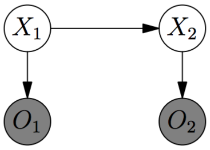
Machine Learning M146 UCLA Spring quarter 2017-2018
Prof. Suhas Diggavi Handout # 24, Friday, Jun 8th 2018

FINAL PRACTICE PROBLEMS # 1

Friday, 8th June 2018

Problem 1 (Hidden Markov Models)

Consider the following Hidden Markov Model.



X_1	$\Pr(X_1)$
0	0.3
1	0.7

X_t	X_{t+1}	$\Pr(X_{t+1}\|X_t)$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

X_t	O_t	$\Pr(O_t\|X_t)$
0	A	0.9
0	B	0.1
1	A	0.5
1	B	0.5

Suppose that $O_1 = A$ and $O_2 = B$ is observed.

(a) What is the probability of $P(O_1 = A, O_2 = B, X_1 = 0, X_2 = 1)$?

(b) What is the most likely assignment for X_1 and X_2?

(c) **True/False** Based on the independent assumptions in HMM, the random variable O_1 is independent of the random variable X_2. Justify your answer.

Problem 2 (EM algorithm and Gaussian Mixture Model)

Consider a two-component Gaussian mixture model for univariate data (i.e. $x \in \mathbb{R}$), in which the probability density for an observation, x, is

$$\frac{1}{2}\mathcal{N}(x|\mu, 1) + \frac{1}{2}\mathcal{N}(x|\mu, 2^2)$$

Here, $\mathcal{N}(x|\mu, \sigma^2)$ denotes the density for x under a univariate normal distribution with mean μ and variance σ^2. Notice that mixing proportions are equal for this mixture model, that the two components have the same mean, and that the standard deviations of the two components are fixed at 1 and 2. There is only one model parameter, μ.

Suppose we wish to estimate the μ parameter by maximum likelihood using the EM algorithm. Answer the following questions regarding how the E step and M step of this algorithm operate, if we have the three data points below:

$$4.0, 4.6, 2.0$$

Here is a table of standard normal probability densities that you may find useful:

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
$N(x\|0,1)$	.40	.40	.39	.38	.37	.35	.33	.31	.29	.27	.24	.22	.19	.17	.15	.13	.11	.09	.08	.07	.05

(a) Find the posterior probabilities that will be computed in the E step if the model parameter estimates from the previous M step are $\mu = 4$,$\sigma_1 = 1$ and $\sigma_2 = 2$. Since the probabilities for the two components must add to one, it is enough to give $r_{i1} = P(\text{component } 1|x_i)$ for $i = 1, 2, 3$. You can leave your answer in terms of a fraction.
Hint: Note that the normal density function with mean μ and variance σ^2 is

$$\mathcal{N}(x|\mu, \sigma^2) = (1/\sigma)\mathcal{N}\left(\frac{x-\mu}{\sigma}\middle|\, 0, 1\right)$$

(b) Using the probabilities that you computed in part (a), find the estimate for μ that will be found in the next M step. Recall that the M step maximizes the expected value of the log of the probability density for x_1, x_2, x_3 and the unknown component indicators, with the expectation taken with respect to the distribution for the component indicators found in the previous E step.

Problem 3 (PCA)

Take four data points $x_1 = (-1, 1), x_2 = (2, 2), x_3 = (-2, -2)$ and $x_4 = (1, -1)$ in $\mathbb{R}^2$ euclidean space.

(a) Find the first principal component vector.

(b) Project the data points onto the subspace of the principal component chosen above. Find the new coordinates in the subspace spanned by the principal component and the variance of the projected data.

(c) Find the representation of the projections obtained in the original 2-d space and compute the reconstruction error. *Note:* Mean reconstruction error is the average squared distance of original points from their estimates.

(d) Remove x_2 and x_3 from the data set and suppose that the steps undertaken in (a), (b) and (c) are repeated on the remaining points. What is the reconstruction error now?

Hint: You need not necessarily solve an SVD or eigen decomposition for solving this question.

Problem 4 (Kernels and SVM)

(a) Properties of Kernels

i. Given n training examples $\{x_i\}_{i=1}^n$, the kernel matrix $\mathbf{A}$ is an $n \times n$ square matrix, where $\mathbf{A}(i, j) = K(x_i, x_j) = \Phi(x_i)^T\Phi(x_j)$. Prove that the kernel matrix is symmetric (i.e, $A_{i,j} = A_{j,i}$).
hints: Your proof will not be longer than 2 or 3 lines.

ii. Prove that the kernel matrix $\mathbf{A}$ is positive semi-definite.
hints: (1) Remember that an $n \times n$ matrix $\mathbf{A}$ is positive semi-definite if and only if for any n dimensional vector $\mathbf{v} \neq \mathbf{0}$, we have $\mathbf{v}^T \mathbf{A} \mathbf{v} \geq 0$. (2) Consider a matrix $\mathbf{B} = [\Phi(x_1), \cdots, \Phi(x_n)]$ and use it to prove A is positive semi-definite.

