Algorithmic Trading: Classifcation

Sebastian Jaimungal
University of Toronto

Jan, 2018

Classification

Classification

- The **classification problem** is as follows: given data

$$(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$$

where

$$\mathbf{X}_i = (X_{i1}, \ldots, X_{id}) \in \mathcal{X} \subset \mathbb{R}^d$$

are the **features**, $\mathcal{X}$ is called **feature space**, and

$$Y_i \in \mathcal{Y} = \{1, \ldots, K\} \text{ (the discrete set of } \textbf{classes})$$

are the observed **classifications**, determine a **classification rule**

$$h : \mathcal{X} \to \mathcal{Y}$$

which minimizes some **performance criteria**.

Classification

- **Generative classifiers** model the feature conditional on the class

$$\mathbb{P}(\mathbf{X} \mid Y = y)$$

- **Discriminative classifiers** model the class conditional on the feature

$$\mathbb{P}(Y = y \mid \mathbf{X})$$

Bayes Classifier

Bayes Classifier

- The **True Error Rate** of a classifier h is defined as

$$\mathcal{E}(h) = \mathbb{P}\left(\{h(\mathbf{X}) \neq Y\}\right)$$

- The **training error rate** of a classifier h is defined as

$$\widehat{\mathcal{E}}(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{h(\mathbf{X}_i) \neq Y_i\right\}$$

Bayes Classifier

- We aim to set a rule based on the probability that $Y = y$ given that $\mathbf{X} = \boldsymbol{x}$. To this end, note that

$$\mathbb{P}(Y = y \mid \mathbf{X} = \boldsymbol{x}) = \frac{f_{\mathbf{X}|Y=y}(\boldsymbol{x})\,\mathbb{P}(Y = y)}{\sum_{k=1}^{K} f_{\mathbf{X}|Y=k}(\boldsymbol{x})\,\mathbb{P}(Y = k)}$$

  where $f_{\mathbf{X}|Y=y}(\boldsymbol{x})$ denotes the conditional density of $\mathbf{X}$, conditional on $Y = y$.

- **Bayes Classification Rule** is to set

$$\begin{aligned} h(\boldsymbol{x}) &= \arg\max_{y} \mathbb{P}(Y = y \mid \mathbf{X} = \boldsymbol{x}) \\ &= \arg\max_{y} f_{\mathbf{X}|Y=y}(\boldsymbol{x})\,\mathbb{P}(Y = y) \end{aligned}$$

  i.e., it seeks assign a class which maximizes the probability that the class is observed at that point in feature space

Bayes Classifier

- The **class probabilities** $\pi_k = \mathbb{P}(Y = k)$ can be estimated empirically

$$\widehat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{Y_i = k\} = \frac{n_k}{n}$$

  i.e., the ratio of the number of observations of class k to all observations

Bayes Classifier

- The **conditional feature probabilities** $f_{\mathbf{X}|Y=k}(\boldsymbol{x})$ requires some modeling (NB: one could use kernel densities estimator)
- The simplest is to assume that the features are **multi-variate normal** conditional on Y, i.e.,

$$\mathbf{X}|_{Y=k} \sim \mathcal{N}(\boldsymbol{\mu}_k \; ; \; \boldsymbol{\Sigma}_k)$$

Bayes Classifier

- In this case, the **Bayes Classification Rule** becomes

$$h(\boldsymbol{x}) = \arg\max_k \left\{ \log \pi_k - \frac{1}{2} \log \det \boldsymbol{\Sigma}_k - \frac{1}{2} \underbrace{(\boldsymbol{x} - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)}_{\textbf{Mahalanobis distance}} \right\}$$

- Note: the classification boundaries are quadratic functions of the feature space.. and it is called a **Quadratic Discriminant Analysis** (QDA)

Bayes Classifier

- To complete the classification, we use the sample conditional means

$$\widehat{\boldsymbol{\mu}_k} = \frac{1}{n_k} \sum_{i=1}^{n} \mathbb{1}_{Y_i = k} \mathbf{X}_i$$

