INF283 Introduction to machine learning
Spring 2019

Exam

18.2.2019

This exam has 6 tasks on 5 pages. You can get at most 50 points. No aids permitted.

1 Basic concepts (10 p.)

1.1 Bias-variance tradeoff

Give short answers to the following questions:

a) What is the bias-variance tradeoff? (No need to show formulas)

b) How is it related to overfitting?

1.2 Cross-validation

Give short answers to the following questions:

c) What is the purpose of using cross-validation?

d) How does cross-validation work?

1.3 k-nearest neighbor classifier

Explain briefly how the k-nearest neighbor classifier works.

2 Probabilistic models (8 p.)

Consider the following Naive Bayes classifier. There are two classes $C \in \{0, 1\}$ and three binary features x_1, x_2, x_3. We have the following probabilities:

$$\begin{aligned} P(C = 1) &= 1/5 \\ P(x_1 = 1 | C = 0) &= 1/3 \\ P(x_1 = 1 | C = 1) &= 2/3 \\ P(x_2 = 1 | C = 0) &= 3/4 \\ P(x_2 = 1 | C = 1) &= 1/6 \\ P(x_3 = 1 | C = 0) &= 5/8 \\ P(x_3 = 1 | C = 1) &= 1/3 \end{aligned}$$

Answer the following questions. **Show your computations**.

a) We have observed that $x_1 = 1$, $x_2 = 0$, and $x_3 = 1$. What is our prediction for the class C?

b) We have observed that $x_1 = 1$ and $x_2 = 0$. What is our prediction for the class C?

3 Ensemble methods (8 p.)

Give short answers to the following questions:

a) What are ensemble methods? Why are they useful?

b) Bagging and boosting are commonly used ensemble methods. How do they differ from each other?

4 Hierarchical clustering (8 p.)

You are given a data set $D = \{0, 5, 9, 12, 18\}$ consisting of 5 one-dimensional points. Find an agglomerative hierarchical clustering using complete linkage (farthest neighbors); use Euclidean distance as your distance measure. **Show intermediate steps**.

5 Dimensionality reduction (8 p.)

5.1 PCA

Consider a data set consisting of 3 two-dimensional points: $(1, 1)$, $(2, 2)$, and $(3, 3)$.

Answer the following questions:

a) What is the first principal component?

b) Suppose we project our data to a one-dimensional space (to the first principal component). How much variance does the projection preserve?

Remember to justify your answers. An intuitive justification is enough; no need to do the math.

5.2 Neural networks

Explain briefly how neural networks can be used in dimensionality reduction.

6 Kernelized regression (8 p.)

Many linear regression and classification methods can be transformed to handle non-linear data. Recall that in linear regression we model the label $y_i \in \mathbb{R}$ using a linear function of the input vector $\mathbf{x}_i \in \mathbb{R}^d$:

$$y_i = \boldsymbol{\beta}^T \mathbf{x}_i + \epsilon_i,$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ is the weight vector and $\epsilon_i \sim N(0, \sigma^2)$ is a noise term.

Typically, one tries to find parameters $\boldsymbol{\beta}$ that minimize a quadratic loss function. Assuming that our data consists of n pairs $(\mathbf{x}_i, y_i)$ and L_2-regularization, we get the following loss function:

$$E(\boldsymbol{\beta}) = \frac{1}{2} \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^T \mathbf{x}_i)^2 + \frac{1}{2} \lambda ||\boldsymbol{\beta}||^2,$$

where λ is a hyperparameter determining the strength of regularization. In other words, the goal is to find a parameter vector $\boldsymbol{\beta}$ such that the loss is

minimized, that is, we want to minimize the sum of the squared errors of the predictions and a complexity penalty.

We represent the data with a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ where row i consists of a data vector $\mathbf{x}_i^T$. Furthermore, class labels are stored in a vector $\mathbf{y} \in \mathbb{R}^n$ where ith element is y_i.

The optimal value for $\boldsymbol{\beta}$ can be found with the following formula:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_d)^{-1}\mathbf{X}^T\mathbf{y},$$

where $\mathbf{I}_d$ is a $d \times d$ identity matrix.

Linear methods can be made non-linear by transforming data to some higher dimensional space using a non-linear transformation and learning a linear model in that space. That is, for each data point $\mathbf{x}_i$, we compute $\phi(\mathbf{x}_i) \in \mathbb{R}^{d'}$ that transforms the point into a d'-dimensional space. We use $\mathbf{H} \in \mathbb{R}^{n \times d'}$ to denote the data matrix of the transformed data. In other words, the ith row of $\mathbf{H}$ is $\phi(\mathbf{x}_i)^T$.

Solving the linear regression problem in the new space, gives us the following weight vector:

$$\boldsymbol{\beta} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{1}_{d'})^{-1}\mathbf{H}^T\mathbf{y}.$$

Note that now $\boldsymbol{\beta}$ is d'-dimensional.

Sometimes it is useful the write the solution in a different form. Using a result called matrix inversion lemma (You can check the proof after the exam from here: `https://danieltakeshi.github.io/2016/08/05/a-useful-matrix-inverse-equality-for-ridge-regression/`), we note that

$$(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{1}_{d'})^{-1}\mathbf{H}^T = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T + \lambda\mathbf{I}_n)^{-1}.$$

Let us denote $\mathbf{B} = \mathbf{H}\mathbf{H}^T$. We note that $\mathbf{B}$ is an $n \times n$ matrix and $\mathbf{B}_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. Now we can write the optimal weight vector in a form

$$\boldsymbol{\beta} = \mathbf{H}^T(\mathbf{B} + \lambda\mathbf{I}_n)^{-1}\mathbf{y}.$$

Further, denoting $\boldsymbol{\gamma} = (\mathbf{B} + \lambda\mathbf{I}_n)^{-1}\mathbf{y}$, we can write $\boldsymbol{\beta} = \sum_{i=1}^n \gamma_i\phi(\mathbf{x}_i)$, where γ_i is the ith element of $\boldsymbol{\gamma}$.

Consider a new data point $\mathbf{x}$. Now the prediction for y is

$$y^* = \boldsymbol{\beta}^T\phi(\mathbf{x}).$$

Answer the following questions:

a) What is a kernel?

b) Consider the formulation of non-linear regression above. We are interested in predicting y given $\mathbf{x}$ but not interested in $\boldsymbol{\beta}$. Modify the formulation to create a kernelized version of the non-linear regression.

c) What is the benefit of using the kernelized version of non-linear regression that you created in b) compared to the formulation presented above?