CS 344 (Spring 2017): Mid-semester Examination*

Instructor: Shivaram Kalyanakrishnan

11.00 a.m. – 1.00 p.m., February 23, 2017, 101/103 New CSE Building

Total marks: 20

Note Provide brief justifications and/or calculations along with each answer to illustrate how you arrived at the answer.

Question 1. Consider the problem of k-means clustering, $k \geq 2$, in 1-dimensional space. The input is a set of distinct real-valued scalars $\{x^1, x^1, \ldots, x^n\}$ $(n > k)$ that are numbered in increasing order:

$$x^1 < x^2 < \cdots < x^n.$$

The expected output is a clustering $\mathcal{C} : \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, k\}$ and a sequence of centres $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_k)$, where for $k' \in \{1, 2, \ldots, k\}$, $\mu_{k'} \in \mathbb{R}$. . Recall that

$$\text{SSE}(\mathcal{C}, \boldsymbol{\mu}) = \sum_{i=1}^{n} (x^i - \mu_{\mathcal{C}(i)})^2.$$

1a. Suppose $(\mathcal{C}, \boldsymbol{\mu})$ is the output of a run of the k-means clustering algorithm. Assume that the centres in $\boldsymbol{\mu}$ are distinct. Show that there cannot exist $r, s, t \in \{1, 2, \ldots, n\}$ with $r < s < t$ such that $\mathcal{C}(r) = \mathcal{C}(t) \neq \mathcal{C}(s)$. (You might find it useful to start by drawing the real line and marking $x^r < x^s < x^t$ upon it. Then consider the constraint that any solution returned by the k-means clustering algorithm must satisfy.) [5 marks]

1b. Based on 1a, what useful property can we expect in an *optimal* clustering of 1-dimensional points? [1 mark]

1c. Now consider the special case of $k = 2$: that is, we wish to partition the set of n points on the line into 2 clusters. Use the property identified in 1b to design an efficient algorithm for finding an optimal clustering in this case . Provide pseudocode for your algorithm, assuming that the input is provided as a sorted array. What is the running time of the algorithm as a function of n (as an order complexity expression)? To receive full marks, you will have to make it as efficient as possible. [5 marks]

Question 2. We never test the same categorical attribute more than once along any given path in a decision tree. Why? [1 mark]

*Questions 2 and 3 are based on exercises in the textbook by Russell and Norvig (2010).

Question 3. We plan to implement a 7-nearest neighbour predictor for regression. For a given query point, let $(\boldsymbol{x}^1, y^1), (\boldsymbol{x}^2, y^2), \ldots, (\boldsymbol{x}^7, y^7)$ be the 7 nearest training data points (in terms of $\boldsymbol{x}$). Our rule is to give $\hat{y}$ as our prediction, where $\hat{y}$ minimises the aggregate L_1 loss with respect to its neighbours:

$$\hat{y} = \min_{y \in \mathbb{R}} \sum_{i=1}^{7} |y - y^i|.$$

Suppose for a query point, the 7 nearest neighbours have the following y values: $14, 1, 6, 4, 21, 13, 88$. What $\hat{y}$ value is predicted? What is the common name in statistics for $\hat{y}$ as a function of $y^1, y^2, \ldots, y^7$? [2 marks]

Question 4. Logistic regression is a supervised learning method for binary classification. The basic model in use is an artificial neuron. However, rather than minimising the squared loss over the training data (as we did in class), the method attempts to find weights that maximise the likelihood that the model generated the data. As a consequence of this difference, the objective function achieves the desirable property of having a single (global) optimum, which can be found using gradient descent.

This question takes you through the steps of convincing yourself that logistic regression enjoys the claimed property. However, rather than prove the result in full generality, we shall assume that our data is 1-dimensional. Thus, there is only a single scalar weight $w \in \mathbb{R}$ to optimise.

