SM1 Assessed Homework 2

Sam Bowyer

November 2022

Question 0

The polynomial feature transform with degree $b = 2$ (**C**) would perform well on this dataset since it could create the roughly circular decision boundary necessary to separate the classes (recall the equation of a circle $x^2 + y^2 = c$), whilst keeping the computational cost low. The feature transforms in **A** and **B** would not be complex enough to capture the shape of this circular decision boundary, whilst those in **D** and **E** would be able to capture this shape, however, they would be more computationally expensive than **C** and would provide no better generalisation to the required decision boundary (**D** adds unneeded cubic terms and **E** would be very expensive and could easily overfit on the data).

Question 1

1.1

The direction of the FDA embedding vector $\mathbf{w}$ would likely be c or d, since the projection of the data onto these directions (which are really the same direction, since $c = -d$) leads to a better separation of the data classes than projection onto a or b.

1.2

1.2.1

Assuming that our inputs $D = \{\mathbf{x}_i\}_{i=1}^N$ are i.i.d. with $\mathbf{x}_i \sim \mathcal{N}_{\mathbf{x}_i}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for $\boldsymbol{\mu} \in \mathbb{R}^2$, $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$, the likelihood function over the dataset would be:

$$\begin{aligned} p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{N} \mathcal{N}_{\mathbf{x}_i}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \prod_{i=1}^{N} (2\pi)^{-1} \det(\boldsymbol{\Sigma})^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right) \\ &= (2\pi)^{-N} \det(\boldsymbol{\Sigma})^{-N/2} \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}))\right). \end{aligned}$$

1.2.2

Since $D_1 > D_2$ we know that the data varies more in the direction of $\mathbf{u}_1$ than in the direction of $\mathbf{u}_2$ (with $||\mathbf{u}_1|| = ||\mathbf{u}_2|| = 1$), hence $\mathbf{u}_1 \in \{a, b\}$.

Question 3

We've seen that we can rewrite the soft-margin SVM objective function from

$$||\mathbf{w}'||^2 + \sum_{i \in D} \epsilon_i$$

to

$$||\mathbf{w}'||^2 + \sum_{i \in D} L(y_i f(\mathbf{x}_i; \mathbf{w}))$$

where $L(z) = \max(0, 1 - z)$. This second form more clearly presents itself as a regularised loss function and might also be written as

$$||\mathbf{w}'||^2 + \sum_{i \in D} \begin{cases} 0 & \text{if } y_i f(x_i) \geq 1, \\ 1 - y_i f(\mathbf{x}_i; \mathbf{w}) & \text{otherwise.} \end{cases}$$

We can therefore change the loss function here to impose a penalty on false negatives that is 1000 times the penalty given to a false positive (which will remain simply as ϵ_i—a correct classification still leads to $\epsilon_i = 0$):

$$||\mathbf{w}'||^2 + \sum_{i \in D} \begin{cases} 1000\epsilon_i & \text{if } y_i = 1 \text{ and } f(x_i) < 0, \\ \epsilon_i & \text{otherwise.} \end{cases}$$

Question 4

4.1

First note that a covariance matrix must be symmetric, which rules out **C** and **E**.

Next, **D** implies that the 5th random variable has a variance of 0, which would mean that all covariances with this (constant) r.v. would also be 0, something that is not found in the matrix, hence **D** is also ruled out.

As discussed in Lecture 13, if P is a Gaussian Markov Network of 5 variables, then the number of non-zero non-diagonal elements in the precision matrix $\Theta = \Sigma^{-1}$ (where Σ is the covariance matrix of P) will be equal to twice the number of edges in the graph (we multiply by two since the edges are undirected). Furthermore, we know that P is sparse, so the number of edges is less than half the number of edges in a K_5, i.e. P has $\frac{1}{2}\binom{5}{2} = 5$ or fewer edges, and hence Θ has $2 \times 5 = 10$ or fewer non-zero non-diagonal entries. Finally we may see that **A** leads to a precision matrix containing *only* non-zero entries:

$$\Theta = \Sigma^{-1} = \begin{pmatrix} 1 & 0.5 & 0 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 & 0 \\ 0 & 0.5 & 1 & 0.5 & 0 \\ 0 & 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0 & 0.5 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 5/3 & -4/3 & 1 & -2/3 & 1/3 \\ -4/3 & 8/3 & -2 & 4/3 & -2/3 \\ 1 & -2 & 3 & -2 & 1 \\ -2/3 & 4/3 & -2 & 8/3 & -4/3 \\ 1/3 & -2/3 & 1 & -4/3 & 5/3 \end{pmatrix}$$

This means that **A** can't represent a sparse Gaussian Markov Network (the same analysis can be used to disqualify **D** and **E**).

Finally, this leaves us with the option B, which *could* be the covariance matrix of P since it is symmetric and gives a precision matrix

$$\Theta = \begin{pmatrix} 1 & 0.5 & 0 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 & 0 \\ 0 & 0.5 & 1 & 0.5 & 0 \\ 0 & 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0 & 0.5 & 1 \end{pmatrix}$$

which contains $8 < 10$ non-zero non-diagonal entries.

4.2

4.2.1

Recalling the definition of factorisation within Bayesian Networks from Lecture 13:

"We say a probability dist. $p(X)$ factorizes over a DAG G if $p(X) = \prod_{v \in V} p(X_v | X_{\text{parent}(X_v)})$".

Applying this to the problem at hand, we can write the required joint probability as:

$$p(y, x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}) = p(y)p(x^{(1)}|y)p(x^{(2)}|y, x^{(1)})p(x^{(3)}|x^{(2)})p(x^{(4)}|x^{(1)}).$$

4.2.2

Lecture 13 also tells us that "Given a DAG G, X_v is independent of $X_{\text{non-desc}(X_v)}$ given $X_{\text{parent}(X_v)}$". Therefore this question's graph encodes the following conditional independence:

- $x^{(2)} \perp x^{(4)} | y, x^{(1)}$.
- $x^{(3)} \perp y, x^{(1)} x^{(4)} | x^{(2)}$.
- $x^{(4)} \perp y, x^{(2)} x^{(3)} | x^{(1)}$.

4.2.3

Using the input features $x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}$ to predict y would rely on calculating:

$$\begin{aligned}
\hat{y} := \operatorname*{argmax}_y p(y|x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}) &= \operatorname*{argmax}_y \frac{p(y, x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})}{p(x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})} \\
&= \operatorname*{argmax}_y \frac{p(y)p(x^{(1)}|y)p(x^{(2)}|y, x^{(1)})p(x^{(3)}|x^{(2)})p(x^{(4)}|x^{(1)})}{p(x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})} \\
&= \operatorname*{argmax}_y p(y)p(x^{(1)}|y)p(x^{(2)}|y, x^{(1)})
\end{aligned}$$

Where we can remove the $p(x^{(3)}|x^{(2)})p(x^{(4)}|x^{(1)})/p(x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$ factor because it does not depend on y meaning that

$$p(y|x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}) \propto p(y)p(x^{(1)}|y)p(x^{(2)}|y, x^{(1)}).$$

Hence you should only use the input features $x^{(1)}$ and $x^{(2)}$ since the other features won't change the prediction of $\hat{y}$ and may only serve to increase computation time.