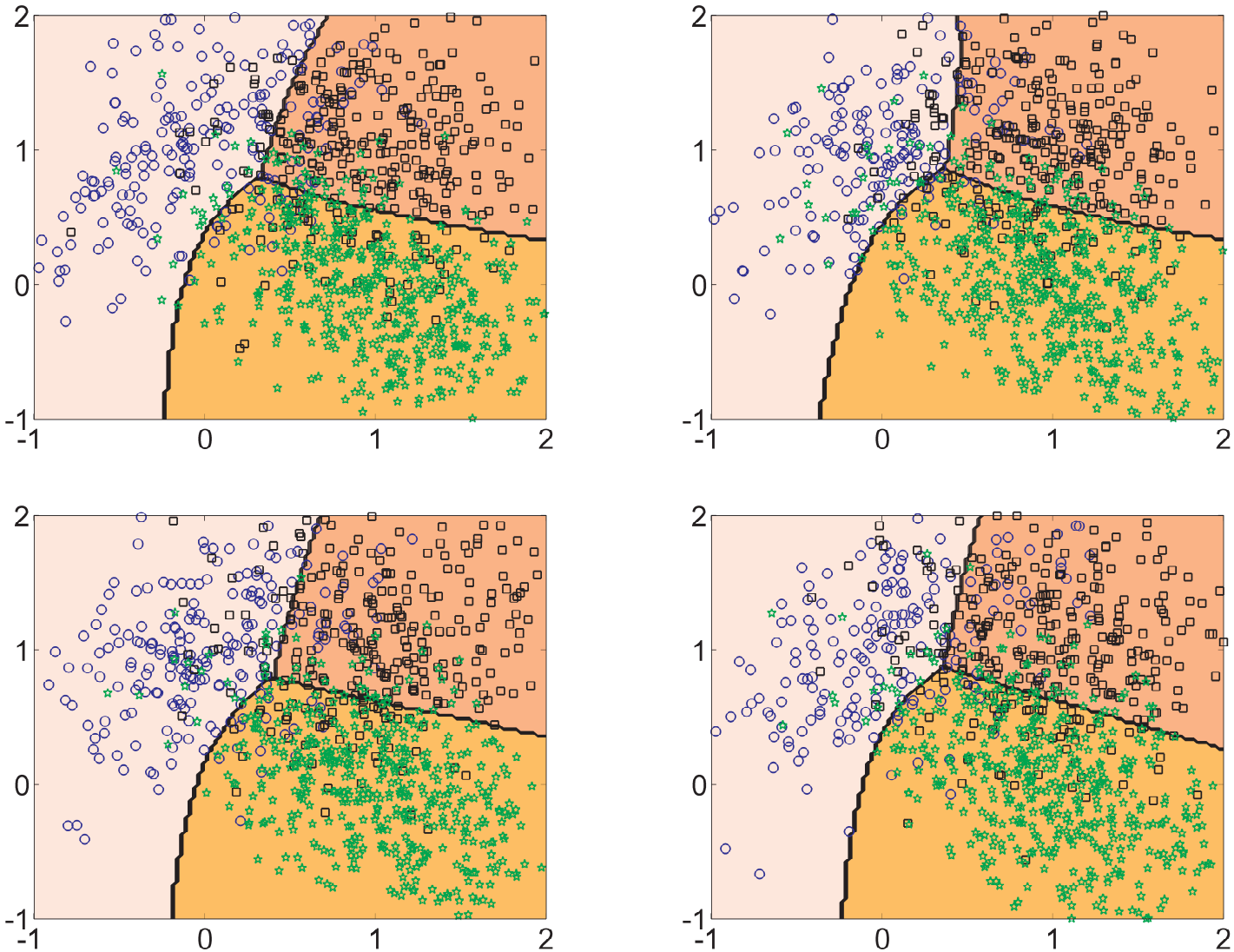
and sample covariance matrices

$$\widehat{\boldsymbol{\Sigma}_k} = \frac{1}{n_k} \sum_{i=1}^{n} \mathbb{1}_{Y_i = k} (\mathbf{X}_i - \widehat{\boldsymbol{\mu}}_k)(\mathbf{X}_i - \widehat{\boldsymbol{\mu}}_k)'$$

