(\cdot,y) = \sum_{x\in\{0,1\}} \#(x,y)$$

   Then consider the "sample mutual information" in a sample of size n:

$$\hat{I}_n = \sum_{x\in\{0,1\}} \sum_{y\in\{0,1\}} \frac{\#(x,y)}{n} \log \frac{n\#(x,y)}{\#(x,\cdot)\#(\cdot,y)}$$

   How big must the "sample mutual information" be, in order for the AIC score of (b) to be lower than the AIC score of (a)?

2 PCA [Min Chi, 20 points]

1. **[4 points:]** An example of raw dataset, D_1, has 64 records from 64 different users, that is $n = 64$. Each record in D_1 is a vector of length 6830, in other words, 6830 features $p = 6830$. If we do Principal Component Analyses (PCA) of this dataset D_1, how many principal components with non-zero variance would we get? Explain why?

2. **[16 points:]** D_2 is a data set about the information on each of US state. Therefore, we have $n = 50$. For each state, there are eight features, which is described in Table 1:

Table 1: Descriptions of the Eight Attributes

Attribute	Explanation
Population	in thousands
Income	dollars per capita
Illiteracy	Percent of the adult population unable to read and write
Life Exp	Average years of life expectancy at birth
Murder	Number of murders and non-negligent manslaughters per 100,000 people
HS Grad	Percent of adults who were high-school graduates
Frost	Mean number of days per year with low temperatures below freezing
Area	In square miles

In the following, we will do two different principal component analyses (PCAs) of this dataset D_2. The two PCAs are called as PCA_1 and PCA_2 respectively. The only difference between the two PCAs is that one has the variables standardized to variance 1 before calculating the covariance matrix and its eigenvalues, the other does not. The summary statistics for these variables (without standardizing to variance 1) are listed in Table 2:

Table 2: Summary Statistics for the Seven Attributes

Attribute	Min	Median	Mean	Max
Population	365	2838	4246	21198
Income	3098	4519	4436	6315
Illiteracy	0.500	0.950	1.170	2.800
Life Exp	67.96	70.67	70.88	73.60
Murder	1.400	6.850	7.378	15.100
HS Grad	37.80	53.25	53.11	67.30
Frost	0.00	114.50	104.46	188.00
Area	1049	54277	70736	566432

Generally speaking, determining which principal components account for which parts of the variance can be done by looking at a Scree Plot. A Scree Plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each principal component(PC). The PCs are ordered, and by definition are therefore assigned a number label, by decreasing order of contribution to total variance. The PC with the largest fraction contribution is labeled with the label name. Such a plot when read left-to-right can often show a clear separation in fraction of total variance where the 'most important' components cease and the 'least important' components begin. The point of separation is often called the 'elbow'. (In the PCA literature, the plot is called a 'Scree' Plot because it often looks like a 'scree' slope, where rocks have fallen down and accumulated on the side of a mountain.)

The Figures and Tables following show some displays for the PCA_1 and PCA_2 respectively, which you will need to use to answer the following questions.

The Scree Plot of PCA_1 is displayed in Figure 1 and projections of the features on to the first two PCs are listed in Table 3.

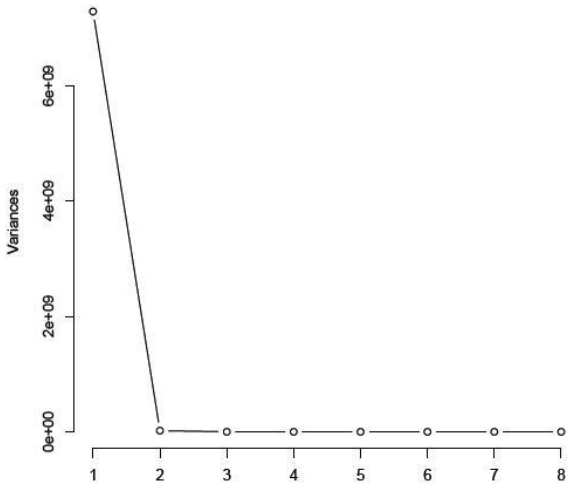


Figure 1: Scree plot for PCA_1

Table 3: Projections of the features on to the first two principal components of PCA_1.

