Imperial College Business School	**(2h)**

Machine Learning and Finance - Final Exam -

Instructors: Arnaud de Servigny & Hachem Madmoun

Notations:

- For all $z \in \mathbb{R}^D$, the $\mathcal{L}^2$ norm on $\mathbb{R}^D$ of z is defined as follows: $||z||_2^2 = z^T z$
- The gradient of a function $f : \theta \in \mathbb{R}^D \mapsto \mathbb{R}$ at $\theta \in \mathbb{R}^D$ is denoted as follows $\nabla_\theta f(\theta) = \left(\frac{\partial f}{\partial \theta_1}(\theta), \dots, \frac{\partial f}{\partial \theta_D}(\theta)\right)$
- $\mathcal{M}_{n,p}(\mathbb{R})$ is the space of the matrices composed of n rows and p columns.
- $I_n \in \mathcal{M}_{n,n}(\mathbb{R})$ is the identity matrix of size n.
- **Convention**: D dimensional row vectors are considered $\mathcal{M}_{D,1}(\mathbb{R})$ matrices.

1 Building a factor model (60 marks)

Algorithmic trading strategies use factor models to quantify the relationship between the return of an asset and the sources of risk that are the main drivers of these returns.

We wish to predict the return of an asset based on M features called **factor premia**.

For each time step t in $\{1, \dots, T\}$, the return of the asset is denoted $r^{<t>}$, and the M features are denoted $(f_i^{<t>})_{1 \leq i \leq M}$.

In the two following sections, the training data is composed of the **feature matrix** F of shape (T, M) and the output observation matrix R of shape $(T, 1)$:

$$F = \underbrace{\begin{pmatrix} f_1^{<1>} & \dots & f_M^{<1>} \\ \vdots & \vdots & \vdots \\ f_1^{<T>} & \dots & f_M^{<T>} \end{pmatrix}}_{M \text{ columns},\ T \text{ rows}} \qquad R = \begin{pmatrix} r^{<1>} \\ \vdots \\ r^{<T>} \end{pmatrix}$$

1.1 Introducing a basic Regression Model

A **factor model** simply decompose the return of the asset at time t (denoted $r^{<t>}$) into the set of **factor premia** $(f_i^{<t>})_{1 \leq i \leq M}$ as follows:

$$\forall t \in \{1, \dots, T\} \quad r^{<t>} = \sum_{i=1}^{M} \beta_i f_i^{<t>} + \alpha + \epsilon \text{ with } (\beta_1, \dots, \beta_M, \alpha) \in \mathbb{R}^{M+1}, \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{1.1}$$

Question 1: What is the name of the model defined in equation 1.1 ? (3 marks)

Let $F^{<1>}, \dots, F^{<T>}$ be the rows of the matrix F and $\beta = (\beta_1, \dots, \beta_M, \alpha)^T \in \mathcal{M}_{M+1,1}(\mathbb{R})$ be the vector of all the parameters we want te estimate using the training data.

We want to re-write the equation 1.1 in a matrix form as follows:

$$\forall t \in \{1, \dots, T\} \quad r^{<t>} = \beta^T \tilde{F}^{<t>} + \epsilon, \text{ with } \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Question 2: Deduce $\tilde{\mathbf{F}}^{<t>}$ from $\mathbf{F}^{<t>}$ for all $\mathbf{t} \in \{\mathbf{1}, \dots, \mathbf{T}\}$. (3 marks)

Let $\tilde{F}$ be the matrix composed of the rows $\tilde{F}^{<t>} \quad \forall t \in \{1, \dots, T\}$

Question 3: What is the shape of the $\tilde{\mathbf{F}}$ matrix ? (3 marks)

Question 4: Show that the optimal vector of parameters $\beta^* \in \mathcal{M}_{\mathbf{M+1,1}}(\mathbb{R})$ is defined as follows:

$$\beta^* = \underset{\beta \in \mathcal{M}_{M+1,1}(\mathbb{R})}{\arg\min} \frac{1}{T} ||\tilde{F}\beta - R||_2^2 \quad \text{(6 marks)} \tag{1.2}$$

The gradient of the function $J(\beta) := \frac{1}{T} ||\tilde{F}\beta - R||_2^2$ with respect to β is given in equation 1.3

$$\nabla_\beta J(\beta) = \frac{2}{T}(\tilde{F}^T \tilde{F}\beta - \tilde{F}^T R) \tag{1.3}$$

We would like to optimize the function J using the **Batch Gradient Descent Algorithm**.

