Machine Learning Midterm Answers

This exam is open book. You may bring in your homework, class notes and textbooks to help you. You will have 1 hour and 15 minutes. Write all answers in the blue books provided. Please make sure YOUR NAME is on each of your blue books. Square brackets [] denote the points for a question.

1. **Linear Algebra**
   (a) [10] Show that the *Woodbury* indentity is true:
   $$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$
   (b) [15] Show that if a matrix A is positive definite, its eigenvalues must be positive also.

Answer to part a:
Premultiply by $(A+BCD)$ and cancel identity matricies on both sides, then rearrange :

$$BCDA^{-1} = B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}+BCDA^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}$$

Cancel DA^{-1},

$$BC = B(C^{-1} + DA^{-1}B)^{-1} + BCDA^{-1}B(C^{-1} + DA^{-1}B)^{-1}$$

Postmultiply by $(C^{-1} + DA^{-1}B)^{-1}$,

$$BC(C^{-1} + DA^{-1}B) = B + BCDA^{-1}B$$

Answer to part b:
Eigenvalue equation is:

$$Av = \lambda v$$

Premultiply by v^T

$$v^T Av = \lambda v^T v$$

Both $v^T Av$ and $v^T v$ are positive, so λ must be positive also.

2. **Entropy**

   (a) [15] Show that where M is the number of states of $\{x_i\}$, the entropy of $p(x_i)$ is maximized by $p(x_i) = \frac{1}{M}$ and $H = \ln M$. Use the Lagrange multiplier technique.

   (b) [10] What is the Kullback-Liebler distance and why is it useful?

Answer to part a:

$$J = -\sum_{i=1}^{M} p(x_i) \log p(x_i) + \lambda(\sum_{i=1}^{M} p(x_i) - 1) \tag{1}$$

$$\frac{\partial J}{\partial p(x_i)} = \log p(x_i) - 1 + \lambda = 0 \tag{2}$$

So

$$\lambda = 1 - \log p(x_i)$$

and since this must work for *all* the $p(x_i)$ and there is only one λ then all the $p(x_i)$ must be equal.

Answer to part b:

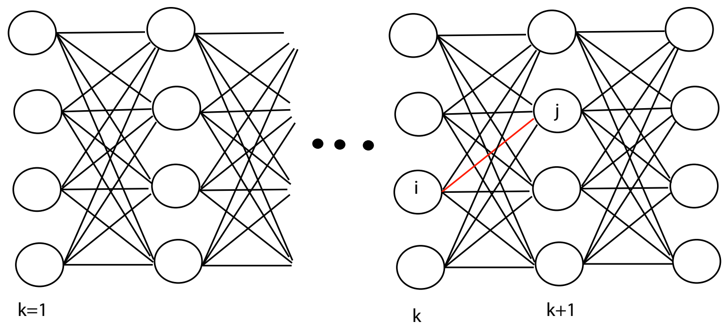
The K-L divergence, or relative entropy is a way of measuring the 'distance' between two distributions, since

$$\sum_k p(x_k) \log \frac{p(x_k)}{q(x_k)} \geq 0$$

This is very useful since in a very large number of applications we want to approximate and ideal distribution $p(x_k)$ with a computable distribution$q(x_k)$.

3. **Optimization**

(a) [20]The following graph represents probabilities for transiting from one state to another in stages. P_{ijk} represents the probability of transiting



from node i to node j at stage k. Specify a dynamic programing algorithm that calculates *the most probable path* through this graph from any node to the end. What is your recursion equation?

(b) [5]In general, when might the dynamic programming method be used over the Hamiltonian method?

Answer to part a:
Let $V(i, k)$ be the most probable path from the ith node in the kth stage until the end - lets call the last stage K. Then $V_{iK} = 0$ and

$$V(i, k-1) = \max_j \{P_{ijk} + V(j, k)\}$$

The most probable paths are given by $V(i, 1)$.

Answer to part b (one of several):
Use DP when it is feasible to represent the state space in discrete form.

4. **Support Vector Machines**

The XOR problem is given by:

x_1	x_2	desired output d
-1	-1	-1
+1	-1	+1
-1	+1	+1
+1	+1	-1

(a) [20] Will the following $\psi(\mathbf{x})$ solve the XOR problem? Show all steps.

$$\psi(\mathbf{x}) = \begin{bmatrix} x_1^2 - x_2^2 \\ x_1 x_2 \\ x_1^2 + x_2^2 \end{bmatrix}$$

(b) [5] What is the Perceptron limitation and how to SVMs deal with it?

Answer to part a:

$$\psi(-1,-1) = \begin{pmatrix} 0 & 1 & 2 \end{pmatrix},\ \psi(-1,1) = \begin{pmatrix} 0 & -1 & 2 \end{pmatrix}$$

$$K = \begin{bmatrix} 5 & 3 & 3 & 3 \\ 3 & 5 & 3 & 3 \\ 3 & 3 & 5 & 3 \\ 3 & 3 & 3 & 5 \end{bmatrix}$$

$$Q(\lambda) = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 - \frac{1}{2}\{5\lambda_1^2 - 6\lambda_1\lambda_2 - 6\lambda_1\lambda_3 + 6\lambda_1\lambda_4 + \cdots\}$$

$$\frac{\partial Q}{\partial \lambda_1} = 0 = 1 - 5\lambda_1 + 3\lambda_2 + 3\lambda_3 + -3\lambda_4$$

From symmetry,

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \frac{1}{2}$$

So plugging in, $Q(\lambda) = \frac{5}{2}$ and $||\mathbf{w}_o|| = \sqrt{5}$. And

$$\mathbf{w}_o = \sqrt{5}\begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix}$$

and finally, $\mathbf{w}_o^T \psi(\mathbf{x}) = 4\sqrt{5} x_1 x_2$.

Answer to part b:
The higher dimensional space of SVMs increases class separations.