Pricing d'options d'achat *Call*

TX : Survey on Machine Learning in Finance

Ambroise THIBAULT

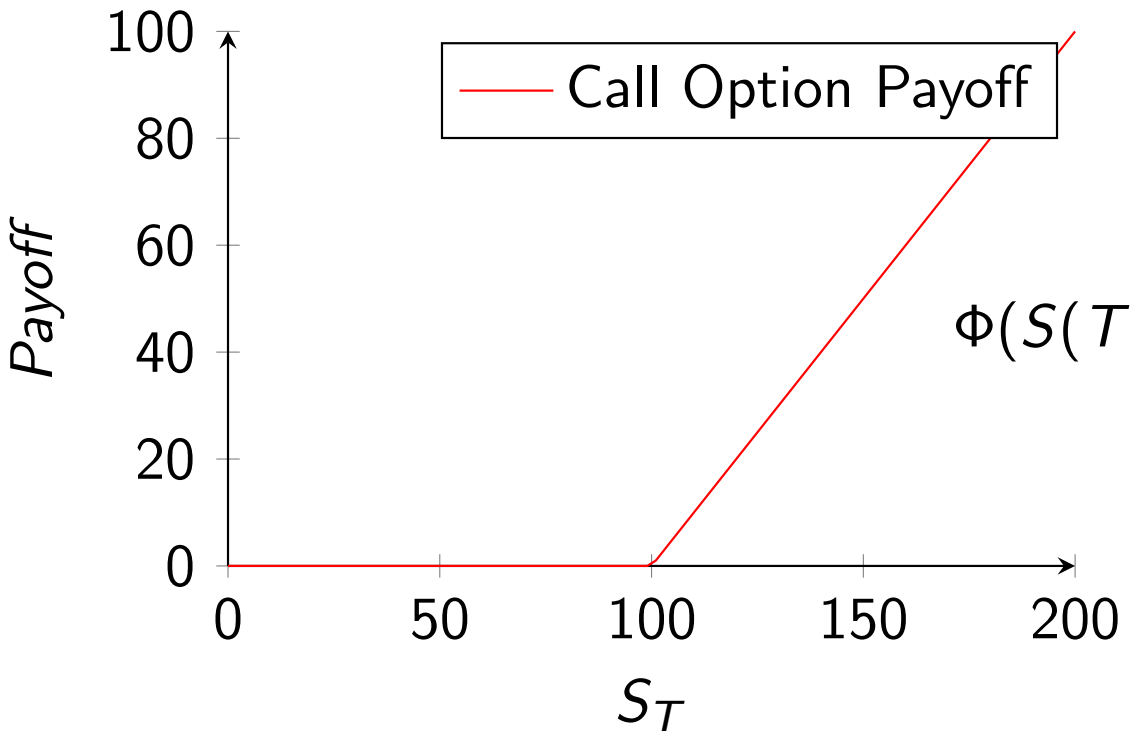
Jeudi 24 avril 2024

Sommaire

Sommaire

Produits dérivés, payoff et prix

Un produit dérivé, ou contigent claim, $\mathcal{X}$ est un actif financier dont la valeur dépend de la valeur d'un autre actif financier (sous-jacent).
On notera $\mathcal{X} : \Phi(S(T))$
De plus on note le prix du produit dérivé à l'instant t comme : $\Pi(t; \mathcal{X})$



$$\Phi(S(T)) = \begin{cases} S_T - K & \text{si} \quad S_T > K \\ 0 & \text{sinon} \end{cases}$$

Modèle de Black-Scholes-Merton

Le modèle de Black-Scholes-Merton (BSM) est composé de 2 actifs ayant les dynamiques suivantes :

$$dB_t = rB_t dt$$

$$dS_t = \mu S_t dt + \sigma S_t dW_t^{\mathbb{P}}$$

Avec :

- $\mu \in \mathbb{R}$ l'espérance de rentabilité de l'action
- $\sigma \in \mathbb{R}$ la volatilité de l'action
- $W^{\mathbb{P}}$ un mouvement brownien sous la probabilité $\mathbb{P}$

On considérera donc deux probabilité :

- $\mathbb{P}$ la probabilité réelle
- $\mathbb{Q}$ la probabilité risque neutre

Formule de B-S pour un option call

On considère une option call de strike K et de maturité T.
On a le payoff suivant :

$$\Phi(S(T)) = \begin{cases} S_T - K & \text{si} \quad S_T > K \\ 0 & \text{sinon} \end{cases}$$

En appliqant la formule d'évaluation risque neutre à ce payoff, on obtient :

$$F_c(t, S_0) = S_0 \mathcal{N}(d_1) - K e^{-r(T-t)} \mathcal{N}(d_2)$$

Avec :

$$d_1 = \frac{1}{\sigma\sqrt{T-t}} \left[\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t) \right]$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

Objectifs

- Utiliser le machine learning pour déterminer le prix d'options
- Utiliser comme base de comparaison la formule de Black-Scholes

Variables nécessaires pour la formule de Black-Scholes :

- S_0 : prix de l'action à l'instant $t = 0$
- K : prix d'exercice de l'option
- T : maturité de l'option
- r : taux d'intérêt sans risque
- σ : volatilité de l'action

Il faut également attribuer à chaque individu son label : prix de l'options à l'instant $t = 0$.