in the classification rule

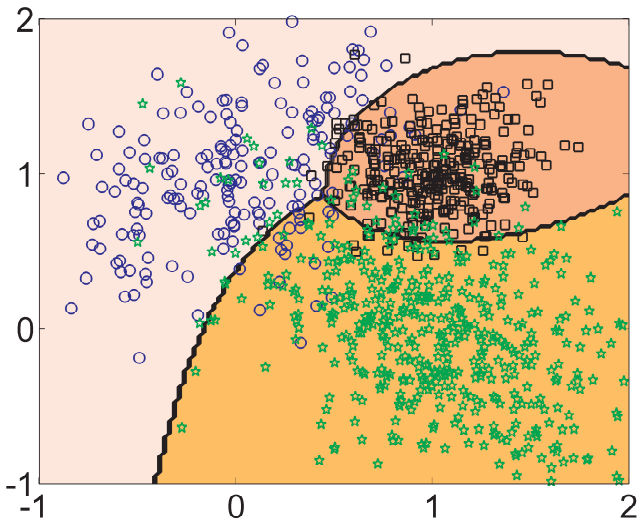
Bayes Classifier

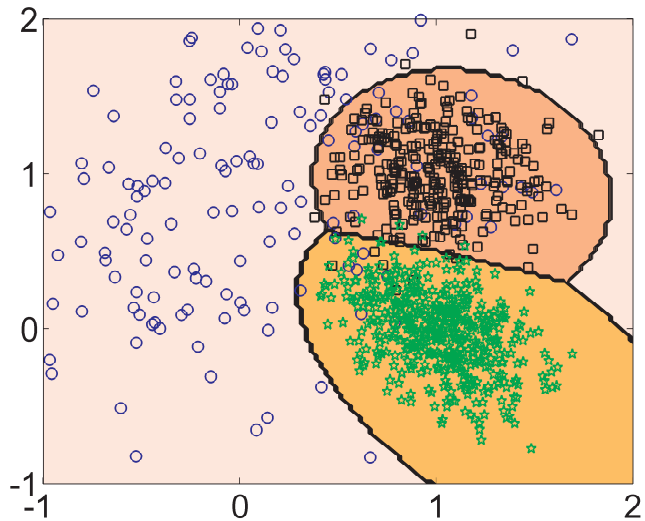
Simulated Classifications...

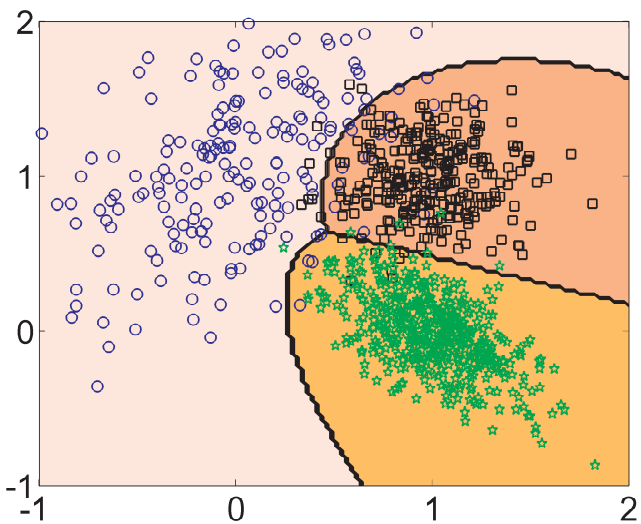


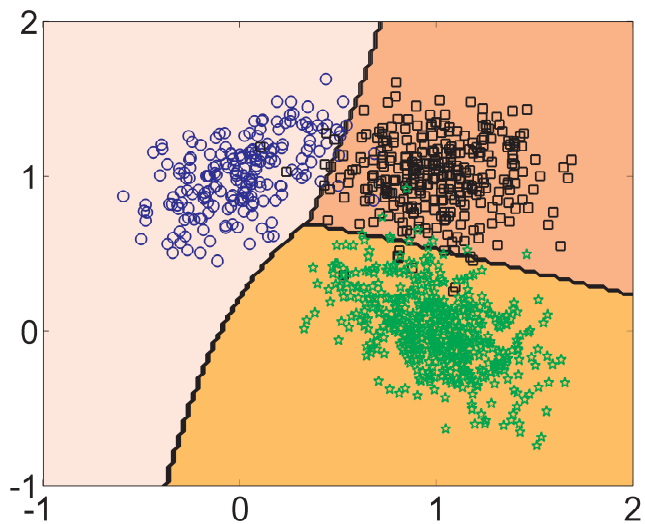
Bayes Classifier

Simulated Classifications...



