CS771: Machine learning: tools, techniques and applications

End-semester exam

Time: 3 hours 27-Nov.-2013
Max marks: 120

1. *Answer all 5 questions. The question paper has 4 pages.*
2. *Please be* **precise** *and* **brief** *in your answers.*
3. **Please answer all parts of a question together do not scatter the parts of a question all over the answer script.**
4. *You can consult* **only your own handwritten notes**. *Photocopies, printouts and any electronic gadgets are not allowed.*

1. This question has 8 parts. It contains short answer questions from the whole syllabus. Please show calculations or give justifications where necessary. Simply writing the final answer may not fetch you credit.

   (a) There are two common ways to estimate classifier performance i) do n-fold cross validation (typically n is 5 or 10) or ii) measure performance on a test set that is distinct from the learning and validation (if used) sets. When will you use i) and when ii) and why?

   Answer:
   Cross validation is useful when the total amount of data is small so setting aside dome data as a test set means learning set $\mathbf{L}$ becomes even smaller. An alternative is to use multiple randomly chosen test sets which is similar to cross-validation.

   If the data set is large we can afford to set aside a separate test set. Also, if we do cross-validation here the different learning sets will have a large degree of overlap - for example in 5-fold cross validation 60% data will be common so the models built for different folds tend to be very similar. So, testing on a single separate test set is usually adequate.

   (b) Given d-dimensional feature vectors whose distribution is modelled by a mixture of m multivariate Gaussians how many unknowns in all have to be estimated if we are doing a maximum likelihood estimate of the parameters of the distribution?

   Answer:
   For one Gaussian we have d means and $\frac{d(d+1)}{2}$ covariance matrix entries. In addition we have m mixing weights α_i but with the constraint that $\sum_{i=1}^{n} \alpha_1 = 1$ so only $m-1$ independent mixing weights. This gives a total of: $m(d + \frac{d(d+1)}{2}) + (m-1)$ parameters in all.

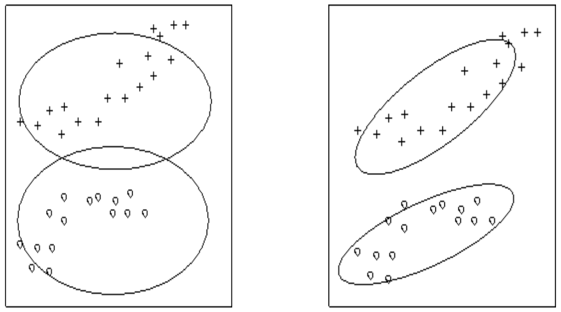
(c) Let $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ be three distinct vectors in $\mathbb{R}^d$. And let $\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{z}_3$ be three fixed vectors in $\mathbb{R}^D$. Does there exist a valid Mercer kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{z}_i, \mathbf{z}_j \rangle$?

Answer:

Yes.

Let ϕ be the mapping such that $\phi(\mathbf{x}_1) = \mathbf{z}_1$, $\phi(\mathbf{x}_2) = \mathbf{z}_2$, $\phi(\mathbf{x}_3) = \mathbf{z}_3$ and $\phi(\mathbf{x}) = \mathbf{0}, \forall \mathbf{x} \neq \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3$. Any K such that $K(u, v) = \langle \phi(u), \phi(v) \rangle$ will be a Mercer kernel since the Gram matrix will be psd.

(d) Given the data distribution shown below containing $+$ and $\circ$ vectors the models by two students A and B are shown in the figures below labelled A and B respectively. Both model the $+$ and $\circ$ by separate Gaussians. Which one is better and why? Draw a third mixture model that could be better than both A and B.



A B

Answer:

A has axes aligned Gaussians so co-variance matrix does not contain non-diagonal terms. B's Gaussians are not axes aligned so it allows non-zero non-diagonal terms in the co-variance matrix. So, B's model is better for the given data set.

