Machine Learning for Finance (FIN 570)
Midterm Exam (Online / Take-home)

Instructor: Jaehyuk Choi

2019-20 Module 3 (2020. 4. 7.)

1. (3 points) Below is a series of steps for handling data and applying a learning method. Catch any incorrect step(s) and correct them.
   (a) The data set was divided into training vs test data set with 8:2 ratio.
   (b) The feature data X in the training set is standardized to
   $$x = \frac{(X - \mu_{\text{TR}})}{\sigma_{\text{TR}}},$$
   where μ_{TR} and σ_{TR} are the mean and standard deviation of X in the training set.
   (c) Using the standardized training set x, we fit a learning method, $F(x)$.
   (d) To measure the performance on the test set, we also standardize the feature X in the test set by
   $$x = \frac{(X - \mu_{\text{TE}})}{\sigma_{\text{TE}}},$$
   where μ_{TE} and σ_{TE} are the mean and standard deviation of X in the test set.
   (e) Finally, we measure the error rate by evaluating $F(x)$ for all x values in the test set.

   Solution: The step (d) is not correct. For the standardization of the test set, you need to use the parameters determined from the training set, i.e., μ_{TR} and σ_{TR}.

2. (2 points) For the tree-based learning models (e.g., decision tree, random forest, AdaBoost), you do not need to standardize (or normalize) the features. For example, see the code on page 230 of the **PML** textbook. The `StandardScaler()` function is applied to logistic regression and K-NN (in pipeline) but not to decision tree. Explain why.

Solution: In decision tree, each step is about finding a feature X_i and a branching value d such that the tree is branched either $X_i > d$ or $X_i < d$. Let $F(x)$ be a monotonic function. Standardization, `StandardScaler()`, is an example of $F(x)$. Then, $F(x)$ preserves order:

$$X_{1i} < X_{2i} \iff F(X_{1i}) < F(X_{2i}).$$

Therefore, the branch $X_i > d$ is equivalent to $F(X_i) > F(d)$ (the same hold for $<$) and the result of the decision tree is unchanged even if we use the transformed data, $F(X)$.

3. (7 points) This question demonstrates with a toy example why you need regularization in logistic regression. The sample data, X and y, is as below.

Observation No.	1	2	3	4
X	-2	-1	1	2
y	False (0)	False (0)	True (1)	True (1)

(a) (2 points) Calculate the log-likelihood function, $\log L(w_0, w_1)$, for the given data set. The logit function is given by

$$\phi(t) = \frac{1}{1+e^{-t}} \quad \text{where} \quad t = w_0 + w_1 X.$$

(b) (2 points) From the symmetry of the data set, we can simply assume $w_0 = 0$. Find the value of w_1 that maximize the log-likelihood, $\log L(w_1)$. Do you encounter any problem? How can you avoid the problem you encounter?

(c) (3 points) If the sample data is changed to

Observation No.	1	2	3	4
X	-2	-1	1	2
y	False (0)	True (1)	False (0)	True (1)

,

Do you encounter the same problem in (b)? What makes the difference (if any) from (b)? Find an **approximate** value of w_1 that maximize the log-likelihood.

Solution:

(a) The log-likelihood is given as below. The terms are arranged in the order of the observation number in the data set.

$$\begin{aligned}\log L(w_0, w_1) &= \log\left(1 - \frac{1}{1+e^{-w_0+2w_1}}\right) + \log\left(1 - \frac{1}{1+e^{-w_0+w_1}}\right) \\ &\quad + \log\left(\frac{1}{1+e^{-w_0-w_1}}\right) + \log\left(\frac{1}{1+e^{-w_0-2w_1}}\right) \\ &= -\log\left(1+e^{w_0-2w_1}\right) - \log\left(1+e^{w_0-w_1}\right) - \log\left(1+e^{-w_0-w_1}\right) - \log\left(1+e^{-w_0-2w_1}\right)\end{aligned}$$

(b) With $w_0 = 0$, the log-likelihood function is simplified to

$$\log L(w_1) = -2\log\left(1 + e^{-w_1}\right) - 2\log\left(1 + e^{-2w_1}\right).$$

The function is a monotonic increasing function of w_1. Therefore, the log-likelihood is maximized ad $w_1 \to \infty$. To avoid the divergence of w_1, we add the regularization terms to the loss function:

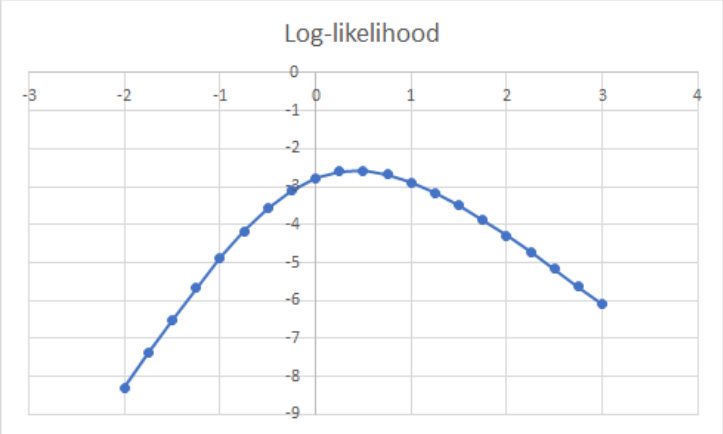
$$J(w_1) = -\log L(w_1) = 2\log\left(1 + e^{-w_1}\right) + 2\log\left(1 + e^{-2w_1}\right) + \lambda_1|w_1| + \lambda_2 w_1^2.$$

Then, we a finite value of w_1 will minimize the loss function.

(c) For the new data set, the log-likelihood is given by

$$\log L(w_1) = -2\log\left(1 + e^{w_1}\right) - 2\log\left(1 + e^{-2w_1}\right).$$

The function has its maximum even without the help of the regularization term. The difference is that, while the y values can be completely separable in the original data set, they are not separable in the modified data set. The log-likelihood has its maximum at around $w_1 \approx 0.5$.



4. (6 points) **(PCA)** The data matrix X and classification response y are given by

$$X = \begin{bmatrix} 2 & -3 \\ 6 & 1 \\ 5 & 0 \\ -1 & 4 \\ 3 & 3 \end{bmatrix} \quad y = \begin{bmatrix} \text{True} \\ \text{False} \\ \text{False} \\ \text{True} \\ \text{False} \end{bmatrix}.$$

(a) (2 points) Find the covariance matrix of X. Make sure to de-mean (i.e., subtract mean) each feature of X.

(b) (2 points) Find the two PCA components and corresponding eigenvalues (in the right order). What is the variance explained ratio of the first PCA component?

(c) (2 points) Transform the data matrix X using the PCA vectors you find from (b). Which PCA component (if you choose one between the two) is better to be used for the classification problem?

Solution:

(a) After removing the mean $\mu(X) = [3, 1]$, the covariance matrix is obtained as

$$\Sigma = \frac{1}{5}\begin{bmatrix} 30 & -10 \\ -10 & 30 \end{bmatrix} = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}.$$

(The answer with $1/4$ instead of $1/5$ is also acceptable.)

(b) The eigenvalue and eigenvector pairs are given by

$$v_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 \\ -1 \end{bmatrix},\ \lambda_1 = 8, \quad v_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} +1 \\ +1 \end{bmatrix},\ \lambda_2 = 4$$

We can verify that the sum of the feature variances is

$$\Sigma_{11} + \Sigma_{22} = \lambda_1 + \lambda_2 = 12.$$

The variance explained ratio by the first PCA component is $8/12 = 66.7\%$.

(c) The transformation matrix W is given by

$$W = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},$$

and the matrix X is transformed to

$$X' = XW = \frac{1}{\sqrt{2}}\begin{bmatrix} 3 & -5 \\ 3 & 3 \\ 3 & 1 \\ -7 & -1 \\ -2 & 2 \end{bmatrix} \quad y = \begin{bmatrix} \text{True} \\ \text{False} \\ \text{False} \\ \text{True} \\ \text{False} \end{bmatrix}.$$

The second PCA component is better for the classification because it can perfectly classify the response as (the second PCA < 0). This is an example showing that the first PCA component is not always the best choice.