Attribute	PC1	PC2
Population	1.18×10^{-3}	-1.00
Income	2.62×10^{-3}	-2.8×10^{-2}
Illiteracy	5.52×10^{-7}	-1.42×10^{-5}
Life Exp	-1.69×10^{-6}	1.93×10^{-5}
Murder	9.88×10^{-6}	-2.79×10^{-4}
HS Grad	3.16×10^{-5}	1.88×10^{-4}
Frost	3.61×10^{-5}	3.87×10^{-3}
Area	1.00	1.26×10^{-3}

The Scree Plot of PCA_2 is displayed in Figure 2 and projections of the features on to the first two PCs are listed in Table 4.

Table 4: Projections of the features on to the first two principal components of PCA_2.

Attribute	PC1	PC2
Population	0.1260	0.4110
Income	−0.2990	0.5190
Illiteracy	0.4680	0.0530
Life Exp	−0.4120	−0.0817
Murder	0.4440	0.3070
HS Grad	−0.4250	0.2990
Frost	−0.3570	−0.1540
Area	−0.0334	0.5880

(a) **[2 point:]** Recall the only difference between the PCA_1 and PCA_2 is that one has the variables standardized to variance 1 before calculating the covariance matrix and its eigenvalues, the other does not. Based on the Scree Plots and Projections of the features on to the first two principal

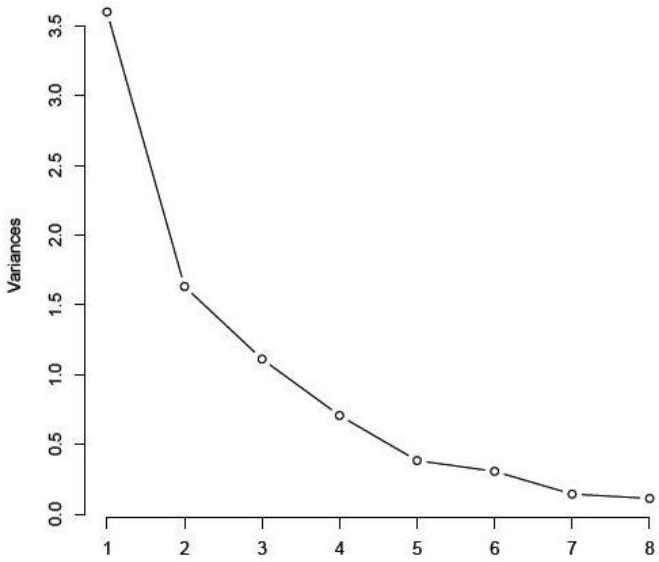


Figure 2: Scree plot for PCA_2

components tables for PCA_1 and PCA_2 respectively, which one has the variables standardized to variance 1 before calculating the covariance matrix and its eigenvalues? Explain briefly.

(b) **[2 point:]** From the Scree Plot of PCA_1 in Figure 1, where is a big gap or elbow and what is the reasonable number of principal components to be retained?

(c) **[3 point:]** Describe, in words, the first two principal components of PCA_1. (Describe which are the features most relevant to each PC and which are characteristics of the variance in the data was captured by each PC.)

(d) **[2 point:]** From the Scree Plot of PCA_2 in Figure 2, where is a big gap or elbow and what is the reasonable number of principal components to be retained?

(e) **[3 point:]** Describe, in words, the first two principal components of PCA_2.(Describe which are the features most relevant to each PC and which are characteristics of the variance in the data was captured by each PC.)

(f) **[4 point:]** Would you rather do PCA_2 or PCA_1 for the PCA analysis? Pick one and explain your choice. (A choice with no or inadequate reasoning will get little or no credit.)

3 Spectral Clustering [Leman, 25 points]

There is a class of clustering algorithms, called spectral clustering algorithms, which has recently become quite popular. Many of these algorithms are quite easy to implement and perform well on certain clustering problems compared to more traditional methods like k-means. In this problem, we will try to develop some intuition about why these approaches make sense and implement one of these algorithms.

Before beginning, we will review a few basic linear algebra concepts you may find useful for some of the problems.

- If A is a matrix, it has an eigenvector v with eigenvalue λ if $Av = \lambda v$.