Question 5: What is the expression of the loss on each batch ? Describe the optimization process. (5 marks)

The following figure shows the prediction error (the Root Mean Square Error denoted RMSE) for each epoch for the training and validation set.

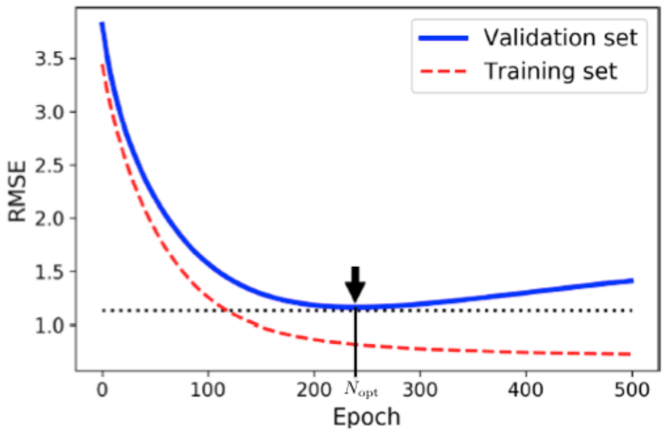


Figure 1: Training and validation error

Question 6: What is the problem highlighted in the the figure 1 and what should be the optimal number of epochs? (5 marks)

1.2 Introducing regularization techniques to the Regression Model

An important theoratical result of Machine Learning is the fact that a model's generalization error can be expressed as the sum of the three following errors: the **bias**, the **variance** and the **irreducible error**.

Question 7: Explain the bias-variance tradeoff. (6 marks)

One popular approach to control the overfitting problem is that of **regularization**, which involves the addition of a penalty term to the error function to discourage the regression coeffictions from reaching large values.

The added penalty turns the optimal linear regression coefficients into the solution to the following minimization problem:

$$\beta^*_{\text{reg}} = \underset{\beta_{\text{reg}} \in \mathcal{M}_{M+1,1}(\mathbb{R})}{\arg\min} \frac{1}{T} \left(||\tilde{F}\beta_{\text{reg}} - R||_2^2 + \lambda \mathcal{S}(\beta_{\text{reg}}) \right) \tag{1.4}$$

These **shrinkage methods** differ by how they calculate the penalty term. The most common versions for the linear regression model are the **ridge regression** and the **lasso regression**.

In this section, let us consider the **ridge regression**. The penalty term is then defined as follows:

$$\mathcal{S}(\beta_{\text{reg}}) = \beta_{\text{reg}}^T \beta_{\text{reg}}$$

Equation 1.4 becomes:

$$\beta^*_{\text{reg}} = \underset{\beta_{\text{reg}} \in \mathcal{M}_{M+1,1}(\mathbb{R})}{\arg\min} \frac{1}{T} \left(||\tilde{F}\beta_{\text{reg}} - R||_2^2 + \lambda \beta_{\text{reg}}^T \beta_{\text{reg}} \right) \tag{1.5}$$

Question 8: Show that the closed solution to the ridge regression problem defined in 1.5 is:

$$\beta^*_{\text{reg}} = (\tilde{F}^T \tilde{F} + \lambda I_{M+1})^{-1} \tilde{F}^T R \quad \text{(8 marks)}$$
$$(\text{Hint} : \forall z \in \mathbb{R}^D \ \forall A \in \mathcal{M}_{D,D}(\mathbb{R}) \quad \nabla_z (z^T A z) = (A + A^T) z)$$