Probably a better model is a mixture of two Gaussian for both + and $\circ$ classes. The two legs or parts in both classes can be modelled by separate Gaussians.

(e) Suppose the SVM formulation with slack variables is applied to a learning set $\mathbf{L}$ that is separable then to minimize the objective function, $\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_i \xi_i$, we can set all ξ_is to 0 (since $\mathbf{L}$ is separable) thus making $\mathbf{w}$ independent of C. So, the value of C does not matter provided $C \neq 0$.

Is the above reasoning correct or wrong? Justify your answer.

Answer:

Wrong.

SVM tries to maximize margin and does not to try to find a perfectly separating hyperplane. If outliers are present then the separating hyperplane may misclassify the outlier to maximize the margin even if the set is separable by using non-zero ξ_i.

(f) In the *bagging* approach to using an ensemble of classifiers (e.g. random forest) the base classifier (e.g. decision tree) must satisfy a necessary condition for the method to work. What is this condition? Briefly say why this condition is necessary.

Answer:
In bagging the different **L**s have a large degree of overlap (average of 63%). So a necessary condition for the ensemble method to work well is that the base classifer should be unstable - that is small changes in **L** should produce very different models. If the base learner is not stable then we will get many similar models and the ensemble will not really have enough diversity for the ensemble method to work well.

(g) Let **R** be the set of retrieved vectors when we search a database of vectors **D** using a query **Q**. Let **L** be the set of vectors relevant to query **Q** in **D**. Draw the Venn diagrams for the cases when precision is 1 and when recall is 1.

Answer:
If all vectors in the set of retrieved vectors are relevant (i.e. no false positives) then precision is 1. So, $R \subset L$.

If all relevant vectors are part of the retrieved set then recall is 1. So converse of above that is $L \subset R$.

(h) Two dimension reduction techniques are i) feature selection where we select features that are "important" or "useful" to the task (you did this in assignment 1) from the ambient set of features and ii) principal components analysis (PCA). What is the difference between the two and which, if any, should we use when we know that the real dimension of the data is smaller than the ambient dimension.

Answer:
PCA linearly combines existing features to create new features while selection simply knocks out features. PCA is more general. If the true dimension is known to be less it is better to use PCA since chances are the reduced dimension contains combinations (linear) of existing dimensions rather than simply a subset of the original features.

[5x8=40]

2. (a) You are given that $P(X|Y) = \frac{2}{3}$, $P(X| \sim Y) = \frac{1}{3}$ and $P(Y) = \frac{1}{3}$. Is it possible to find $P(Y|X)$? If yes find the value; if no then say why not.

Answer:
By Bayes rule $P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$. We aleady have values for the numerator. We can calculate $P(X)$ by

$$\begin{aligned} P(X) &= P(X|Y)P(Y) + P(X|\sim Y)P(\sim Y) \\ &= \frac{2}{3}\frac{1}{3} + \frac{1}{3}\frac{2}{3} \\ &= \frac{4}{9} \end{aligned}$$

So, $P(Y|X) = \frac{2}{3}\frac{1}{3}/\frac{4}{9} = \frac{1}{2}$.

(b) The unconditional probability of getting diabetes (label ω_1) is $P(\omega_1) = 0.2$ and not getting it (label ω_0) is $P(\omega_0) = 0.8$. The following probabilities are observed for having diabetes based on three categories of *weight* (Overweight, Normal, Underweight).

	Overweight	Normal	Underweight
$P(weight\|\omega_1)$	0.5	0.3	0.2
$P(weight\|\omega_0)$	0.1	0.3	0.6

i. Calculate the conditional probability $P(\omega_i|\mathbf{x})$, $i \in \{0, 1\}$ for each weight type.