- For any $m \times m$ symmetric matrix A, the *Singular Value Decomposition* of A yields a factorization of A into

$$A = USU^T$$

where U is an $m \times m$ orthogonal matrix (meaning that the columns are pairwise orthogonal) and $S = diag(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_m|)$ where the λ_i are the eigenvalues of A.

Given a set of m data points $x_1, \ldots, x_m$, the input to a spectral clustering algorithm typically consists of a matrix, A, of pairwise similarities between datapoints. A is often called the *affinity matrix*. The choice of how to measure similarity between points is one which is often left to the practitioner. A very simple affinity matrix can be constructed as follows:

$$A(i,j) = A(j,i) = \begin{cases} 1 & \text{if } d(x_i, x_j) < \Theta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $d(x_i, x_j)$ denotes Euclidean distance between points x_i and x_j.

The general idea of spectral clustering is to construct a mapping of the datapoints to an eigenspace of A with the hope that points are well separated in this eigenspace so that something simple like k-means applied to these new points will perform well.

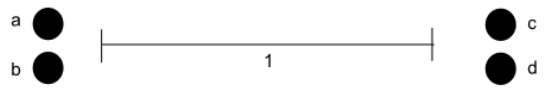


Figure 3: Simple dataset.

As an example, consider forming the affinity matrix for the dataset in Figure 3 using Equation 1 with $\Theta = 1$. Then we get the affinity matrix in Figure 4(a).

$$A = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 0 & 0 \\ b & 1 & 1 & 0 & 0 \\ c & 0 & 0 & 1 & 1 \\ d & 0 & 0 & 1 & 1 \end{array} \qquad \tilde{A} = \begin{array}{c|cccc} & a & c & b & d \\ \hline a & 1 & 0 & 1 & 0 \\ c & 0 & 1 & 0 & 1 \\ b & 1 & 0 & 1 & 0 \\ d & 0 & 1 & 0 & 1 \end{array}$$

(a) (b)

Figure 4: Affinity matrices of Figure 3 with $\Theta = 1$.

Now for this particular example, the clusters $\{a, b\}$ and $\{c, d\}$ show up as nonzero blocks in the affinity matrix. This is, of course, artificial, since we could have constructed the matrix A using any ordering of $\{a, b, c, d\}$. For example, another possibile affinity matrix for A could have been as in Figure 4(b).

The key insight here is that the eigenvectors of matrices A and $\tilde{A}$ have the same entries (just permuted). The eigenvectors with nonzero eigenvalue of A are: $e_1 = (.7, .7, 0, 0)^T$, $e_2 = (0, 0, .7, .7)^T$. And the nonzero eigenvectors of $\tilde{A}$ are: $e_1 = (.7, 0, .7, 0)^T$, $e_2 = (0, .7, 0, .7)^T$. Spectral clustering embeds the original data points in a new space by using the coordinates of these eigenvectors. Specifically, it maps the point x_i to the point $(e_1(i), e_2(i), \ldots, e_k(i))$ where $e_1, \ldots, e_k$ are the top k eigenvectors of A. We refer to this mapping as the spectral embedding. See Figure 5 for an example.

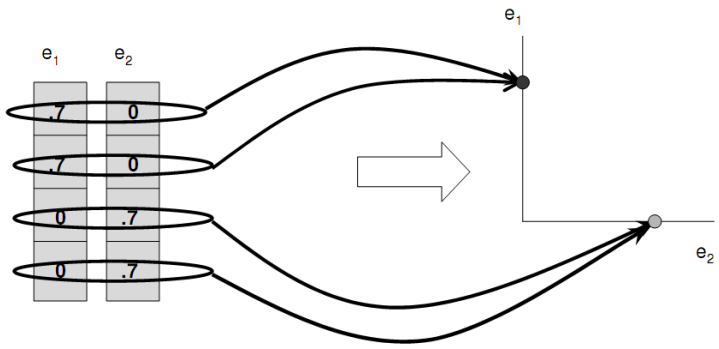


Figure 5: Using the eigenvectors of A to embed the data points. Notice that the points $\{a, b, c, d\}$ are tightly clustered in this space.

