CS 391L Machine Learning Assignment 3

Name: Shun Zhang
Email address: jensen.zhang@utexas.edu
EID: sz4554

Problem 1

(a)

$$
\begin{aligned}
Cov(y) &= E((y - E(y)(y - E(y)^T) && (1)\\
&= E((Ax + b - E(Ax + b))(Ax + b - E(Ax + b))^T) && (2)\\
&= E((Ax - E(Ax))(Ax - E(Ax))^T) && \text{Linearity of } E \quad (3)\\
&= AE((x - E(x))(x - E(x))^T)A^T && (4)\\
&= A\Sigma A^T && \text{Def. of Cov.} \quad (5)
\end{aligned}
$$

(b) Base case: when $k = 0$, $Ix = \lambda^0 x$. When $k = 1$, by the definition of eigenvalue and eigenvector, $Ax = \lambda x$.

Inductive hypothesis: assume $A^k x = \lambda^k x$ for some $k \in N$. Want to show $A^{k+1}x = \lambda^{k+1}x$.

$$
\begin{aligned}
A^k x &= \lambda^k x && \text{I.H.} \quad (6)\\
A^{k+1} x &= A\lambda^k x && (7)\\
A^{k+1} x &= \lambda^k Ax && (8)\\
A^{k+1} x &= \lambda^{k+1} x && Ax = \lambda x \quad (9)
\end{aligned}
$$

If $k < 0$, we already know $A^{-k}x = \lambda^{-k}x$. By algebra, $\lambda^k A^{-k} x = x$, $\lambda^k x = A^k x$.

Problem 2

(a)

$$r = 1 - \frac{H(Y|X)}{H(X)} \tag{10}$$
$$= \frac{H(X) - H(Y|X)}{H(X)} \tag{11}$$
$$= \frac{I(Y|X)}{H(X)} \qquad \text{Def. of Mutual Information} \tag{12}$$

(b) $H(Y|X) \geq 0$, $H(X) > 0$. So $\frac{H(Y|X)}{H(X)} \geq 0$, $1 - \frac{H(Y|X)}{H(X)} \leq 1$.

$H(Y|X) < H(X)$, so $\frac{H(Y|X)}{H(X)} \leq 1$, $1 - \frac{H(Y|X)}{H(X)} \geq 0$.

Therefore, $0 \leq r \leq 1$.

(c) $r = 0$ when two variables are independent. $r = 1$ when two variables are perfectly correlated (positively linearly related).

Problem 3

(a) $\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$. Compared to $\frac{1}{1+e^{-x}}$, tanh has a steeper slope. In the literature, tanh has a good property in its derivative to reduce error as calculated by an error function [1].

To be an appropriate sigmoid function, tanh needs to be scaled to range of from 0 to 1.

(b) We know W^k is 2×2.

$$\frac{\partial H}{\partial w_{ij}^k} = \frac{\partial}{\partial w_{ij}^k} (\lambda^{k+1})^T g(W^k x^k) \tag{13}$$
$$= \frac{\partial}{\partial w_{ij}^k} (\lambda_1^{k+1} g(w_{11}x_1^k + w_{12}x_2^k) + \lambda_2^{k+1} g(w_{21}x_1^k + w_{22}x_2^k)) \tag{14}$$
$$= \lambda_1^{k+1} \frac{\partial}{\partial w_{ij}^k} g(w_{11}x_1^k + w_{12}x_2^k) + \lambda_2^{k+1} \frac{\partial}{\partial w_{ij}^k} g(w_{21}x_1^k + w_{22}x_2^k) \tag{15}$$

Therefore, for $w_{11}, w_{12}, w_{21}, w_{22}$, there is $\frac{\partial H}{\partial w_{ij}^k} = \lambda^{k+1} x_j^k g'(w_i^k x^k)$.

