CSC 411/2515 Example MID-TERM Fall 2017

Name: Student Number:

Read the following instructions carefully:

1. Do not turn the page until told to do so.
2. If a question asks you to do some calculations, you must *show your work* to receive full credit.
3. You can use either pen or pencil for the exam. **But please be aware that you are not allowed to dispute any credit after the exam is returned if you use a pencil.**
4. Use the back of the page if you need more space on a question.
5. Lastly, enjoy the problems!

1. True/False questions (15 points)

Statement	**True**	**False**
Decision trees can achieve zero classification error on any training data (assuming each training data point is unique).		
Assume that you have training data with continuous features and targets. Linear regression trained with L_2 loss is robust to outliers in the training data.		
Let X and Y be two discrete random variables and H denote the entropy. Then $H(X\|Y)+H(Y) = H(Y\|X)+H(X)$.		
Assume that you have training data with continuous features. You should always transform the features to lie in the range $[0, 1]$ before using nearest neighbors.		
If you divide your data into train-validation-test sets to fit and evaluate your model then you cannot overfit to your validation set.		

Solution

True, False, True, False, False

2. Fill in the blanks(5 points)

Given a discriminative model with parameters θ and training data pairs $\mathbf{x}, y$.

- The likelihood is ....

- MAP estimation maximizes .... × ....

Solution

- $p(y|\mathbf{x}, \theta)$
- $p(\theta) \times$ ~~$p(\mathbf{x}, y|\theta)$~~ (or prior × likelihood).

3. Effect of linear transformation (30 points)

Assume we are preprocessing our data using an **invertible** linear transformation on the features of our training data. The transformation can either be some orthogonal (i.e. rotations) matrix or some diagonal matrix.

Say if this can have any effect on the performance of the following algorithms, and explain in no more than two sentences.

- Orthogonal preprocessing on decision tree classification.

Solution

Will have an effect. Decision trees make axis-aligned cuts, rotations will change the axes.

- Diagonal preprocessing on decision tree classification.

Solution

Will not have an effect. Rescaling the features may shift our axis-cuts but will not change our decisions.

- Orthogonal preprocessing on nearest neighbor classification.

Solution

Will not have an effect. Orthogonal transformations are distance preserving and so the distance between datapoints will remain the same.

- Diagonal preprocessing on nearest neighbor classification.

Solution

Will have an effect. Nearest neighbors is sensitive to feature scales as distances will not be preserved.

4. L_2 regularization and robustness to noise (25 points)

Given input $\mathbf{x} \in \mathbb{R}^d$ and target $y \in \mathbb{R}$, define $\hat{\mathbf{x}} = \mathbf{x} + \boldsymbol{\epsilon}$ to be a noisy pertubation of $\mathbf{x}$ where we assume

- $\mathbb{E}[\epsilon_i] = 0$
- for $i \neq j$: $\mathbb{E}[\epsilon_i \epsilon_j] = 0$
- $\mathbb{E}[\epsilon_i^2] = \lambda$

We define the following objective that tries to be robust to noise

$$\mathbf{w}^* = \arg\min \mathbb{E}_\epsilon[(\mathbf{w}^T \hat{\mathbf{x}} - y)^2] \tag{1}$$

Show that it is equivalent to minimizing L_2 regularized linear regression, i.e.

$$\mathbf{w}^* = \arg\min \left[(\mathbf{w}^T \mathbf{x} - y)^2 + \lambda ||\mathbf{w}||^2\right] \tag{2}$$

Solution

We can write the inner term as,

$$\begin{aligned}
(\mathbf{w}^T \hat{\mathbf{x}} - y)^2 &= (\mathbf{w}^T \mathbf{x} + \mathbf{w}^T \boldsymbol{\epsilon} - y)^2 && (3)\\
&= (\mathbf{w}^T \mathbf{x} - y)^2 + 2\mathbf{w}^T \boldsymbol{\epsilon}(\mathbf{w}^T \mathbf{x} - y) + (\mathbf{w}^T \boldsymbol{\epsilon})^2 && (4)\\
&= (\mathbf{w}^T \mathbf{x} - y)^2 + 2\mathbf{w}^T \boldsymbol{\epsilon}(\mathbf{w}^T \mathbf{x} - y) + (\mathbf{w}^T \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \mathbf{w}) && (5)
\end{aligned}$$

Under the expectation the second term will be zero as it is a linear combination of the elements of $\boldsymbol{\epsilon}$. The final term will be the quadratic form of $\mathbf{w}$ with the covariance of $\boldsymbol{\epsilon}$. The covariance is simply λI. Thus we are minimizing,

$$(\mathbf{w}^T \mathbf{x} - y)^2 + \lambda ||\mathbf{w}||^2$$

which is exactly the objective of L2-regularized linear regression.

5. Naive Bayes (25 points)

Naive Bayes defines the joint probability of each datapoint $\mathbf{x} \in \mathbb{R}^d$ and it's class label c as follows:

$$p(\mathbf{x}, c|\boldsymbol{\theta}) = p(c)p(\mathbf{x}|c, \theta_c) = p(c) \prod_{i=1}^{d} p(x_i|c, \theta_{cd}) \tag{6}$$

For this question, we will consider only the Bernoulli Naive Bayes model, where

$$p(x_i|c, \theta_{cd}) = \theta_{cd}^{x_i}(1 - \theta_{cd})^{1-x_i}$$

, for all $i = 1 \cdots d$.

(a) True or false: In the Naive Bayes model, any two features x_i and x_j, where $i \neq j$, are independent given c.

Solution

True

(b) True or false: Naive Bayes is a non-parametric model.

Solution

False

(c) Now assume that there are K classes and $p(c) = \frac{1}{k}$. Derive the class predictive log-likelihood for the Naive Bayes model, $\log p(c|\mathbf{x}, \boldsymbol{\theta})$ for a single data point.

Solution

$$\begin{aligned}
\log p(c|x,\theta) &= \log \frac{p(x|c,\theta)p(c)}{p(x|\theta)} && (7)\\
&= \log(p(c)) + \sum_{i=1}^{d}(x_i \log(\theta_{c,i}) + (1-x_i)\log(1-\theta_{c,i})) - \log(\sum_{c} p(x,c|\theta)) && (8)\\
& && (9)
\end{aligned}$$

Final term is given in the question definition.

(d) (*For those who want even more to do.*) Now additionally assume that $d = 10$. For a single data point we observe $\mathbf{x}_a = [x_1, \cdots, x_5]$, but do not observe $\mathbf{x}_b = [x_6, \cdots, x_{10}]$. Derive $p(\mathbf{x}_b|\mathbf{x}_a, \theta)$ - the distribution over the unobserved features conditioned on the features which we have observed.

Solution

$$\begin{aligned}
p(\mathbf{x}_b|\mathbf{x}_a,\theta) &= \sum_c p(\mathbf{x}_b|\mathbf{x}_a,c,\theta)p(c|\mathbf{x}_a,\theta) && (10)\\
&= \sum_c p(\mathbf{x}_b|c,\theta)p(c|\mathbf{x}_a,\theta) \qquad \text{(conditional independence)} && (11)\\
&= \sum_c p(\mathbf{x}_b|c,\theta)\frac{p(\mathbf{x}_a|c,\theta)p(c)}{p(\mathbf{x}_a|\theta)} && (12)\\
&= \sum_c \frac{p(\mathbf{x}|c,\theta)p(c)}{p(\mathbf{x}_a|\theta)} && (13)\\
&= \frac{p(\mathbf{x}|\theta)}{p(\mathbf{x}_a|\theta)} && (14)
\end{aligned}$$