INF283 Introduction to machine learning
Fall 2018

Exam

3.12.2018

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 Overfitting

Give short answers to the following questions:

a) What is overfitting?

b) Why is it a problem?

c) How can overfitting be detected?

d) How can overfitting be avoided? (list at least three ways)

1.2 Model selection and evaluation

Suppose we have performed polynomial regression with different degrees of a polynomial. Our goal is to find a model that predicts well labels of unseen objects. We have measured the performance of the learned models by computing the mean squared error on the training set and on a separate validation set. The errors for varying degree polynomials can be seen in the table below.

Degree	Training error	Validation error
1	10	12
2	5	6
3	3	4
4	2	2.5
5	1.3	1.8
6	1.2	2.1
7	1.1	2.4
8	1.02	2.8
9	0.95	3.5
10	0.9	5

Give short answers to the following questions:

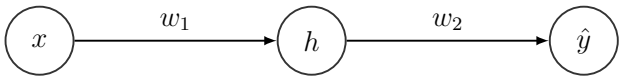
e) Which polynomial would you choose? Why?

f) Given the information that we have, can we get an unbiased estimate of the mean squared error of the chosen model on unseen data? Why/why not?

2 Neural networks (8 p.)

Consider the following simple neural network:

We have a one-dimensional input $x \in \mathbb{R}$ and a one-dimensional output $y \in \mathbb{R}$. Furthermore, we have one hidden layer consisting of one neuron and ReLU activation function. The output layer is linear.



That is, we have $z = w_1 x$, $h = f(z)$ where $f(z) = \max(0, z)$ and $\hat{y} = w_2 h$. We consider squared loss $L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$.

Suppose that initial weights have values $w_1 = 2$ and $w_2 = 3$. We have observed one data point with $x = 1$ and $y = 5$. Perform one update of parameters w_1 and w_2 using gradient descent with learning rate $\gamma = 0.1$. Show intermediate steps.

3 K-means clustering (8 p.)

You are given a data set $D = \{0, 1, 3, 7, 9\}$ consisting of 5 one-dimensional points. Find a 2-means clustering of D by simulating Lloyd's algorithm with initial cluster centers 3 and 7. Show intermediate steps.

4 PCA (8 p.)

Explain what principal component analysis (PCA) does. What is the goal? What is the interpretation of the output? (Feel free to use illustrations to clarify your point; no need to describe the algorithm)

5 Independencies in Bayesian networks (8 p.)

Consider the following DAG:

List all pairs of variables that are d-separated by some set of variables in the DAG; for each pair of d-separated variables, give one set that d-separates those variables.

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 $\mathbf{x}_i \in \mathbb{R}^d$:

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

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

Typically, one tries to find parameters $\mathbf{w}$ 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:

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

where λ is a hyperparameter determining the strength of regularization. In other words, the goal is to find a parameter vector $\mathbf{w}$ 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 $\mathbf{w}$ can be found with the following formula:

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

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 $\mathbf{\Phi} \in \mathbb{R}^{n \times d'}$ to denote the data matrix of the transformed data. In other words, the ith row of $\mathbf{\Phi}$ is $\phi(\mathbf{x}_i)^T$.

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

$$\mathbf{w} = (\mathbf{\Phi}^T \mathbf{\Phi} + \lambda \mathbf{1}_{d'})^{-1} \mathbf{\Phi}^T \mathbf{y}.$$

Note that now $\mathbf{w}$ 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

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

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

$$\mathbf{w} = \mathbf{\Phi}^T(\mathbf{H} + \lambda\mathbf{I}_n)^{-1}\mathbf{y}.$$

Further, denoting $\boldsymbol{\alpha} = (\mathbf{H} + \lambda\mathbf{I}_n)^{-1}\mathbf{y}$, we can write $\mathbf{w} = \sum_{i=1}^{n} \alpha_i\phi(\mathbf{x}_i)$, where α_i is the ith element of $\boldsymbol{\alpha}$.

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

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

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 $\mathbf{w}$. 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?