3.1 Another Simple Dataset

In this problem we will analyze the operation of one of the variants of spectral clustering methods on another simple dataset shown in Figure 6.

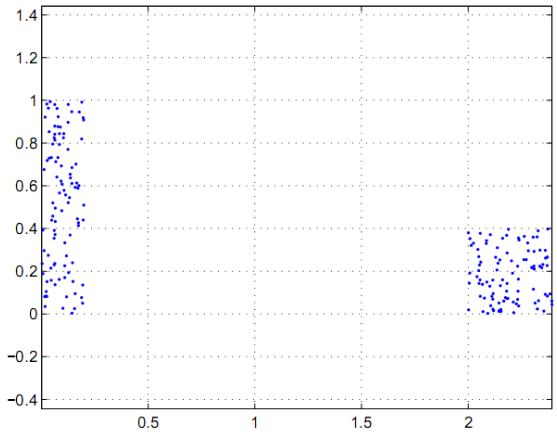


Figure 6: Dataset with rectangles.

1. **[3 points]** For the dataset in Figure 6, assume that the first cluster has m_1 points and the second one has m_2 points. If we use Equation 1 to compute affinity matrix A, what Θ value would you choose and why?
2. **[4 points]** The second step is to compute first k dominant eigenvectors of the affinity matrix, where k is the number of clusters we want to have. For the dataset in Figure 6 and the affinity matrix defined by Equation 1, is there a value of Θ for which you can analytically compute the first two eigenvalues and eigenvectors? If not, explain why not. If yes, compute and write these eigenvalues and eigenvectors down. What are the other eigenvalues? Explain briefly.
3. **[2 points]** As in Figure 5 we can now compute the spectral embedding of the data points using the k top eigenvectors. For the dataset in Figure 6 write down your best guess for the coordinates of the $k = 2$ cluster centers using the Θ that you picked in the first part.

3.2 Implementing Spectral Clustering

Frequently, the affinity matrix is constructed using the Gaussian kernel as

$$A(i,j) = exp(-\frac{d(x_i, x_j)^2}{\sigma}) \tag{2}$$

where σ is some user-specified parameter. The best that we can hope for in practice is a near block-diagonal affinity matrix. It can be shown in this case, that after projecting to the space spanned by the top k eigenvectors, points which belong to the same block are close to each other in a euclidean sense. We will not try to prove this, but using this intuition, you will implement one (of many) possible spectral clustering algorithms. This particular algorithm is described in *On Spectral Clustering: Analysis and an algorithm Andrew Y. Ng, Michael I. Jordan, Yair Weiss (2001).*

The steps of the algorithm are as follows:

- Construct an affinity matrix A using Equation 2.
- Symmetrically 'normalize' the rows and columns of A to get a matrix N such that $N(i,j) = \frac{A(i,j)}{\sqrt{d(i)d(j)}}$, where $d(i) = \sum_k A(i,k)$. In matrix form, $N = D^{-1/2}AD^{-1/2}$.
- Construct a matrix Y whose columns are the first k eigenvectors of N.
- Normalize each row of Y such that it is of unit length.
- Cluster the dataset by running k-means on the set of spectrally embedded points, where each row of Y is a data point.