Answer:

$$\begin{aligned} P(overweight) &= P(overweight|\omega_1)P(\omega_1) + P(overweight|\omega_0)P(\omega_0) \\ &= 0.5 \times 0.2 + 0.1 \times 0.8 \\ &= 0.18 \end{aligned}$$

$$\begin{aligned} P(normal) &= 0.3 \times 0.2 + 0.3 \times 0.8 \\ &= 0.30 \end{aligned}$$

$$\begin{aligned} P(under) &= 0.2 \times 0.2 + 0.6 \times 0.8 \\ &= 0.52 \end{aligned}$$

Bayes rule gives us the following table.

	Overweight	Normal	Underweight
$P(weight\|\omega_1)$	$\frac{0.1}{0.18}$	$\frac{0.06}{0.30}$	$\frac{0.04}{0.52}$
$P(weight\|\omega_0)$	$\frac{0.08}{0.18}$	$\frac{0.24}{0.30}$	$\frac{0.48}{0.52}$

ii. What is the Bayes rule for predicting diabetes given a person's weight?

Answer:
The Bayes decision rule can be written down directly from the table above.
If (overweight) give label ω_1 (diabetes) elseif (normal or underweight) give label ω_0 (no diabetes).

iii. Find the unconditional Bayes error rate.

Answer:

$$\begin{aligned}
Error &= \sum_{wt \in \{\text{over,normal,under}\}} P(wt)\ min[P(\omega_1|wt), P(\omega_0|wt)] \\
&= 0.18 \times \frac{0.08}{0.18} + 0.30 \times \frac{0.06}{0.30} + 0.52 \times \frac{0.04}{0.52} \\
&= 0.08 + 0.06 + 0.04 \\
&= 0.18
\end{aligned}$$

That is 18%.

[5,(6,4,5)=20]

3. (a) Instead of estimating the parameters of an assumed probability distribution and then applying Bayes rule we can directly construct the discriminant funtion in terms of the parameters of the probability distribution. If we write the discriminant function $g_{\omega_j}(\mathbf{x})$ in terms of log probabilities and assign the class label ω_j to $\mathbf{x}$ if $g_{\omega_j}(\mathbf{x}) > g_{\omega_i}(\mathbf{x})$, $j \neq i$ what is the exact form of $g_{\omega_j}(\mathbf{x})$ if the assumed class conditional probability distribution is a multi-variate normal distribution $p(\mathbf{x}|\omega_j) \sim N(\overline{\mu}, \overline{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\overline{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\overline{\mu})^T \overline{\Sigma}^{-1}(\mathbf{x}-\overline{\mu})}$ where d is the dimension of $\mathbf{x}$ and $|\overline{\Sigma}|$ is the determinant of the covariance matrix $\overline{\Sigma}$.

Answer:
We have $g_{\omega_j}(\mathbf{x}) = \frac{p(\omega_j|\mathbf{x})P(\omega_j)}{P(\mathbf{x})}$.
If we work with log probabilities then:

$$\begin{aligned}
&= log(P(\omega_j)) + log(p(\omega_j|\mathbf{x})) - log(P(\mathbf{x})) \\
&= log(P(\omega_j)) + [-\frac{1}{2}(\mathbf{x} - \mu_\mathbf{j})^T \mathbf{\Sigma_j}^{-1}(\mathbf{x} - \mu_\mathbf{j}) - \frac{d}{2} log(2\pi) - \frac{1}{2} log(|\mathbf{\Sigma_j}|)] - log(P(\mathbf{x}))
\end{aligned}$$

Since $P(\mathbf{x})$ and $\frac{d}{2}log(2\pi)$ will be same in all discriminant functions we can drop them getting:

$$g_{\omega_j}(\mathbf{x}) = log(P(\omega_j)) - \frac{1}{2}(\mathbf{x} - \mu_\mathbf{j})^T \mathbf{\Sigma_j}^{-1}(\mathbf{x} - \mu_\mathbf{j}) - \frac{1}{2} log(|\mathbf{\Sigma_j}|)$$

(b) If there are only two classes what kind of separation boundaries do we expect between classes in part (a) above.

