Machine Learning for Finance (FIN 570)
Midterm Exam

Instructor: Jaehyuk Choi

2021-22 Module 1 (2021. 11. 1.)

The acronyms are defined same as in the class. For example, machine learning (**ML**), logistic regression (**LR**), principal component analysis (**PCA**), support vector machine (**SVM**), neural network (**NN**), etc. If you are not sure, please ask.

1. ($3 = 0.5 \times 6$ points) Classify the following ML algorithms as supervised (**S**) or unsupervised (**U**).
   (a) SVM
   (b) Linear discriminant analysis (LDA)
   (c) Random forest
   (d) PCA
   (e) NN
   (f) Word2Vec (in sentiment analysis)

   Solution:
   (a) SVM: **S**upervised
   (b) LDA: **S**upervised
   (c) Random forest: **S**upervised
   (d) PCA: **U**nsupervised
   (e) NN: **S**upervised
   (f) Word2Vec: **U**nsupervised

2. ($5 = 1 \times 5$ points) **(Hyperparameters)** For the following ML methods, give one example of hyperparameters and explain what will happen to the bias and variance when you increase this parameter value.

Example: PCA as a dimensionality reduction. **Answer:** The number of PCA components to use. Increasing the number of PCA components will increase variance and decrease bias.

(a) K-NN for classification.

(b) SVM with RBF kernel.

(c) Random forest.

(d) NN.

(e) LR with L_1 regularization.

Solution:

(a) K-NN for classification.

- The number of neighbors, K. Increasing K will increase variance and decrease bias.

(b) SVM with RBF kernel.

- The violation budget C (`sklearn` parameter C). Increasing C (decreasing `sklearn` C) will decrease variance and increase bias.

(c) Random forest.

- Max number of leaves. More leaves will increase variance and decrease bias.
- Max number of branching. More branching will increase variance and decrease bias.

(d) NN.

- Number of hidden layers. More layers will increase variance and decrease bias.
- Number of nodes in a hidden layer. More nodes will increase variance and decrease bias.

(e) LR with L_1 regularization.

- The regularization penalty λ (`sklearn` parameter C). Increasing λ (decreasing C) will decrease variance and increase bias.

3. (3 points) **(SVD v.s. PCA)** Suppose that $\boldsymbol{X}$ is the data matrix of size $n \times p$ (n samples, p features), and that the SVD of $\boldsymbol{X}$ is given by

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T,$$

where the singular values (i.e, the diagonals of $\boldsymbol{S}$) are $s_1 \geq \cdots \geq s_n \geq 0$. Also assume that the data has zero mean for each feature, so the covariance matrix of X is given by

$$\boldsymbol{\Sigma} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} \quad (p \times p).$$

If λ_k ($\lambda_1 \geq \cdots \geq \lambda_p$) and $\boldsymbol{w}_k$ are the eigenvalues and eigenvectors of $\boldsymbol{\Sigma}$, respectively,

$$\boldsymbol{\Sigma}\boldsymbol{w}_k = \lambda_k \boldsymbol{w}_k,$$

we know that $\boldsymbol{w}_k$ are the PCA components of the data. How is λ_k and $\boldsymbol{w}_k$ are related to the SVD of $\boldsymbol{X}$, i.e., $\boldsymbol{U}$, $\boldsymbol{S}$ (or s_k), $\boldsymbol{V}$? Explain in detail.

Solution: Plugging $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ into the equation for the covariance matrix,

$$\boldsymbol{\Sigma} = \frac{1}{n}(\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T)^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T.$$

This implies

$$\boldsymbol{\Sigma}\boldsymbol{V} = \frac{1}{n}\boldsymbol{V}\boldsymbol{S}^2 \quad \text{and} \quad \boldsymbol{\Sigma}\boldsymbol{v}_k = \frac{s_k^2}{n}\boldsymbol{v}_k,$$

where $\boldsymbol{v}_k$ is the k-th column vector of $\boldsymbol{V}$. Therefore,

$$\lambda_k = \frac{s_k^2}{n} \quad \text{and} \quad \boldsymbol{w}_k = \boldsymbol{v}_k \text{ (the } k\text{-th column vector of } \boldsymbol{V}).$$

4. (3 points) **(Ridge regression)** The linear regression with L_2 regularization, called the ridge regression, minimizes the following cost function:

$$J(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2,$$

where $\boldsymbol{y}$ is the column vector of the dependent variable, $\boldsymbol{X}$ is the data matrix, and $\boldsymbol{w}$ is the column vector of weights, and $\|\boldsymbol{w}\|^2$ is the square of the vector norm, $\|\boldsymbol{w}\|^2 = \boldsymbol{w}^T\boldsymbol{w}$. Find the weight vector $\hat{\boldsymbol{w}}$ that minimizes $J(\boldsymbol{w})$.
(Hint: solve $\frac{\partial}{\partial \boldsymbol{w}}J(\boldsymbol{w}) = 0$. When $\lambda = 0$, your answer should be the same as the weight of the simple linear regression.)