We are given a data set $(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)$, where for $i \in \{1, 2, \ldots, n\}$, $x^i \in \mathbb{R}$ and $y^i \in \{-1, 1\}$. Assume that the x-values are distinct, and that there is at least one point in each class. We wish to fit the data with an artificial neuron, which calculates $y = \sigma(wx)$. For $\alpha \in \mathbb{R}$,

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)}.$$

We interpret that for a given data point, the y-value was generated by tossing a coin with bias $\sigma(wx)$. The y-value $+1$ is interpreted as a head, and -1 as a tail. Hence, the likelihood $L(w)$ that w generated the training data set is given by

$$L(w) = \left(\prod_{i \in \{1,2,\ldots,n\}, y^i = +1} \sigma(wx^i) \right) \left(\prod_{i \in \{1,2,\ldots,n\}, y^i = -1} (1 - \sigma(wx^i)) \right).$$

Your task is to show that $L(w)$ has a unique maximum. In order to do so, derive expressions for the following quantities. (You might find it convenient to use $\sigma'(\alpha) = \sigma(\alpha)(1 - \sigma(\alpha))$.)

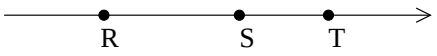
- The natural logarithm of $L(w)$, denoted $LL(w)$ (for *log-likelihood*). [1 mark]
- $\frac{d}{dw} LL(w)$. [1 mark]
- $\frac{d^2}{dw^2} LL(w)$. [1 mark]
- $\lim_{w \to \infty} \frac{d}{dw} LL(w)$. [1 mark]
- $\lim_{w \to -\infty} \frac{d}{dw} LL(w)$. [1 mark]

Based on the expressions you have derived, put together an argument that indeed $L(w)$ has a unique maximum. [1 mark]

Solutions

1a. Let us consider arbitrary $r, s, t \in \{1, 2, \ldots, n\}$ such that $r < s < t$. We shall show that $(\mathcal{C}, \boldsymbol{\mu})$ cannot be such that $\mathcal{C}(r) = \mathcal{C}(t) \neq \mathcal{C}(s)$. For convenience, we use the names $R, S, T, M_1,$ and M_2 to denote our points of interest in $\mathbb{R}$. The points R, S, and T are shown below on the real line.

$$R = x^r; S = x^s; T = x^t; M_1 = \mu_{\mathcal{C}(r)} = \mu_{\mathcal{C}(t)}; M_2 = \mu_{\mathcal{C}(s)}.$$



Since $(\mathcal{C}, \boldsymbol{\mu})$ is the output of the k-means clustering algorithm, the points must satisfy the following *proximity* constraints.

$$RM_1 \leq RM_2.$$
$$SM_1 \geq SM_2.$$
$$TM_1 \leq TM_2.$$

We consider every possible configuration of M_1 and M_2, and show that in each case, not all three proximity constraints can be simultaneously satisfied. From the description, we know that R, S, and T are distinct, and also that $M_1 \neq M_2$.

- Assume $M_2 \in (R, T)$. Then (1) $M_1 < M_2$ would imply $TM_2 < TM_1$, and (2) $M_1 > M_2$ would imply $RM_2 < RM_1$. Hence, we must have $M_2 \in (-\infty, R]$ or $M_2 \in [T, \infty)$. Since these cases are symmetric, we only examine the former.
- Assume $M_2 \in (-\infty, R]$. Then (1) $M_1 < M_2$ would imply $RM_2 < RM_1$, and (2) $M_2 < M_1 \leq S$ would imply $SM_1 < SM_2$. (3) If $M_1 > S$, the constraints $RM_1 \leq RM_2$ and $SM_1 \geq SM_2$ cannot both hold. To see why, assume that $M_1 > S$ and $RM_1 \leq RM_2$. Then, $SM_2 = RS + RM_2 \geq RS + RM_1 = 2RS + SM_1 > SM_1$.

Our proof is done.