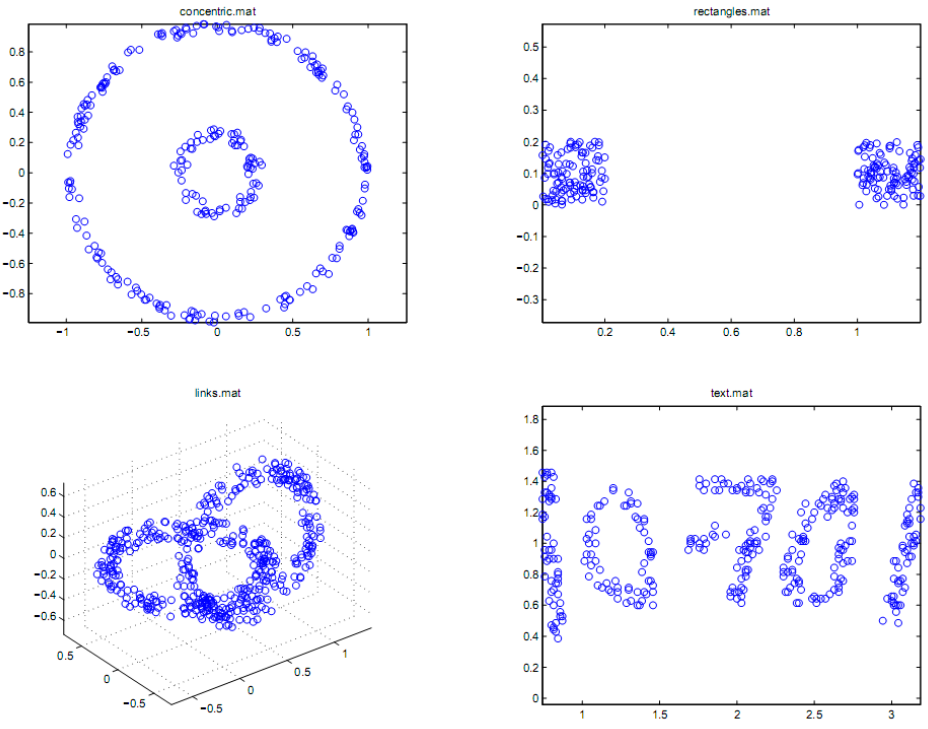


Figure 7: The four synthetic datasets.

4. **[2 points]** Run k-means on the *four* datasets provided on the class webpage and provide plots of the results. For 'text.mat', take $k = 6$. For all others use $k = 2$. Recall that k-means result is dependent on the initialization. You may try several runs with different initial centroids and give the plots for the best one you think. *Hints: For this exercise, you may use the MATLAB function 'kmeans'. A function 'plotClusters.m' has been provided to help visualize clustering results.*

5. **[5 points]** Implement the above spectral clustering algorithm and run it on the provided datasets using the same k. Plot your clustering results using $\Theta = .025, .05, .2, .5$. Also print all your code in your hard-copy submission. Soft-copy submission is not required in this assignment. *Hints: You may find the MATLAB functions 'pdist' and 'eig' helpful.*

6. **[2 points]** How do k-means and spectral clustering compare?

3.3 Theoretic analysis

In the above algorithm we make use of the matrix $N = D^{-1/2}AD^{-1/2}$. Remember that A is an affinity matrix with $a_{ij} = a_{ji}$ being a non-negative distance between points x_i and x_j. D is a dioganal matrix whose i^{th} diagonal element, d_{ii}, is the sum of A's i^{th} row. In the following you will prove several properties related to spectral clustering.

7. **[3 points]** Show that a vector $v_1 = [\sqrt{d_{11}}\sqrt{d_{22}}\dots\sqrt{d_{nn}}]^T$ is an eigenvector of N with an eigenvalue $\lambda_1 = 1$.

For the following proof, you might find the following property useful: $\lambda_1 = 1$ is in fact the largest eigenvalue of N and all the other eigenvectors (that are orthogonal to v_1) have an eigenvalue strictly smaller than 1, that is, $|\lambda_i| < 1$ for $\forall i > 1$.

Now consider $P = D^{-1}A$, where $p_{ij} = a_{ij}/d_{ii}$, which is 'kind of' the probability of transitioning from point i to point j. In other words, we normalize each row of P, so that it sums up to 1 and therefore is a valid probability transition matrix. Hence, P^t is a matrix whose $\{i, j\}^{th}$ element shows the probability of being at vertex j if started at vertex i, after t number of steps.

8. **[4 points]** Show that $P^\infty = D^{-1/2}v_1v_1^TD^{1/2}$.

   This property shows that if points are viewed as vertices in a Markov graph with transition probabilities proportional to distances between points (elements of A), then v_1 is the only eigenvector needed to compute the probability distribution over states matrix P^∞.

4 Neural Networks [TK, 20 points]

In this problem, whenever asked to give a neural network you should clearly specify the structure of the network (numbers of hidden layers and hidden units), weights between layers, and activation functions (and/or parameters therein) used in the hidden units.

4.1 [10 points] How many hidden layers?