Solution: The derivative of $J(\boldsymbol{w})$ is

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = -\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \lambda\boldsymbol{w}.$$

The weight $\hat{\boldsymbol{w}}$ satisfying $\frac{\partial}{\partial \boldsymbol{w}}J(\hat{\boldsymbol{w}}) = 0$ is,

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}.$$

When $\lambda = 0$, $\hat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$ is same as that in the simple linear regression. We can observe that $\hat{\boldsymbol{w}} \to 0$ as $\lambda \to \infty$.

5. (4 points) **(Tf-idf)** We want to run a sentiment analysis on the three abstracts of the financial ML papers introduced in the class. Each abstract is a *document*.

   (a) Please calculate the tf-idf vectors representing the following terms:

   (corporate, machine, neural, networks, predictive, risk).

   For the inverse document frequency, use the definition:

   $$idf(t) = \log \frac{1 + n_D}{1 + df(t)}.$$

   For others functions, follow the definition from **PML**. (For log evaluation, you can use the approximation $\log(1 + x) \approx x$. Please take this opportunity to read the abstracts carefully.)

   (b) By calculating the cosine similarity of the tf-idf vectors obtained from (a), identify the most similar pair of abstracts.

D1 Can algorithms assist firms in their decisions on nominating corporate directors? Directors predicted by algorithms to perform poorly indeed do perform poorly compared to a realistic pool of candidates in out-of-sample tests. Predictably bad directors are more likely to be male, accumulate more directorships, and have larger networks than the directors the algorithm would recommend in their place. Companies with weaker governance structures are more likely to nominate them. Our results suggest that machine learning holds promise for understanding the process by which governance structures are chosen and has potential to help real-world firms improve their governance.

D2 We perform a comparative analysis of machine learning methods for the canonical problem of empirical asset pricing: measuring asset risk premiums. We demonstrate large economic gains to investors using machine learning forecasts, in some cases doubling the performance of leading regression-based strategies from the literature. We identify the best-performing methods (trees and neural networks) and trace their predictive gains to allowing nonlinear predictor interactions missed by other methods. All methods agree on the same set of dominant predictive signals, a set that includes variations on momentum, liquidity, and volatility.

D3 We show that machine learning methods, in particular, extreme trees and neural networks (NNs), provide strong statistical evidence in favor of bond return predictability. NN forecasts based on macroeconomic and yield information translate into economic gains that are larger than those obtained using yields alone. Interestingly, the nature of unspanned factors changes along the yield curve: stock- and labor-market-related variables are more relevant for short-term maturities, whereas output and income variables matter more for longer maturities. Finally, NN forecasts correlate with proxies for time-varying risk aversion and uncertainty, lending support to models featuring both channels.

Solution:

(a) We construct the term frequency $tf(D,t)$, inverse document frequency $idf(t)$, and $tf-idf(D,t)$ as below:

t	$tf(D,t)$			$df(t)$	$idf(t)$	tf-$idf(D,t)$		
	D1	D2	D3			D1	D2	D3
corporate	1	0	0	1	$\log 2 \approx 1$	1	0	0
machine	1	2	1	3	0	0	0	0
neural	0	1	1	2	$\log 4/3 \approx 1/3$	0	1/3	1/3
networks	1	1	1	3	0	0	0	0
predictive	0	2	0	1	$\log 2 \approx 1$	0	2	0
risk	0	1	1	2	$\log 4/3 \approx 1/3$	0	1/3	1/3

- **tf-idf for D1**: (1, 0, 0, 0, 0, 0)
- **tf-idf for D2**: (0, 0, 1/3, 0, 2, 1/3)
- **tf-idf for D3**: (0, 0, 1/3, 0, 0, 1/3)

(b) The cosine similarities for D1–D2 and D1–D3 are zero. The similarity for D2–D3 is

$$\frac{2/9}{\sqrt{4+2/9}\cdot\sqrt{2/9}} = \frac{2}{\sqrt{76}}.$$

The D2–D3 is the most similar pair among the three abstracts.

6. ($4 = 2 \times 2$ points) **(NN and activation function)** Consider an NN model for classification. Explain why the following functions are **not** proper for an activation function in the hidden layers.

(a) $\phi(x) = H(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases}$

(b) $\phi(x) = x$

Solution: See online source.

(a) $\phi'(x) = 0$ except at $x = 0$. Therefore, the gradient always vanish.

(b) Two reasons. Only one is enough.

- $\phi'(x) = 1$. Because $\phi'(x)$ does not change, it is difficult to learn the optimal weight w.
- Suppose an NN with one hidden layer.