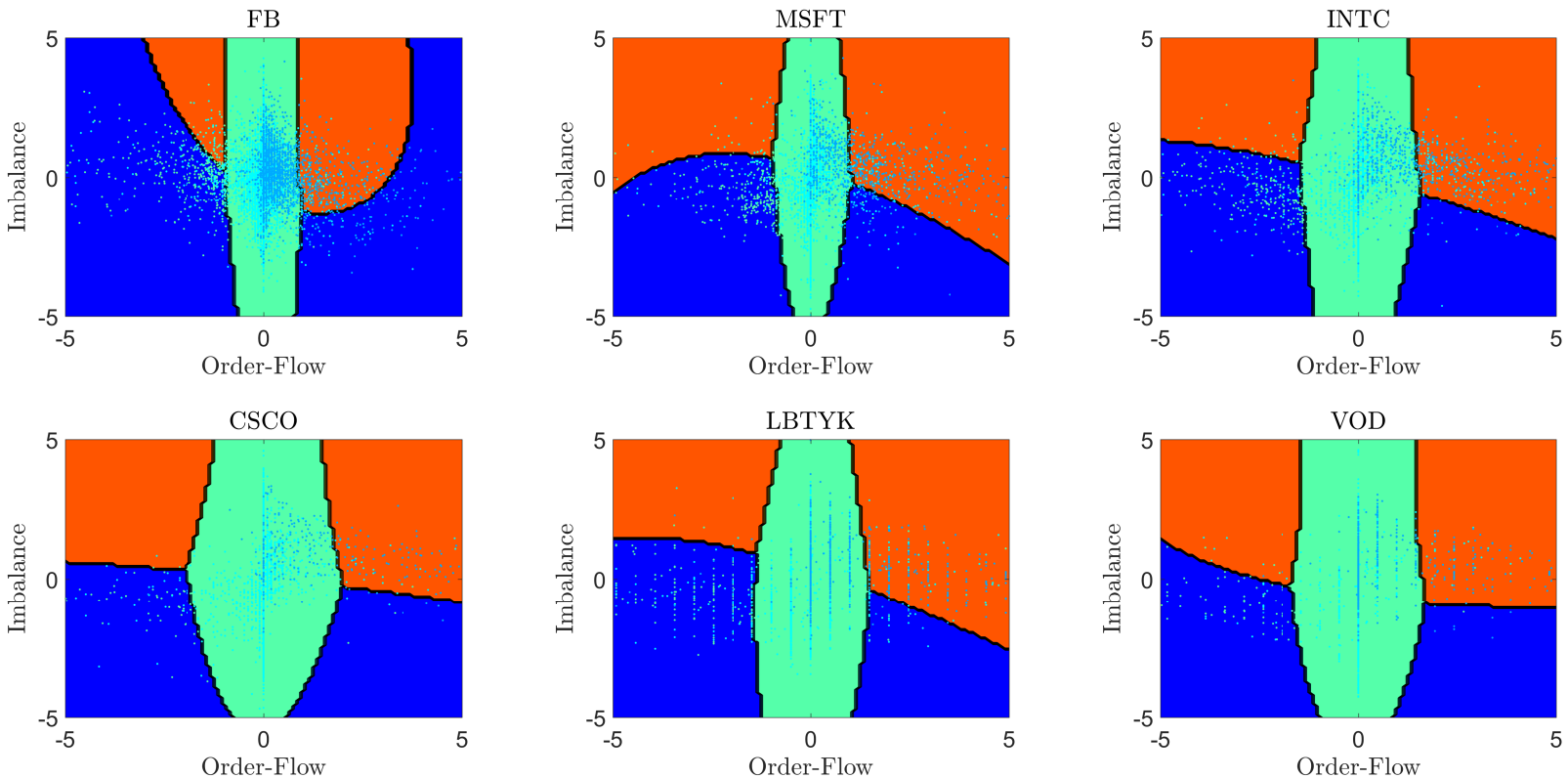






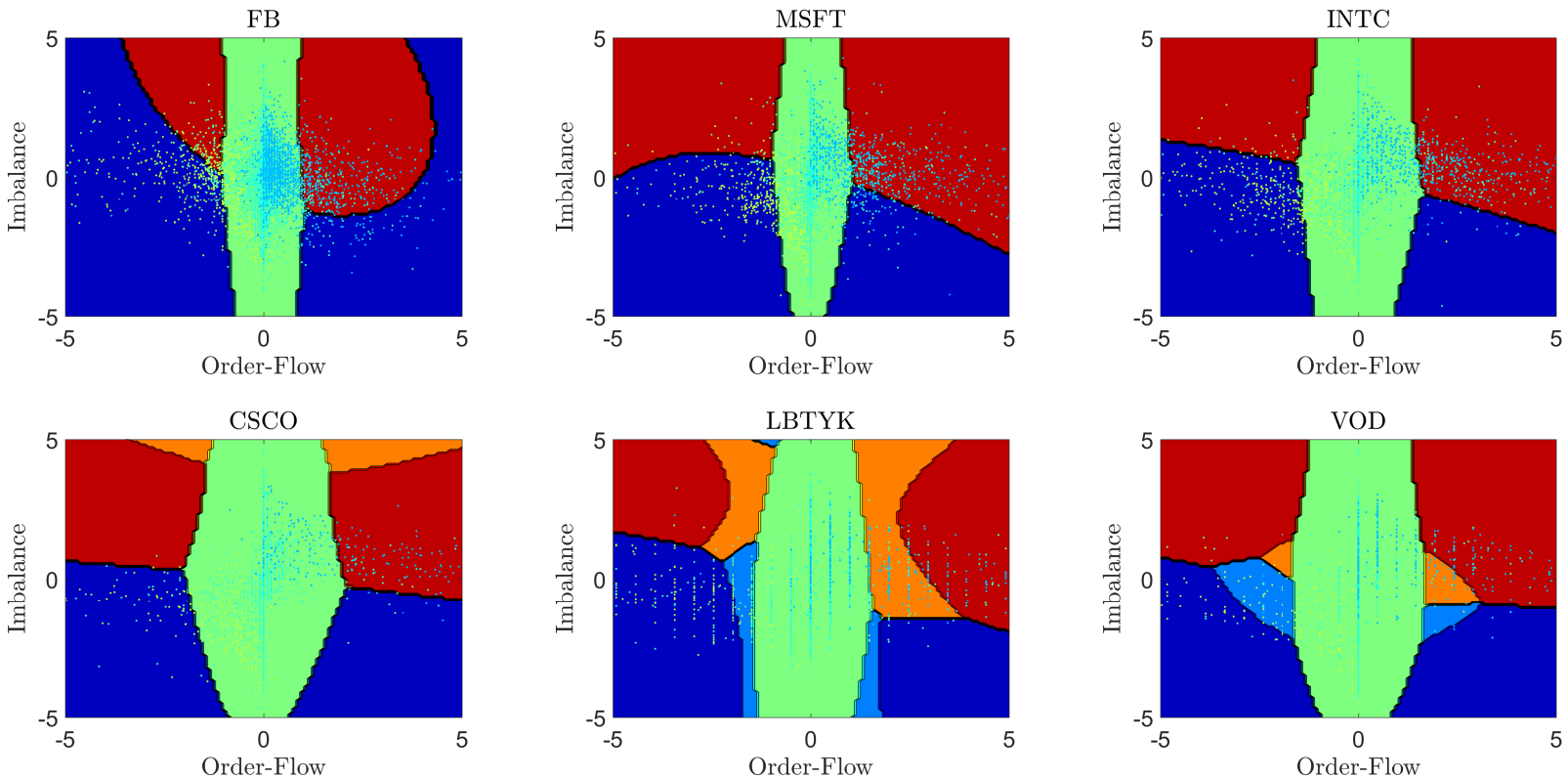
Bayes Classifier

3 classes Price movement : Order Imbalance & Order flow (1s)



Bayes Classifier

5 classes Price movement : Order Imbalance & Order flow (1s)



Bayes Classifier

- If we use an (independent) **kernel density estimator** for the conditional feature probabilities $f_{\mathbf{X}|Y=k}(\boldsymbol{x})$ then we get the **Naïve Bayes Classifier**
- That is, we estimate

$$\widehat{f}_{\mathbf{X}|Y=k}(\boldsymbol{x}) = \widehat{f}_{k,1}(\boldsymbol{x}_1) \times \widehat{f}_{k,2}(\boldsymbol{x}_1) \times \cdots \times \widehat{f}_{k,d}(\boldsymbol{x}_d)\,,$$

with marginal densities

$$\widehat{f}_{k,i}(x) = \frac{1}{n_k} \sum_{i=1}^{N} \mathbb{1}_{Y_i=k}\, \phi_{x_i}(\boldsymbol{x}_i; \varepsilon)\,,$$

and $\phi_{x_i}(\boldsymbol{x}_i; \varepsilon)$ is a kernel density, e.g., gaussian with mean x_i and variance ε^2.