5. (3 points) **Occam's razor** (or **Ockham's razor** or **law of parsimony**) is a principle from philosophy. Suppose there exist two explanations for an occurrence. In this case **the one that requires the smallest number of assumptions** is usually correct. Another way of saying it is that the more assumptions you have to make, the more unlikely an explanation. Occam's razor applies especially in the philosophy of science, but also more generally. (Simple Wikipedia)

For example, suppose that two trees have fallen down during a windy night. Think about these two possible explanations:

- The wind has blown them down.
- Two meteorites have each taken one tree down and, after striking the trees, hit each other removing any trace of themselves.

Even though both are possible, several other unlikely things would also need to happen for the meteorites to have knocked the trees down, for example: they would have to hit each other and not leave any marks. In addition, meteorites are fairly rare. Since this second explanation needs several assumptions to all be true, it is probably the wrong answer. Occam's razor tells us the wind blew the trees down, because this is the simplest answer therefore probably the right one.

Many desirable practices in machine learning we learned in class is consistent with Occam's razor. Give a few examples and explain why.

Solution: Reference: This question was motivated by a question from the 2015 exam in Introduction to Machine Learning at U.C. Berkeley.

Occam's razor tells us that complicated machine learning models are not likely the true pattern we are looking for. Any practice in machine learning that keeps models from growing complexity is consistent with Occam's razor. Examples are as below:

- Feature selection (using LASSO, random forest, etc)
- Feature extraction/dimensionality reduction: PCA, LDA, etc
- Regularization: L1 regularization and limiting tree depth or number of leaves are direct examples as they limit the number of features. All other regularization method (e.g., L2, dropout, etc) are also consistent with Occam's razor.

6. (6 points) You are evaluating different binary classification models. The models should be evaluated based on the following criteria:

   - Must have a recall rate of at least 80%.
   - Must have a false positive rate (FPR) of 10% or less
   - Must minimize business costs from incorrect classification. Assume that a false positive result is 5 times more costly than a false negative result.

   After creating four binary classification models (M1, $\cdots$, M4), you obtain the corresponding confusion matrices.

M1)

Confusion Matrix		Predicted		
		P^*	N^*	Total
Actual	P	78	22	100
	N	9	91	100

M2)

Confusion Matrix		Predicted		
		P^*	N^*	Total
Actual	P	90	10	100
	N	4	96	100

M3)

Confusion Matrix		Predicted		
		P^*	N^*	Total
Actual	P	79	21	100
	N	1	99	100

M4)

Confusion Matrix		Predicted		
		P^*	N^*	Total
Actual	P	82	18	100
	N	2	98	100

(a) (3 points) What is the best model in terms of the requirements above?

(b) (3 points) If you consider F1-score instead of the above requirements above, what is the best model?

Solution: Reference: The setting and part (a) of this question is from sample question 5 from AWS Certified Machine Learning – Specialty.

First, we compute various measures (recall, PFR, business cost, and precision) for the models:

Model	Recall	FPR	Business Cost	Precision	F-1 score
M1	78%	9%	$5 \times 9 + 22 = 67$	89.7%	83.4%
M2	90%	4%	$5 \times 4 + 10 = 30$	95.7%	92.8%
M3	79%	1%	$5 \times 1 + 21 = 26$	98.8%	87.8%
M4	82%	2%	$5 \times 2 + 18 = 28$	97.6%	89.1%

(a) Based on the recall and FPR for each model computed above, Models **M2** and **M4** satisfy the requirements for recall and FPR. But **M4** incurs less business cost.

(b) Model **M2** has the best F-1 score.

7. (3 points) The financial crime risk team in HSBC uses logistic regression to build a fraud detection model. While the model accuracy is 99%, the model detects only 10% of the fraud cases because of extreme class imbalance. Which of the following actions will **definitively** make the model detect more than 10% of fraud cases?

   A. Using under-sampling to balance the data set

   B. Using regularization to reduce overfitting

   C. Using over-sampling to balance the data set

   D. Decreasing the probability threshold for classification

Solution: Reference: This question is modified from sample question 6 from AWS Certified Machine Learning – Specialty.

D. – Decreasing the class probability threshold makes the model more sensitive and, therefore, marks more cases as the positive class, which is fraud in this case. This will increase the likelihood of fraud detection. However, it comes at the price of lowering precision. This is covered in the Discussion section of the paper at this paper.

All the rest items are good practices in machine learning, but they cannot guarantee that the ratio of the fraud will increase.