Question 9: What would be the closed solution to the linear regression problem 1.2 ? (4 marks)

Question 10: Why did we choose an iterative learning algorithm to solve the linear regression problem 1.2 and a closed solution for the ridge regression problem 1.5 ? (5 marks)

Question 11: The hyperparameter λ controls the strength of the regularization. How would you choose the appropriate λ ? (6 marks)

Question 12: Describe three other methods used to prevent the overfitting problem in the context of Neural Networks or Tree Based Models. (6 marks)

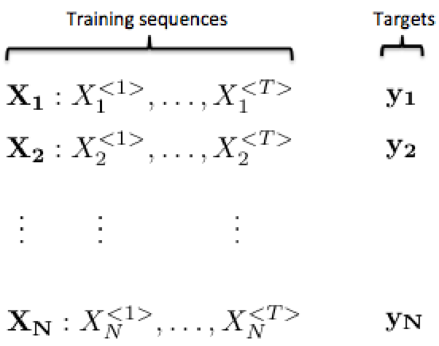
2 Building a Sentiment Analysis model (60 marks)

We wish to create a sentiment analysis model to classify financial news into three possible labels: **positive**, **negative** or **neutral**.

The training dataset is composed of N sentences $(X_i)_{1 \le i \le N}$. Each sentence is composed of T words.

Let V be the vocabulary size. The first step of the processing consists in creating a dictionary to map each word to a discrete category in $\{1, \ldots, V\}$.

We end up with N sequences $(X_i)_{1 \leq i \leq N}$ of categories $X_i^{<t>}$ in $\{1, \ldots, V\}$ for all $i \in \{1, \ldots, N\}$ and for all $t \in \{1, \ldots, T\}$ as shown in the following figure:



The three possible labels are encoded as follows: 0 for the negative sentiment, 1 for the neutral sentiment and 2 for the positive sentiment.

2.1 Using a Generative Classifier

In this second section, we would like to create a generative classifier. For that, we need to train three class conditional density functions, one for each target value $k \in \{0, 1, 2\}$

Each class conditional density function associated with the target $k \in \{0, 1, 2\}$ is parameterized by θ_k, which enables us to calculate $p_{\theta_k}(X|y = k)$ for a given a sequence $X = (X^{<1>}, \ldots, X^{<T>}) \in \{1, \ldots, V\}^T$.

2.1.1 Predicting the target

Question 13: On which data are we going to train each class conditional discrete density function ? (3 marks)

Question 14: Let $\mathbf{X} = (\mathbf{X}^{<1>}, \ldots, \mathbf{X}^{<T>}) \in \{\mathbf{1}, \ldots, \mathbf{V}\}^{\mathbf{T}}$ be a new sequence. Express $\mathbf{p}(\mathbf{y} = \mathbf{k}|\mathbf{X})$ as a function of $(\mathbf{p}(\mathbf{y} = \mathbf{j}))_{\mathbf{j} \in \{\mathbf{0},\mathbf{1},\mathbf{2}\}}$ and $(\mathbf{p}_{\theta_\mathbf{j}}(\mathbf{X}|\mathbf{y} = \mathbf{j}))_{\mathbf{j} \in \{\mathbf{0},\mathbf{1},\mathbf{2}\}}$ (6 marks)

2.1.2 Using a Hidden Markov Model as a class conditional discrete density estimator on the discrete categories

Let k be in $\{0, 1, 2\}$. We would like to use a Hidden Markov Model (HMM) as a class conditional discrete density estimator $p_{\theta_k}(X|y = k)$ (where $X = (X^{<1>}, \ldots, X^{<T>}) \in \{1, \ldots, V\}^T$).

Let M be the number of hidden states.