Answer:
The equation for $g_{\omega_j}(\mathbf{x})$ is quadratic in $\mathbf{x}$. So, discrimination boundaries will be quadratic functions/surfaces. In 2-D conics - that is algebraic curves of degree 2.

(c) In the k-nearest neighbour based approach to calculating the density estimate $\hat{p}(\mathbf{x}) = \frac{k}{nV}$, we choose k thereby fixing $\frac{k}{n}$ ($n = \sum_{i=1}^{C} n_i$, is the number of vectors in the sample) and then build an expanding ball with $\mathbf{x}$ as centre till we have k points from the sample in the ball.

For a multi-class classification problem with C classes use the above method of estimating densities together with the Bayes decision rule to derive the k-nearest neighbour decision rule for assigning class labels to $\mathbf{x}$.

[Note: the k-nearest neighbour decision rule is: assign label ω_i to $\mathbf{x}$ if $k_i \geq k_j$, $i \neq j$, i.e. amongst the k-nearest neighbours of $\mathbf{x}$ assign the label that is most frequent with ties broken arbitrarily.]

Answer:
We have $k = \sum_{i=1}^{C} k_i$ where k_i is the number vectors with label ω_i in the k-nearest neighbours. So, estimate $\hat{p}(\mathbf{x}|\omega_i) = \frac{k_i}{n_i V}$ where n_i is the total number of samples labelled ω_i in $\mathbf{L}$ where $\sum_{i=1}^{C} n_i = n$. So, we also have $\hat{P}(\omega_i) = \frac{n_i}{n}$.
The decision rule is:
Give label ω_i if $p(\omega_i|\mathbf{x}) \geq p(\omega_j|\mathbf{x}) \quad i \neq j$.
Using estimates it becomes:

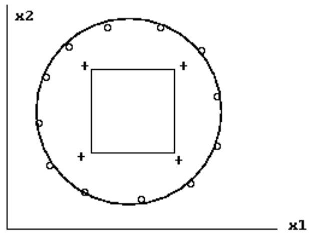
$$\hat{p}(\mathbf{x}|\omega_i)\hat{P}(\omega_i) \geq \hat{p}(\mathbf{x}|\omega_j)\hat{P}(\omega_j) \quad i \neq j$$

Substituting for values in the expression above:

$$\frac{k_i}{n_i V} \times \frac{n_i}{n} \geq \frac{k_j}{n_j V} \times \frac{n_j}{n}$$
$$k_i \geq k_j \quad i \neq j$$

(d) You are given the two-class data distribution below where the $\circ$ vectors are on the circle and the + are at the four corners of the square. Vector $\mathbf{x} = (x_1, x_2)$ is 2-dimensional. Assuming the centroid of the circle and the square coincide what label (+ or $\circ$) will you give to $\mathbf{x}$ if $\mathbf{x}$ is the centroid. Justify your answer.

Answer:
If all we have is the above data then a blind application of nearest neighbour can mislead. Note that while the variance of + is smaller than $\circ$ the apriori probability of a vector



being a ∘ is significantly higher (3 times). So, a better decision is to label **x** as ∘ rather than +.

[5,3,8,4=20]

4. (a) You are given: if $K(\mathbf{x}_i, \mathbf{x}_j)$ is a valid Mercer kernel then $e^{K(\mathbf{x}_i,\mathbf{x}_j)}$ is a valid kernel. Argue that the Gaussian kernel $e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{\sigma^2}}$ is a valid kernel, where $\sigma^2 > 0$ is a positive constant. Carefully, justify each step in your argument.
[Hint: Expand $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ and then use results for combining kernels.]