1b. It follows from the condition listed in 1a that every optimal clustering (a clustering that minimises SSE) must have *contiguous* clusters: that is, there must exist $r_1, r_2, \ldots, r_{k-1} \in \{1, 2, \ldots, n-1\}$ such that every point in $\{1, 2, \ldots, r_1\}$ is assigned the same cluster, every point in $\{r_1 + 1, r_1 + 2, \ldots, r_2\}$ is assigned the same cluster, $\ldots$, and every point in $\{r_{k-1}+1, r_{k-1}+2, \ldots, n\}$ is assigned the same cluster. In other words, an optimal clustering segments the real line into k intervals, each associated with a separate cluster.

1c. Finding an optimal clustering for $k = 2$ reduces to the problem of finding a single number $r \in \{1, 2, \ldots, n-1\}$ such that if every point in $\{1, 2, \ldots, r\}$ is assigned cluster 1, and every point in $\{r+1, r+2, \ldots, n\}$ is assigned cluster 2, the SSE is minimised. Hence, we may construe SSE as a function of r. In other words, we have to find $\min_{r \in \{1,2,\ldots,n-1\}} SSE(r)$, where

$$\text{SSE}(r) = \sum_{i=1}^{r} (x^i - \mu_1(r))^2 + \sum_{i=r+1}^{n} (x^i - \mu_2(r))^2,$$

$$\mu_1(r) = \frac{1}{r} \sum_{i=1}^{r} x^i \text{ and } \mu_2(r) = \frac{1}{n-r} \sum_{i=r+1}^{n} x^i.$$

The most natural procedure to find the minimising value of r would be to iterate from 1 to $n-1$, calculate the SSE for each iterate r, and pick the minimiser. If we calculate SSE each time based on the formula above, each calculation would take $O(n)$ steps, leading to an overall complexity of $O(n^2)$ for our procedure. However, upon closer inspection, we observe that an $O(n)$ *preprocessing* step can help bring down the SSE calculation for a given r to merely an $O(1)$ operation.

For $i \in \{1, 2, \ldots, n\}$, define $A_i = \sum_{j=1}^{i} x^j$, and $B_i = \sum_{j=1}^{i} (x^j)^2$. Now observe that

$$\begin{aligned}
\text{SSE}(r) &= \sum_{i=1}^{r} (x^i - \mu_1(r))^2 + \sum_{i=r+1}^{n} (x^i - \mu_2(r))^2 \\
&= \sum_{i=1}^{r} \left((x^i)^2 + (\mu_1(r))^2 - 2x_i\mu_1(r) \right) + \sum_{i=r+1}^{n} \left((x^i)^2 + (\mu_2(r))^2 - 2x_i\mu_2(r) \right) \\
&= \sum_{i=1}^{n} (x^i)^2 - r \cdot (\mu_1(r))^2 - r \cdot (\mu_2(r))^2 \\
&= B_n - r \left(\left(\frac{A_r}{n} \right)^2 + \left(\frac{A_n - A_r}{n - r} \right)^2 \right).
\end{aligned}$$

Hence, SSE(r) only depends on A_r, A_n, and B_n, which can all be pre-computed, as in the following linear-time algorithm.

To find an optimal clustering when $k = 2$

$A_1 \leftarrow x^1; B_1 \leftarrow (x^1)^2$.
For $i = 2, 3, \ldots, n$:
    $A_i \leftarrow A_{i-1} + x^i; B_i \leftarrow B_{i-1} + (x^i)^2$.
$\text{SSE}_{\min} \leftarrow \infty$.
$r_{\min} \leftarrow -1$.
For $r = 1, 2, \ldots, n-1$:
    $\text{SSE}_r \leftarrow B_n - r \left(\left(\frac{A_r}{n} \right)^2 + \left(\frac{A_n - A_r}{n-r} \right)^2 \right)$.
    If $\text{SSE}_r < \text{SSE}_{\min}$ **then**:
        $\text{SSE}_{\min} \leftarrow \text{SSE}_r; r_{\min} \leftarrow r$.
For $i \in \{1, 2, \ldots, r_{min}\}$:
    $\mathcal{C}(i) \leftarrow 1$.
For $i \in \{r_{min} + 1, r_{min} + 2, \ldots, n\}$:
    $\mathcal{C}(i) \leftarrow 2$.
$\mu_1 \leftarrow \frac{A_{r_{min}}}{r_{min}}; \mu_2 \leftarrow \frac{A_n - A_{r_{min}}}{n - r_{min}}$.
Return $\mathcal{C}, (\mu_1, \mu_2)$.