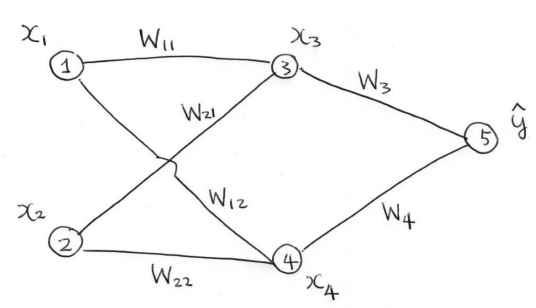
$$\boldsymbol{a}^{(2)} = \phi(\boldsymbol{a}^{(1)} = \boldsymbol{x}^{(0)}\boldsymbol{W}^{(1)})\boldsymbol{W}^{(2)},$$

where $\boldsymbol{W}^{(1)}$ is the weight matrix connecting the input (0-th) to the 1st layer and $\boldsymbol{W}^{(2)}$ connecting the 1st to 2nd layer. When $\phi(x) = x$,

$$\boldsymbol{a}^{(2)} = \boldsymbol{x}^{(0)}\boldsymbol{W}^{(1)}\boldsymbol{W}^{(2)},$$

so $\boldsymbol{W}^{(1)}\boldsymbol{W}^{(2)}$ plays the role of the the weight matrix directly connecting the input to the 2nd layer. Therefore, the hidden (1-st) layer becomes unnecessary.

7. ($8 = 2 \times 4$ points) **(NN and backpropagation)** Consider an NN model consist of $2 \times 2 \times 1$ nodes and sigmoid activation function.



Node 1, 2 (input layer) : x_1 and x_2

Node 3 (hidden layer) : $a_3 = x_1 w_{11} + x_2 w_{21}, \quad x_3 = \phi(a_3)$

Node 4 (hidden layer) : $a_4 = x_1 w_{12} + x_2 w_{22}, \quad x_4 = \phi(a_4)$

Node 5 (output) : $a_5 = x_3 w_3 + x_4 w_4, \quad \hat{y} = \phi(a_5),$

Here, $\phi(t)$ is the sigmoid function. The loss function for a sample (x_1, x_2) and y is given by

$$\begin{aligned} J(w_{11}, \cdots, w_3, w_4) &= -y \log \hat{y} - (1-y)\log(1-\hat{y}) \\ &= -y \log \phi(a_5) - (1-y)\log(1-\phi(a_5)). \end{aligned}$$

Let us define the "error" at the nodes $j = 3$, 4, and 5 as

$$\delta_j = \frac{\partial J}{\partial a_j} \quad \text{for} \quad j = 3, 4, 5.$$

(a) Show that $\delta_5 = \hat{y} - y$ at the output node ($j = 5$),

(b) Using the chain rule and δ_5, find

$$\frac{\partial J}{\partial w_3} \quad \text{and} \quad \frac{\partial J}{\partial w_4}.$$

(c) Find δ_3 and δ_4.

(d) Find

$$\frac{\partial J}{\partial w_{11}} \quad \text{and} \quad \frac{\partial J}{\partial w_{12}}.$$

(You may also solve $\frac{\partial J}{\partial w_{21}}$ and $\frac{\partial J}{\partial w_{22}}$ in the same way, but you don't have to do it.)

Hint and instruction: You may use the sigmoid derivative property, $\phi'(t) = \phi(t)(1-\phi(t))$. Therefore, do not leave $\phi'(\cdot)$ in your answers.

Solution:

(a) Using that $\phi'(t) = \phi(t)(1-\phi(t))$,

$$\begin{aligned}\delta_5 = \frac{\partial J}{\partial a_5} &= -\frac{y}{\phi(a_5)}\phi'(a_5) + \frac{1-y}{1-\phi(a_5)}\phi'(a_5) \\ &= -y(1-\phi(a_5)) + (1-y)\phi(a_5) = \phi(a_5) - y = \hat{y} - y.\end{aligned}$$

(b)

$$\begin{aligned}\frac{\partial J}{\partial w_3} &= \frac{\partial J}{\partial a_5}\frac{\partial a_5}{\partial w_3} = \delta_5 x_3 = (\hat{y}-y)\, x_3 \\ \frac{\partial J}{\partial w_4} &= \frac{\partial J}{\partial a_5}\frac{\partial a_5}{\partial w_4} = \delta_5 x_4 = (\hat{y}-y)\, x_4\end{aligned}$$

(c)

$$\begin{aligned}\delta_3 = \frac{\partial J}{\partial a_3} &= \frac{\partial J}{\partial a_5}\frac{\partial a_5}{\partial x_3}\frac{\partial x_3}{\partial a_3} = \delta_5 w_3 \phi'(a_3) = \delta_5 w_3 \phi(a_3)(1-\phi(a_3)) \\ \delta_4 = \frac{\partial J}{\partial a_4} &= \frac{\partial J}{\partial a_5}\frac{\partial a_5}{\partial x_4}\frac{\partial x_4}{\partial a_4} = \delta_5 w_4 \phi'(a_4) = \delta_5 w_4 \phi(a_4)(1-\phi(a_4))\end{aligned}$$

(d)

$$\begin{aligned}\frac{\partial J}{\partial w_{11}} &= \frac{\partial J}{\partial a_3}\frac{\partial a_3}{\partial w_{11}} = \delta_3 x_3 = \delta_5 w_3 \phi(a_3)(1-\phi(a_3))\, x_3 \\ \frac{\partial J}{\partial w_{12}} &= \frac{\partial J}{\partial a_4}\frac{\partial a_4}{\partial w_{12}} = \delta_4 x_4 = \delta_5 w_4 \phi(a_4)(1-\phi(a_4))\, x_4.\end{aligned}$$