Answer:

$$\begin{aligned} e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2} &= e^{-\frac{1}{\sigma^2}(\|\mathbf{x}_i\|^2+\|\mathbf{x}_j\|^2-\mathbf{x}_i^T\mathbf{x}_j} \\ &= e^{-\frac{\|\mathbf{x}_i\|^2}{\sigma^2}} e^{-\frac{\|\mathbf{x}_j\|^2}{\sigma^2}} e^{\frac{2\mathbf{x}_i^T\mathbf{x}_j}{\sigma^2}} \end{aligned}$$

If $f : X \to \mathbb{R}$ then $K(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i)f(\mathbf{x}_j)$ is a kernel. So, the product of the first two terms is a kernel, say K_1. The third term is a kernel since: an inner product of two vectors is a kernel; multiplying this kernel by a positive constant yields another kernel and exponentiating this gives yet another kernel using the result given - let this be K_2. Now the product of two kernels is a kernel so K_1K_2 is a kernel which completes the proof.
[Note that if K is a kernel then cK is a kernel only for $c \in \mathbb{R}^+$. A common error is to assume all linear combinations are kernels.]

(b) Consider the RBF kernel $K_R(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{1}{2}\|\mathbf{x}_i-\mathbf{x}_j\|^2}$. Argue that the squared Euclidean distance between two vectors in the mapped space using the non-linear mapping $\phi(\mathbf{x})$ is < 2 - that is $\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 < 2$. Note that $\langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle = K(\mathbf{x}_i, \mathbf{x}_j)$.

Answer:

$$\begin{aligned} \|\phi(\mathbf{x}_i)\phi(-\mathbf{x}_j\|^2 &= (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))^T(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) \\ &= \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_i) + \phi(\mathbf{x}_j)^T\phi(\mathbf{x}_j) - 2\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) \\ &= K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned}$$

Since $\|.\| \geq 0$ and $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{1}{2}\|\mathbf{x}_i-\mathbf{x}_j\|^2}$ we can infer that $0 < K(\mathbf{x}_i, \mathbf{x}_j) \leq 1$. So $\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = 1 + 1 - 2K(\mathbf{x}_i, \mathbf{x}_j) \Rightarrow < 2$.

(c) The SVM decision rule for a vector $\mathbf{x}$ is given by the sign of the decision function: $g(\mathbf{x}) = \sum_{\mathbf{x}_i \in SV} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0$, where SV is the set of support vectors and instead of using extended vectors we have used normal vectors and written w_0 explicitly. Now consider a separable learning set $\mathbf{L}$ and a test vector $\mathbf{x}_{outlier}$ that is far from all the support vectors. Show that $g(\mathbf{x}_{outlier}) \approx w_0$.

Answer:
$\mathbf{x}_{outlier}$ is far from $\mathbf{x} \in SV \Rightarrow \|\mathbf{x}_{outlier} - \mathbf{x}\| >> 0$. Using the RBF kernel this means $K(\mathbf{x}_{outlier}, \mathbf{x}) \approxeq 0$. Consequently, $\sum_{\mathbf{x}_i \in SV} \alpha_i y_i K(\mathbf{x}_{outlier}, \mathbf{x}_i) \approxeq 0$ since ever term is $\approxeq 0$ and number of SVs is finite. This in turn implies that $\sum_{\mathbf{x}_i \in SV} K(\mathbf{x}_{outlier}, \mathbf{x}_i) + w_0 \approxeq w_0$ thus completing the proof.

[7,6,7=20]

5. (a) Let $\mathbf{L} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \sim U(-\theta, \theta)$ where $U(-\theta, \theta)$ is the uniform distribution defined by:

$$p(x) = \begin{cases} 0 & x < -\theta \\ \frac{1}{2\theta} & -\theta \leq x \leq \theta \\ 0 & x > \theta \end{cases}$$

What is the MLE estimate $\hat{\theta}$ for the parameter θ? Justify your answer.