(Observe that even $B_1, B_2, \ldots, B_n$ are not needed to for finding the optimal clustering.)

2. When we split on a categorical attribute A, each resulting child node inherits training data points that all have the same value v of that attribute. In the child nodes and their descendants, attribute A no longer provides any additional information about the label; splitting further on A would result in one child with value v that inherits all the training data, and children with no data for every other attribute value. Hence, if "$A = v$" is a split, the same split serves no purpose downstream. (It is less common to split categorical variables, say, on disjunctions such as "$A = v_1 \vee A = v_2$". In such a case, a downstream split of the form "$A = v_1$" would still be useful.)

3. The L_1 loss is minimised when $\hat{y}$ is the *median* of the 7 y-values, which, in this case, is 13.

4.

$$LL(w) = \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1}} \log(\sigma(wx^i)) + \sum_{i\in\{1,2,\ldots,n\}, y^i=-1} \log(1-\sigma(wx^i)).$$

$$\begin{aligned}\frac{d}{dw}LL(w) &= \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1}} \frac{\sigma'(wx^i)x^i}{\sigma(wx^i)} + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1}} \frac{-\sigma'(wx^i)x^i}{1-\sigma(wx^i)} \\ &= \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1}} (1-\sigma(wx^i))x^i - \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1}} \sigma(wx^i)x^i.\end{aligned}$$

$$\begin{aligned}\frac{d^2}{dw^2}LL(w) &= -\sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1}} \sigma'(wx^i)(x^i)^2 - \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1}} \sigma'(wx^i)(x^i)^2 \\ &= -\sum_{i\in\{1,2,\ldots,n\}} \sigma(wx^i)(1-\sigma(wx^i))(x^i)^2.\end{aligned}$$

$$\begin{aligned}\lim_{w\to-\infty}\frac{d}{dw}LL(w) &= \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1\\ x^i>0}} x^i + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1\\ x^i=0}} 0 + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1\\ x^i<0}} 0 + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1\\ x^i>0}} 0 + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1\\ x^i=0}} 0 + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1\\ x^i<0}} (-x^i) \\ &= \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1\\ x^i>0}} x^i - \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1\\ x^i<0}} x^i.\end{aligned}$$

$$\begin{aligned}\lim_{w\to\infty}\frac{d}{dw}LL(w) &= \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1\\ x^i>0}} 0 + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1\\ x^i=0}} 0 + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1\\ x^i<0}} x^i + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1\\ x^i>0}} (-x^i) + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1\\ x^i=0}} 0 + \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1\\ x^i<0}} 0 \\ &= \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=+1\\ x^i<0}} x^i - \sum_{\substack{i\in\{1,2,\ldots,n\}\\ y^i=-1\\ x^i>0}} x^i.\end{aligned}$$

We see $\lim_{w\to-\infty}\frac{d}{dw}LL(w) \geq 0$, and $\lim_{w\to\infty}\frac{d}{dw}LL(w) \leq 0$. Since there is at least one point in each class, and the x-values are distinct, it follows that either $\lim_{w\to-\infty}\frac{d}{dw}LL(w)$ or $\lim_{w\to\infty}\frac{d}{dw}LL(w)$ is not exactly equal to 0. Additionally, $\frac{d^2}{dw^2}LL(w) < 0$, which implies that $\frac{d}{dw}LL(w)$ monotonically decreases, and so reaches 0 at exactly one point $w^\star \in \mathbb{R}$. We conclude that $LL(w)$ has a unique maximum. Since $L(w) = \exp(LL(w))$, it must also have a unique maximum.