# INF264 Introduction to machine learning

Model solutions and grading criteria

10.2.2020

## 1 Basic concepts

1. A model is overfitting if it performs well on training data but it has bad generalization performance (it performs poorly on unseen validation data). Bad generalization performance is a problem because the main goal in machine learning is generalization.

2. A kernel if a function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. It is a dot product in some feature space: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. Kernels are useful because the data points do not have to be explicitly projected to $d'$-dimensional feature space but we can do our computations in the $d$-dimensional input space. This is computationally efficient.

3. Missing data can be handled, for example, by ignoring the data points with missing values or by imputing the values of missing features.

4. When classifying a particular test point, we start by finding $k$ training points that ar nearest to the test point. Then we predict the most common class among the $k$ points.

5. The $k$-means clustering tries to minimize the squared distance between data points and the cluster centers. This tend to lead "spherical" clusters.

**Grading**: max 2 points for each task. Full two points for mentioning all key points, 1 point for some correct elements. Deductions of 0.5-1 points for incorrect statements.

# 2 Boosting

1. Boosting is an ensemble method. In boosting, one learns a sequence of classifiers such that the data points are weighted based on how well the previous classifiers classify them.

2. (a) Repeat $k$ times:
   i. Train a classifier
   ii. Increase the weight of the misclassified points
   (b) Let the learned classifiers vote

3. Boosting works with simple base learners (high bias, low variance). By combining several simple classifiers, boosting can decrease bias and therefore increase accuracy.

**Grading**:
What is boosting 4p., Pseudocode 3p., Why does it work 3p.

# 3 Neural networks

Forward pass:

$z = w_1 x = 3 \cdot 1 = 3$
$h = f(z) = \max(0, 3) = 3$
$\hat{y} = w_2 h = 2 \cdot 3 = 6$

Backward pass:

We are going to need the following derivatives:
$\frac{\partial L}{\partial \hat{y}} = -(y - \hat{y}) = \hat{y} - y$
$\frac{\partial \hat{y}}{\partial w_2} = h$
$\frac{\partial \hat{y}}{\partial h} = w_2$

$$\frac{\partial h}{\partial z} = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

$\frac{\partial z}{\partial w_1} = x$

The compute the gradient, we use backpropagation. Using chain rule, we get the following partial derivatives:

$$\begin{aligned}\frac{\partial L}{\partial w_1} &= \frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial h}\frac{\partial h}{\partial z}\frac{\partial z}{\partial w_1} \\ &= (\hat{y} - y) \cdot w_2 \cdot 1 \cdot x \\ &= (6 - 5) \cdot 2 \cdot 1 \cdot 1 \\ &= 2\end{aligned}$$

and

$$\begin{aligned}\frac{\partial L}{\partial w_2} &= \frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial w_2} \\ &= (y - \hat{y}) \cdot h \\ &= (6 - 5) \cdot 3 \\ &= 3\end{aligned}$$

Updates with gradient descent:

$$\begin{aligned}w_1 &\leftarrow w_1 - \gamma\frac{L}{\partial w_1} \\ &= 3 - 0.1 \cdot 2 \\ &= 2.8\end{aligned}$$

$$\begin{aligned}w_2 &\leftarrow w_2 - \gamma\frac{L}{\partial w_2} \\ &= 2 - 0.1 \cdot 3 \\ &= 1.7\end{aligned}$$

**Grading**:
Forward pass 3p. Backpropagation 4p. Parameter update 3p.
Deductions:
Errors in derivatives 0.5-1p. Numerical errors 0.5-1p.

# 4 PCA

1. PCA tries to find a projection that maximizes variance
2. (a) Center the data
   (b) Calculate the covariance matrix of the centered data
   (c) Find the eigenvalues of the covariance matrix
   (d) Select $m$ eigenvectors that correspond to the $m$ largest eigenvalues to create a projection matrix
   (e) Use this matrix to project data to a new subspace
3. The first principal component is the direction that maximizes variance and the second principal component is orthogonal to the first one. The principal components are shown in Figure 1.
4. The data will lie on a line; see Figure 2.
5. The projected data will lie on a one-dimensional subspace. We have lost the variability to the direction of the second principal component; see Figure 3.

**Grading**:
Criterion 2p., Pseudocode 3p., Principal components 2p., Data projected to the first PC 1p., Data projected back to 2D 2p.