Consider the following "XOR" like function in the two-dimensional space:

$$f(x_1, x_2) = \begin{cases} 1 & x_1, x_2 \geq 0 \text{ or } x_1, x_2 < 0 \\ -1 & \text{otherwise.} \end{cases}$$

You want to represent this function with a neural network, and decide to only use the threshold function

$$h_\theta(v) \;=\; \begin{cases} 1 & v \geq \theta, \\ -1 & \text{otherwise}, \end{cases}$$

as the activation function in the hidden units and the output unit. In the following you will show that the smallest number of hidden layers needed to represent this function is two.

1. **[3 points]** Give a neural network with two hidden layers of threshold functions that represent f.
2. **[7 points]** Show that no neural network with one hidden layer of threshold functions can represent f. For simplicity, you shall assume the number of hidden units in a hidden layer is finite, but can be arbitrarily large.
   Hint: When there is only one hidden layer, each hidden unit can be viewed as a line in the 2-d plane such that the hidden unit outputs 1 on one side of the line and -1 on the other. Consider a neighborhood of the origin such that every line (hidden unit) crossing this neighborhood passes through the origin. Can any neural network with one hidden layer represent f in this neighborhood?

4.2 [4 points] NN and SVM

Recall that the decision function in Support Vector Machine can be written as follows:

$$f(\mathbf{x}) \;=\; \sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b,$$

where $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ are training pairs of p-dimensional feature vectors and binary ($\{1, -1\}$) labels, α_i's are the dual variables, $K(\cdot, \cdot)$ is the kernel function, and b is a bias term. Consider the polynomial kernel:

$$K(\mathbf{x}, \mathbf{x}') \;:=\; (\gamma \mathbf{x}^\top \mathbf{x}' + r)^d,$$

where γ, r and d are hyper-parameters of the kernel. Give a neural net with a single hidden layer that represents the above SVM decision function with the polynomial kernel.

4.3 [6 points] NN and Boolean function

Consider the set of binary functions over d boolean (binary) variables:

$$F \;:=\; \{f : \{0,1\}^d \mapsto \{0,1\}\}.$$

Show that any function in F can be represented by a neural network with a single hidden layer. (Note: you can work with values other than $\{0, 1\}$, such as $\{1, -1\}$, as long as the function is still binary.)

5 Comparison of Machine Learning Algorithms [Jayant, 15 pts]

In this problem, you will review the important aspects of the algorithms we have learned about in class since the midterm. For every algorithm listed in the two tables on the next pages, fill out the entries under each column according to the following guidelines. Do not fill out the greyed-out cells. Turn in your completed table with your problem set.

Guidelines:

1. **Generative or Discriminative** – Choose either "generative" or "discriminative"; you may write "G" and "D" respectively to save some writing.
2. **Loss Function** – Write either the name or the form of the loss function optimized by the algorithm (e.g., "exponential loss"). For the clustering algorithms, you may alternatively write a short description of the loss function.
3. **Decision Boundary** – Describe the shape of the decision surface, e.g., "linear". If necessary, enumerate conditions under which the decision boundary has different forms.
4. **Parameter Estimation Algorithm / Prediction Algorithm** – Name or concisely describe an algorithm for estimating the parameters or predicting the value of a new instance. Your answer should fit in the provided box.
5. **Model Complexity Reduction** – Name a technique for limiting model complexity and preventing overfitting.
6. **Number of Clusters** – Choose either "predetermined" or "data-dependent"; you may write "P" and "D" to save time.
7. **Cluster Shape** – Choose either "isotropic" (i.e., spherical) or "anisotropic"; you may write "I" and "A" to save time.

Learning Method	Generative or Discriminative?	Loss Function	Parameter Estimation Algorithm	Prediction Algorithm	Model Complexity Reduction
Bayes Nets					
Hidden Markov Models					
Neural Networks					

Table 5: Comparison of Classification Algorithms

Learning Method	Loss Function	Number of clusters: Predetermined or Data-dependent	Cluster shape: isotropic or anisotropic?	Parameter Estimation Algorithm
K-means				K-means
Gaussian Mixture Models (identity covariance)				
Single-Link Hierarchical Clustering				
Spectral Clustering				

Table 6: Comparison of Clustering Algorithms