Multi-Class Logistic Regression

Multi-Class Logistic Regression

- **Multi-Class Logistic Regression** is sometimes also called a **maximum entropy classifier**
- It is a **discriminative model** and assumes that

$$\mathbb{P}(Y = c \mid \mathbf{X} = \boldsymbol{x}) := \mu_c(\boldsymbol{x}) = \frac{e^{\boldsymbol{w}_c' \boldsymbol{x}}}{\sum_{c=1}^{C} e^{\boldsymbol{w}_c' \boldsymbol{x}}}$$

  one often sets $\boldsymbol{w}_C = \mathbf{0}$ for identifiability

- This is a "generalized" logistic model

$$\mathbb{P}(Y = 1 \mid \mathbf{X} = \boldsymbol{x}) = \left(1 + e^{-\boldsymbol{w}'\boldsymbol{x}}\right)^{-1}$$

Multi-Class Logistic Regression

- We use the the maximum a posteriori **classifier**

$$\begin{aligned} h(\boldsymbol{x}) &= \arg\max_c \ \mathbb{P}(Y = c \mid \mathbf{X} = \boldsymbol{x}) \\ &= \arg\max_c \ \boldsymbol{w}_c' \boldsymbol{x} \end{aligned}$$

  which leads to **linear decision boundaries**

- How to **estimate the model parameters** $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{C-1}$?... **maximum likelihood**

Multi-Class Logistic Regression

- The **likelihood** function $\mathcal{L}(\mathbf{\Theta} \,|\, \mathbf{X}_1)$ is the probability of the observed outcomes $\mathbf{X}$ with model parameters $\mathbf{\Theta}$, i.e,

$$\mathcal{L}(\mathbf{\Theta} \,|\, \mathbf{X}) = \mathbb{P}(\mathbf{X} \,|\, \mathbf{\Theta})$$

- In the case of multi-class logistic model, we have

$$\mathcal{L}(\mathbf{W} \,|\, ((Y_1, \mathbf{X}_1), \ldots, (Y_N, \mathbf{X}_N)) = \prod_{n=1}^{N} \mathbb{P}(Y = Y_n \,|\, \mathbf{X} = \mathbf{X}_n)$$

and the **log-likelihood** function $\ell(\mathbf{W}) = \mathcal{L}(\mathbf{\Theta} \,|\, \mathbf{X})$ is

$$\boxed{\ell(\mathbf{W}) = \sum_{n=1}^{N} \left(\boldsymbol{y}_n \mathbf{W} \boldsymbol{x}_n - \log \sum_{c=1}^{C} e^{\boldsymbol{w}_c' \boldsymbol{x}_n} \right)}$$

where

$$\mathbf{W} = (\boldsymbol{w}_1 \; \boldsymbol{w}_2 \; \ldots \; \boldsymbol{w}_C)' \;, \qquad \text{and} \qquad [\boldsymbol{y}_n]_c = \mathbb{1}_{Y_n = c}$$

Multi-Class Logistic Regression

- To maximize $\ell(\mathbf{W})$, we must use numerical methods
- Wish to find $\widehat{\mathbf{W}}$ such that

$$\boldsymbol{\mathcal{S}}(\widehat{\mathbf{W}}) = \mathbf{0}\,, \quad \text{where the score function} \quad \mathcal{S}_{ci}(\mathbf{W}) := \partial_{w_{ci}}\ell(\mathbf{W})$$

- **Newton-Raphson** method:
  - Let $\widehat{\mathbf{W}}^{(k)}$ be the current estimate
  - Then, set

$$\widehat{\mathbf{W}}^{(k+1)} = \widehat{\mathbf{W}}^{(k)} + \mathcal{I}(\widehat{\mathbf{W}}^{(k)})^{-1}\,\boldsymbol{\mathcal{S}}(\widehat{\mathbf{W}}^{(k)})$$

    where $\mathcal{I}(\mathbf{W})$ is the **Fisher-Information matrix**

$$[\mathcal{I}(\mathbf{W})]_{c,i,c',i'} = -\frac{\partial^2}{\partial_{w_{ci}w_{c'i'}}}\ell(\mathbf{W})$$

  - Repeat until converged

Multi-Class Logistic Regression

- For multi-class logistic we have
  - The score function

$$\mathcal{S}(\mathbf{W}) = \sum_{n=1}^{N} (\boldsymbol{y}_n - \boldsymbol{\mu}_n).\boldsymbol{x}_n$$

  - The Fisher-Information matrix

$$\mathcal{I}(\mathbf{W}) = \sum_{n=1}^{N} (\texttt{diag}(\boldsymbol{\mu}_n) - \boldsymbol{\mu}_n \boldsymbol{\mu}_n') \otimes (\boldsymbol{x}_n \boldsymbol{x}_n')$$

Multi-Class Logistic Regression

3 classes Price movement : Order Imbalance & Order flow