Problem 4

(a) IG(Color) = 0.1043
IG(Size) = 0.4086
IG(Noise) = 0.0207

For small size,

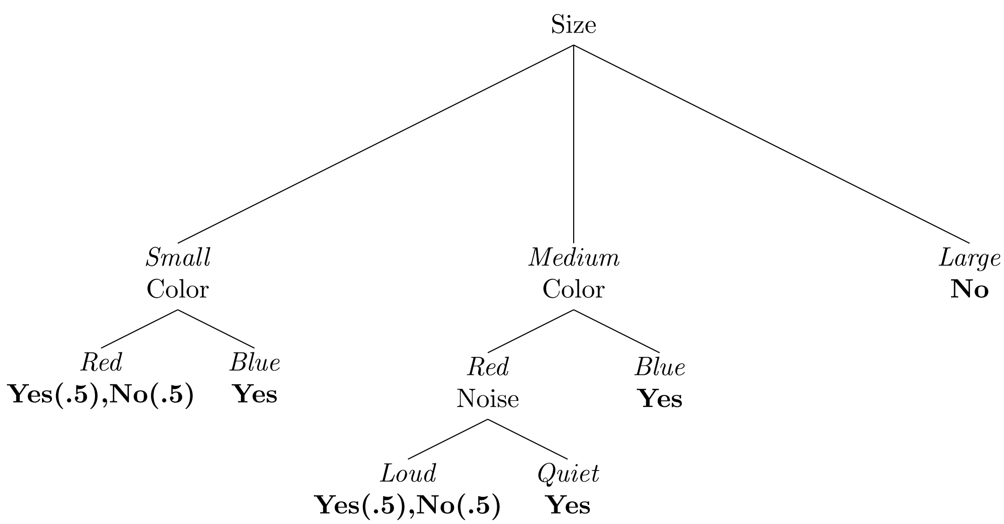
IG(Color) = 0.3219
IG(Noise) = 0.0729

For medium size,

IG(Color) = 0.1226
IG(Noise) = 0.1226

For large size, IG is clearly 0.

The decision tree is



For inconsistent data, the probability of each possible outcome is shown in the parenthesis.

(b) If the event of missing a datum is uniformly random over all the attributes, then it doesn't harm if we simply delete that line.

Problem 5

(a) $\min \dfrac{1}{3}\pi r^2 h$ such that $A = \pi r\sqrt{r^2+h^2} + \pi r^2$,

Use Lagrange multiplier, $H = \frac{1}{3}\pi r^2 h + \lambda(\pi r\sqrt{r^2+h^2} + \pi r^2 - A)$.

$h = \sqrt{\frac{2A}{\pi}}$, $r = \sqrt{\frac{A}{4\pi}}$.

Problem 6

Each node has m possible attributes. So there are no more than m^n configuration of the decision tree. The number of different classifiers is also no more than m^n.
We know that for data set with size of k, a decision tree should have 2^k decisions. So the VC dimension is $\log_2 m^n = n \log_2 m = O(n \log(m))$.

Problem 7

The smallest positive integer p is 2.
When $p = 1$, $k(x, x_i) = 1 + x^T x_i = 1 + x_1 x_{i1} + x_2 x_{i2}$. So $\Phi(x) = (1, x_1, x_2)^T$. Clearly, as (x_1, x_2) cannot be separated by SVM, adding a bias of 1 doesn't help either.
When $p = 2$ — same case as the class note — $k(x, x_i) = (1 + x^T x_i)^2 = (1 + x_1 x_{i1} + x_2 x_{i2})^2 = 1 + x_1^2 x_{i1}^2 + x_2^2 x_{i2}^2 + 2x_1 x_{i1} + 2x_2 x_{i2} + 2x_1 x_{i1} x_2 x_{i2}$. Then $\Phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2)^T$.
Let $\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix}$. $Q(\lambda) = I\lambda - \frac{1}{2}\lambda^T A \lambda$, where $A = \begin{pmatrix} 9 & -1 & -1 & 1 \\ -1 & 9 & 1 & -1 \\ -1 & 1 & 9 & -1 \\ 1 & -1 & -1 & 9 \end{pmatrix}$, the elements of which are computed by the kernel function. The solution is $\lambda_i = \frac{1}{8}$ for $i = 1, 2, 3, 4$. They are all support vectors, and they are linearly separable.

Using a value of p larger than minimum would unnecessarily map the data to higher dimension. The classifier becomes more flexible in lower dimension, and thus may overfit the data.
However, if it has a regularization term, it doesn't have such downside. For example, if the regularization term is $||w||_1$, then it becomes sparse coding.

References

[1] Kalman, B.L.; Kwasny, S.C., *Why tanh: choosing a sigmoidal function*, Neural Networks, 1992. IJCNN., International Joint Conference on , vol.4, no., pp.578,581 vol.4, 7-11 Jun 1992.