Question 15: What are the parameters $\theta_\mathbf{k}$ of the HMM class conditional discrete density estimator? (5 marks)

Question 16: By choosing reasonable values of V and M, compare the number of parameters of the HMM model with the number of parameters of a simple Markov Model (6 marks)

Question 17: What training method can we use to estimate the parameters of the HMM? (3 marks)

2.1.3 Using a Hidden Markov Model as a class conditional density estimator on the embedding vectors

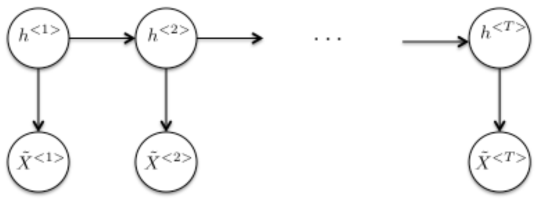
Instead of modeling $p_{\theta_k}(X|y=k)$ for $X = (X^{<1>}, \dots, X^{<T>}) \in \{1, \dots, V\}^T$, we would like to consider instead a class conditional density estimator on the sequences of some embedding vectors.

Each category in $\{1, \dots, V\}$ represents a word and can be mapped into a D-dimensional vector, where each dimension encodes part of the information about the corresponding word.

Question 18: Give two examples of unsupervised models used to create such embedding vectors and describe briefly one of them. (6 marks)

For each new sequence $X = (X^{<1>}, \dots, X^{<T>}) \in \{1, \dots, V\}^T$, we define $\tilde{X} = (\tilde{X}^{<1>}, \dots, \tilde{X}^{<T>})$ the sequence of the embedding vectors $\tilde{X}^{<t>} \in \mathbb{R}^D$ associated with the categories $X^{<t>} \in \{1, \dots, V\}$ for each $t \in \{1, \dots, T\}$.

We would like to use an HMM model in order to learn the distribution of a sequence of D-dimensional embedding vectors $\tilde{X} = (\tilde{X}^{<1>}, \dots, \tilde{X}^{<T>})$ as represented in the following figure:



For each $t \in \{1, \dots, T\}$, the embeding vector $\tilde{X}^{<t>} \in \mathbb{R}^D$ is associated with a hidden state $h^{<t>} \in \{1, \dots, M\}$.

Let m be in $\{1, \dots, M\}$. The emission probability distribution of the observation $X^{<t>}$ conditioned on the hidden state $h^{<t>} = m$ is parameterized by a multivariate normal distribution with a mean vector $\mu_m \in \mathbb{R}^D$ and a covariance matrix $\Sigma_m \in \mathcal{M}_{D,D}(\mathbb{R})$, as explained in equation 2.1

$$\forall t \in \{1, \dots, T\} \ \forall m \in \{1, \dots, M\} \quad X^{<t>}|h^{<t>} = m \sim \mathcal{N}_D(\mu_m, \Sigma_m) \tag{2.1}$$

Let $\tilde{\theta_k}$ be the set of the parameters of the HMM class conditional density estimator $p_{\tilde{\theta}_k}(\tilde{X}|y=k)$ (where $\tilde{X}$ is a sequence of embedding vectors $(\tilde{X}^{<1>}, \dots, \tilde{X}^{<T>})$)

Question 19: What are the parameters $\tilde{\theta_k}$ of the HMM class conditional density estimator on the embedding vectors? (6 marks)

Question 20: By choosing reasonable values for M, D and V, compare the number of parameters of the HMM model on the discrete observations in $\{1, \dots, V\}$ with the number of parameters of the HMM on the D-dimensional embedding vectors. (6 marks)

2.2 Using a Sequential Neural Network

We would like to use a **Sequential Neural Network** model with LSTM layers and pre-trained D-dimensional word vectors to perform the same classification of news.

Question 21: Define such a model by specifying the different layers, the different hyperparameters for each layer and how the shape of the data is changing after each layer transformation. (10 marks)

Question 22: What loss function should we use? (3 marks)

Question 23: Describe an appropriate optimizer with an adaptive learning rate. (6 marks)