(b) Given a dataset $D = \{x_i, y_i\}, x_i \in \mathbb{R}^k, y_i = \{-1, +1\}, 1 \leq i \leq N$.
A hard SVM solves the following formulation

$$\min_{w,b} \quad \frac{1}{2} w^T w \qquad \text{s.t} \quad \forall i, y_i(w^T x_i + b) \geq 1, \tag{1}$$

and soft SVM solves

$$\min_{w,\xi_i,b} \quad \frac{1}{2} w^T w + C \sum_i \xi_i \qquad \text{s.t} \quad \forall i, y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \forall i, \xi_i \geq 0 \tag{2}$$

i. Complete:
If $C =$ ________________, soft SVM will behave exactly as hard SVM.
In order to reduce over-fitting, one should ________________ (decrease or increase) the value of C.

ii. Show that when $C = 0$, the soft SVM returns a trivial solution and cannot be a good classification model.

iii. **True/False** The slack variable ξ_i in soft SVM for a data point x_i always takes the value 0 if the data point is correctly classified by the hyper-plane. Explain your answer.

iv. **True/False** The optimal weight vector w can be calculated as a linear combination of the training data points. Explain your answer. [You do not to prove this.]

v. We are given the dataset in Figure 1 below, where the positive examples are represented as black circles and negative points as white squares. (The same data is also provided in Table 1 for your convenience). Recall that the equation of the separating hyperplane is $\hat{y} = \mathbf{w}^T \mathbf{x} + b$.

i. Write down the parameters for the learned linear decision function.

$W = (w_1, w_2) =$ ________________. $b =$ ________________

ii. Circle all support vectors in Figure 1.

Problem 5 (Kernelized logistic regression)

In this problem, we explore how logistic regression can be kernelized.

We are given a set of N training examples, $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ where $\mathbf{x}_n \in \mathbb{R}^D$, $y_n \in \{0, 1\}$. We learn a logistic regression model $h_\theta(\mathbf{x}) = \sigma(\theta^T \mathbf{x})$ using gradient descent where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

In iteration t of gradient descent, we update $\theta \leftarrow \theta - \eta \sum_n \epsilon_n \mathbf{x}_n$ where $\epsilon_n = h_\theta(\mathbf{x}_n) - y_n$ is the error for the n^{th} training sample, and η is the step size or learning rate.

index	x_1	x_2	y
1	0	0	−
2	0	-4	−
3	-1	-1	−
4	-2	-2	−
5	3	0	+
6	0	3	+
7	1	1	+
8	3	-1	+

Table 1: The dataset S

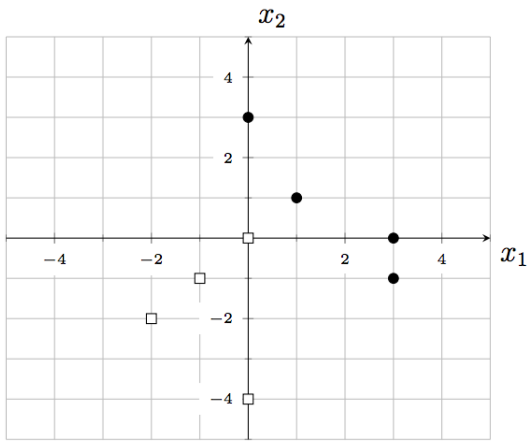


Figure 1: Linear SVM

We map $\mathbf{x}$ to $\phi(\mathbf{x})$ and we would like to learn a logistic regression model $\sigma(\theta^T\phi(\mathbf{x}))$ while only working with the inner products $\phi^T(\mathbf{x})\phi(\mathbf{x}')$.

(a) Assume we initialize θ to zero in the gradient descent algorithm, *i.e.*, $\theta \leftarrow \mathbf{0}$. Show that at the end of every iteration of gradient descent, θ is always a linear combination of the training samples: $\theta = \sum_{n=1}^{N} \alpha_n \phi(\mathbf{x}_n)$.

(b) Using the above result, show how we can write $h_\theta(\mathbf{x})$ to make a prediction on a new input $\phi(\mathbf{x})$ by only using inner products of the form $\phi(\mathbf{x})^T\phi(\mathbf{x}')$.

(c) The final step in kernelization is to show that we do not need to explicitly store θ. Instead from part (a), we can implicitly update θ by updating α_n. Show how α_n is intialized and how it is updated.

Problem 6 (Linear Regression)

In this problem, you will examine the behavior of a certain type of input perturbation for a probabilistic linear regression setting.
Consider the following general generative model for regression:

- $\boldsymbol{x} \sim p(\boldsymbol{x})$ is a distribution over input vectors $\boldsymbol{x} \in \mathbb{R}^d$
- $y|\boldsymbol{x} \sim p(y|\boldsymbol{x})$ is a distribution over output scalars $y \in \mathbb{R}$ given $\boldsymbol{x}$

Assume that the relationship between y and $\boldsymbol{x}$ is well modeled by a linear function $y = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}$, where $\boldsymbol{w} \in \mathbb{R}^d$, so that in the infinite dataset limit, the objective to be minimized for the regression problem is

$$\mathcal{L}_0(\boldsymbol{w}) = \mathbb{E}\left[(\boldsymbol{w}^T\boldsymbol{x} - y)^2\right]$$

where $\mathbb{E}$ stands for expectation. Now suppose the inputs are perturbed by zero-mean Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \lambda\boldsymbol{I})$, which is independent of the training data. The new objective is

$$\mathcal{L}(\boldsymbol{w}) = \mathbb{E}\left[(\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) - y)^2\right]$$

(a) Compute and simplify $\mathcal{L}(\boldsymbol{w})$. Show all your work in detail, and write your answer in terms of $\mathcal{L}_0$.

(b) Is there a relationship between this particular type of input perturbation and some type of regularization? If so, what kind of regularizer is involved?