Sommaire

Les 3 sources de données

Une base de donnée Dolthub contenant des données de marché d'options

- Bid et Ask
- Strike K
- date d'expiration et de création
- Type d'option (Call ou Put)
- Grecks (Delta, Gamma, Vega, Theta, Rho)
- Volatilité
- Symbol de l'action sous jacente

Une base de donnée Dolthub contenant des données de marché d'actions

- Prix de l'action
- Date
- Symbol de l'action

Le site du trésoir américain pour les taux d'intérêt sans risque r

Construction du jeu de données

Finalement après pre-processing, on obtient un jeu de données contenant sous la forme d'un fichier csv avec

- 945 505 individus
- L'ensemble des variables nécessaires pour la formule de Black-Scholes
- Label (prix de l'option) est la moyenne du bid (demande) et de l'ask (offre)

Méthodes utilisées pour déterminer le prix d'une option

- Régression par processus gaussien : chapitre 3 *Bayesian Regression and Gaussian Processes* de *Machine Learning: A Probabilistic Perspective*
- Réseaux de neurones : *Option pricing using Machine Learning* de Codruț-Florin Ivașcu - *Option Pricing with Deep Learning* de Anlexander Ke et Andrew Yang
- Autres méthodes de machine learning : Support Vector Regression, Random Forest, XGBoost, LightGBM *Option pricing using Machine Learning* de Codruț-Florin Ivașcu

Sommaire

Régression par processus gaussien

Modèle de régression non paramétrique.
Modèle qui cherche à approximer une fonction f à partir d'un ensemble de données $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ en supposant que les valeurs de f sont des variables aléatoires gaussiennes.
Adapté pour les problèmes avec du bruit et de l'incertitude.
Procesuss Gaussien : Famille de variable aléatoire tel que est une variable aléatoire gaussienne avec loi jointe gaussienne.
Utilisation d'un noyaux : une fonction qui mesure la similarité entre deux points. Fonction de covariance Les paramètres de la fonction de covariance sont appris à partir des données en maximisant la vraisemblance des données d'entrainemnts Construction de matrice de covariance entre les points d'entrainement Construction de la matrice de covariance entre les points d'entrainement et de test

Formalisation du modèle

On un ensemble de données d'entrainement X, y avec X une matrice $n \times d$ et y un vecteur $n \times 1$.
Une fonction vient d'un processus gaussien si pour tout ensemble de points $X = \{x_1, \ldots, x_n\}$, le vecteur $(f(x_1), \ldots, f(x_n)) \sim \mathcal{N}(\mu, K_{X,X})$
On choisit une fonction de covariance $k(x_i, x_j)$
On estime les paramètres de la fonction de covariance en maximisant la vraisemblance des données d'entrainement

Formalisation du modèle

On est en mesure de calculer une matrice de covariance K entre les points d'entrainement et une matrice de covariance K_* entre les points d'entrainement et de test.
On cherche maintenant à déterminer $y|x \sim \mathcal{N}(f(x), \sigma^2)$ La distribution prédictive conditionnellement à un point de test x_* est donnée par :

$$f_*|x_*, X, y \sim \mathcal{N}(\mathbb{E}[f_*|x_*, X, y], var(f_*|x_*, X, y))$$

Avec :

$$\mathbb{E}[f_*|x_*, X, y] = K_*(K + \sigma^2 I)^{-1} y$$

$$var(f_*|x_*, X, y) = k(x_*, x_*) - K_*(K + \sigma^2 I)^{-1} K_*^T$$

MAPE : Mean Absolute Percentage Error $= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \times 100$

MAE : Mean Absolute Error $= \frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|$