Answer:
Since the $x_i \in \mathbf{L}$ are iid the likelihood is:

$$\mathcal{L}(L; \theta) = \prod_{i=1}^{n} p(x_i|\theta)$$

and the estimate is:

$$\begin{aligned} \hat{\theta} &= \underset{\theta}{argmax}\ \mathcal{L}(\mathbf{L}; \theta) \\ &= \underset{\theta}{argmax}\ \prod_{i=1}^{n} p(x_i|\theta) \end{aligned}$$

If we choose a value for θ such that for some i either $x_i > \theta$ or $x_i < -\theta$ then $p(x_i|\theta)$ is 0 and so is $\mathcal{L}(\mathbf{L}; \theta)$. The only way to ensure that every $p(x_i|\theta) \neq 0$ is to choose $\hat{\theta} \geq max(|x_1|, \ldots, |x_n|)$ so that no $p(x_i|\theta)$ is 0. Clearly, the value for which $\hat{\theta}$ will be largest will be when $\frac{1}{2\theta}$ is largest which means $\hat{\theta} = max(x_i, \ldots, x_n)$. A larger value of θ will clearly reduce $p(x_i|\theta)$ and thereby $\mathcal{L}$.

(b) In the Adaboost algorithm consider the weighted training error e_t over the training set for the t^{th} weak classifier. Which is likely to be larger e_t or e_{t+1}? Why?

Answer:

Normally, $e_{t+1} > e_t$. The weights of repeatedly misclassified vectors will keep increasing by $\frac{1-e_t)}{e_t}$ in every round. Also, these hard to classify vectors will be much more likely to be present in **L** since it is constructed by drawing vectors with replacement from the orginal **L** according to a probability distribution given by the weights.

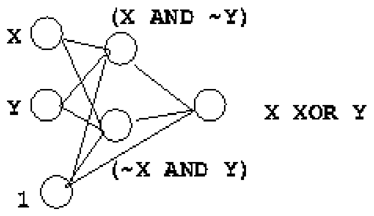
(c) We know that a 2-layer neural network cannot implement the XOR function. Using the network architecture below and the sigmoid activation function

$$y = \frac{1}{1 + e^{-(w_0 + w_1 x + w_2 y)}}$$

find the connection weights only from the set $\{10, -10, 100, -100\}$ such that the XOR function X XOR Y can be implemented. Use:

$$X \text{ XOR } Y = \begin{cases} > \frac{1}{2} & X \neq Y \\ < \frac{1}{2} & \text{else} \end{cases}$$

The meanings of the hidden nodes are also given. The bias node gives a constant output of 1. In your answer draw the neural network and write the weights on the connections.



Answer:

We can infer the weights based on the following observations:

- Because of the sigmoid activation function if the exponent of e is negative then $y \approx 0$, therefore $y < \frac{1}{2}$ and if it is positive then $y \approx 1$ making it $> \frac{1}{2}$. This is the key observation.
- The output node is an OR of the two hidden nodes. When $X \neq Y$ exactly one of the hidden nodes is 1 and we expect XOR to be one so the two hidden-to-output weights must be equal and positive and larger than the bias in absolute value so that exponent of e is positive. Also when $X = Y$ both hidden nodes should be 0 and XOR should also be 0 so the bias should be negative. So, this means the hidden-to-output weights are 100 and the bias-to-output weight is -10.

We can make a similar argument for the input-to-hidden weights. When $X = Y$ - that is when both are 0 or both are 1 we want both hidden unit outputs to be 0 so both bias-to-hidden weights are -10 and input-to-hidden weights for each hidden neuron must have opposite sign and be greater than bias-to-hidden weight in absolute value. When $X \neq Y$ we want one of the hidden units to give output 1 (the other will be 0). So, the two weights originating at X and Y should have opposite sign and equal value.

So summarizing:
All bias weights are -10.
One of the following.

	Upper hidden	Lower hidden
X	100	-100
Y	-100	100

or (if you flip the semantic labels given to the hidden neurons)

	Upper hidden	Lower hidden
X	-100	100
Y	100	-100

[8,3,9=20]