# 5 Model selection

We start by dividing the data into three non-overlapping sets: training, validation, and test. We can use, for example, 1500 data points for training, 250 data points for validation and 250 data points for testing.

We train all the models on the training data. Then, we use the trained models to make predictions about the value of $y$ given $\mathbf{x}$ on the validation data. We compare the predictions to the true labels. As we are solving the classification task, we can use, for example, accuracy as the performance measure. Then, we select the model with the highest accuracy on the validation data. (Alternative, we could skipped dividing the data into separate training and validation sets and used cross-validation.)

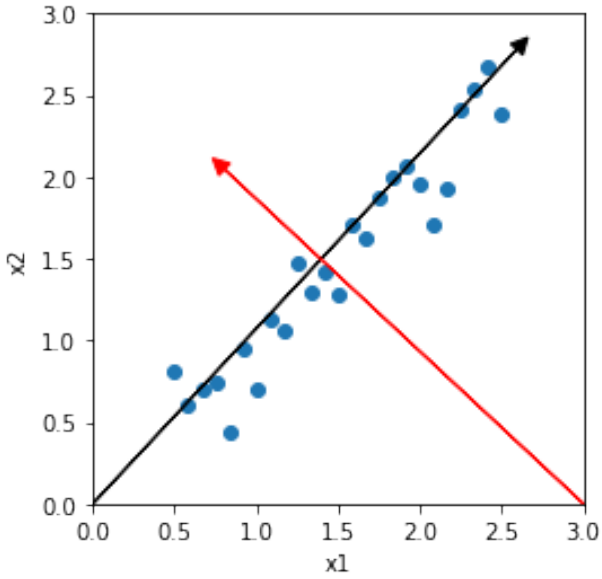


Figure 1: Principal components. The first component is black and the second components is red.

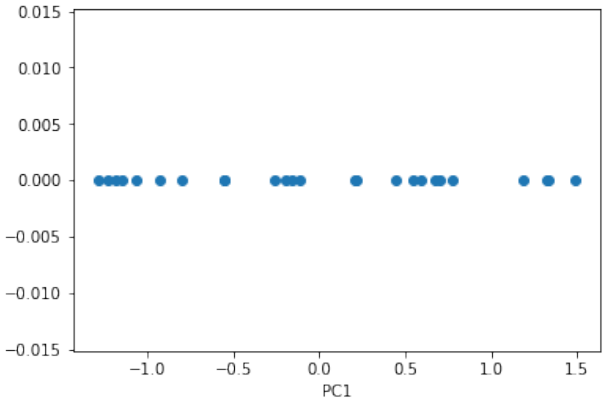


Figure 2: Data projected to one dimensional space.

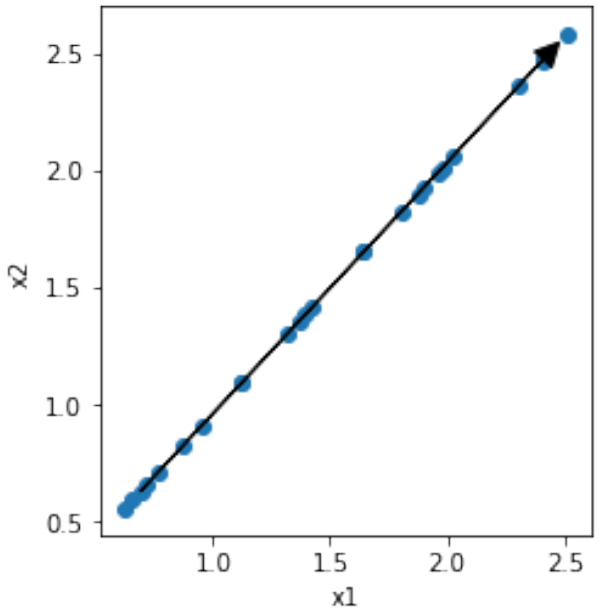


Figure 3: Data projected back to the original space.

Due to model selection, validation accuracy over-estimates generalization accuracy. Thus, we take the selected classifier and make predictions on the test data. The accuracy on the test set gives as an unbiased estimate of the accuracy on unseen data

**Grading**:

Separate training, validation, and test sets 3p., Train on the training data 1p., Select the model with the smallest validation error 3p., Generalization error from the test data 3p.

If cross-validation is used:

Separate training and test sets 3p., Cross-validation done properly 4p., Generalization error from the test data 3p.