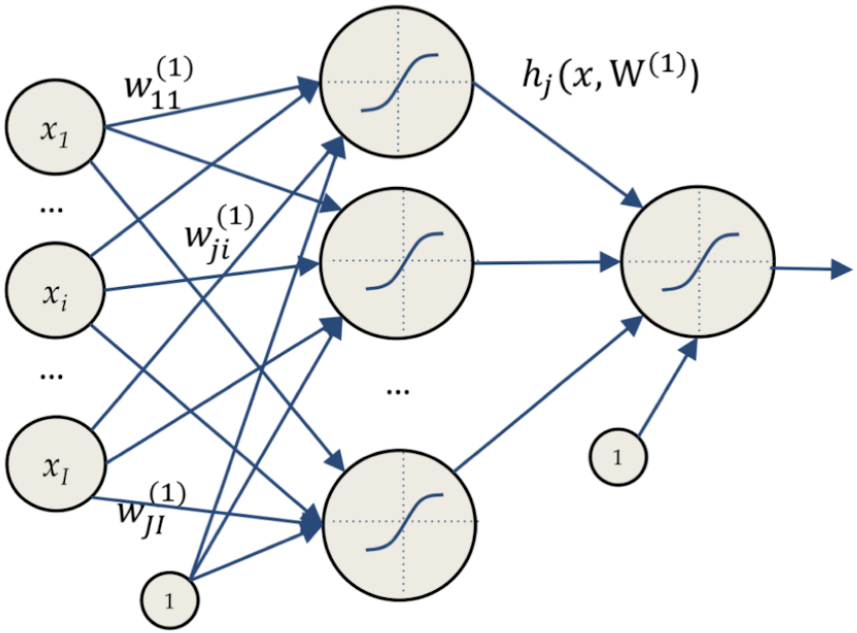
MSE : Mean Squared Error $= \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$

R2 : Coefficient de détermination $= 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$

Selection des données	Nombre de lignes	MAPE	MAE	MSE	R2
Random	100	3.30	4.16	51.2	-0.22
	500	4.20	3.33	62.7	0.0148
	1000	4.00	4.10	262.7	0.0492
	2000	3.99	2.90	113	0.106
	3000	3.45	2.76	115	0.01
By activity	100	0.170	1.03	8.86	0.99
	500	0.13	0.862	5.01	0.996
	1000	0.431	1.01	11.7	0.98
	2000	1.62	1.63	63.6	0.98
	3000	2.06	1.71	63.3	0.98
By index random	100	1.73	1.86	16.7	0.54
	500	2.57	2.19	27.8	0.33
	1000	1.68	4.04	184	-0.003
	2000	1.75	3.08	77.8	0.006
	3000	1.14	3.11	161	0.217
Black&Scholes	100	3.30	4.16	51.2	-0.22
	500	1.56	0.646	1.90	0.97
	1000	0.528	0.737	3.93	0.964
	2000	0.58	0.66	2.80	0.98
	3000	0.53	0.702	3.66	0.97

Sommaire

Fonctionnement d'un réseau de neurones



$$g_1(x_n|w) = g_1\left(\sum_{j=0}^{J} w_{1j}^{(2)} \cdot h_j\left(\sum_{i=0}^{I} w_{ji}^{(1)} \cdot x_{i,n}\right)\right)$$

On cherche à minimiser : $E(w) = \sum_{n=1}^{N}(t_n - g_1(x_n, w))^2$
On utilise principe de back propagation pour minimiser cette fonction
Chain rule pour calculer les dérivées partielles de $E(w)$ par rapport à w puis on applique descente de gradient.

Fonctions d'activations

Fonctions utilisées :

- Sigmoïde : $\sigma(x) = \frac{1}{1+e^{-x}}$
- Tanh : $\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
- ReLU : $f(x) = \max(0, x)$
- Leaky ReLU : $f(x) = \max(0.01x, x)$
- Softmax : $\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$

On utilisera pour les couches cachées Leaky ReLU et pour la couche de sortie ReLU

Résultats

MAPE : Mean Absolute Percentage Error $= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \times 100$

MAE : Mean Absolute Error $= \frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|$

MSE : Mean Squared Error $= \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$

R2 : Coefficient de détermination $= 1-\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$

Selection des données	Nombre de lignes	MAPE	MAE	MSE	R2
	1000	0.170	1.20	3.55	0.99
Random	10000	0.759	051	1.25	0.99
	100000	0.329	0.271	0.421	0.99
	1000	0.170	1.03	8.86	0.99
By activity	10000	0.13	0.862	5.01	0.996
	100000	0.431	0.262	0.370	0.999
	1000	0.416	0.541	1.75	0.98
By index random	10000	0.69	0.539	1.154	0.98
	10000	0.394	0.282	0.486	0.996

Références

John Hull.
Options, Futures, et Autres Actifs Dérivés, 11ème édition.
Pearson, 2018.

Matthew F. Dixon, Igor Halperin, Paul Bilokon.
Machine Learning in finance.
Springer, 2020.

Codruț-Florin Ivașcu.
Option pricing using Machine Learning.

Alexander Ke, Andrew Yang
Option Pricing with Deep Learning.