Introduction to Machine Learning

GBM

Sara El Bouch

February 27, 2025

Rappels

- Données : $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,d}) \in \mathbb{R}^d$
- Valeurs à prédire : $y_i \in \mathbb{R}$ (régression)
- Ensemble d'entraînement : $\mathcal{D}_m = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$

Le modèle linéaire est donné par :

$$y_i = \beta_0 + \sum_{j=1}^{d} \beta_j x_{i,j} + \quad i \in \{1, \ldots, m\} \tag{1}$$

où :

- $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)^\top \in \mathbb{R}^{d+1}$ est le **vecteur des paramètres inconnus** à estimer.

Choix de la fonction coût : moindres carrés

Nous cherchons $\boldsymbol{\beta}$ en minimisant l'erreur quadratique :

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2} \sum_{i=1}^{m} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{i,j} \right)^2 \tag{2}$$

$$= \arg\min_{\boldsymbol{\beta}} \frac{1}{2} RSS(\boldsymbol{\beta}) \tag{3}$$

où $RSS(\boldsymbol{\beta})$ est la somme des erreurs quadratiques résiduelles.

En notation matricielle, on a :

$$RSS(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \tag{4}$$

avec :

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ 1 & x_{2,1} & \dots & x_{2,d} \\ \vdots & \vdots & & \vdots \\ 1 & x_{m,1} & \dots & x_{m,d} \end{pmatrix} \in \mathbb{R}^{m\times(d+1)} \tag{5}$$

Optimisation par annulation du gradient

Nous voulons minimiser :

$$\min_{\boldsymbol{\beta}} RSS(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \tag{6}$$

Théorème (condition d'optimalité) : Si $RSS(\boldsymbol{\beta})$ est convexe, alors le minimum est obtenu lorsque :

$$\nabla RSS(\widehat{\boldsymbol{\beta}}) = \mathbf{0}. \tag{7}$$

Le gradient est donné par :

$$\nabla RSS(\boldsymbol{\beta}) = -\mathbf{X}^\top \mathbf{y} + \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} \tag{8}$$

Solution analytique

L'optimisation donne la solution analytique :

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \tag{9}$$

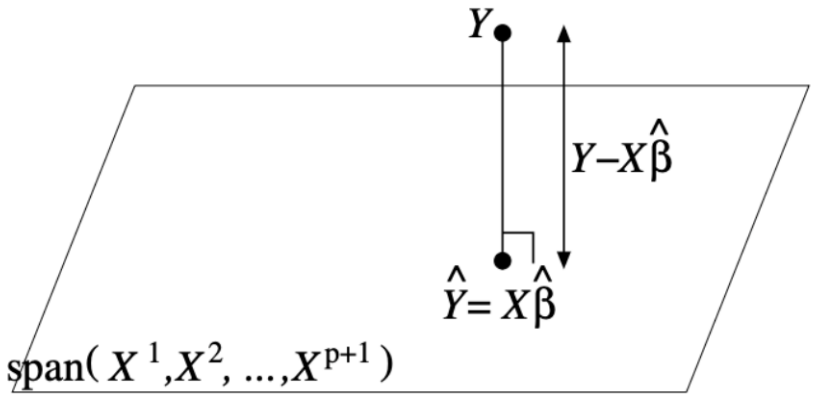
sous la condition que $\mathbf{X}^\top \mathbf{X}$ soit inversible ($m > d + 1$).
Remarque. Si d est très grand, inverser $\mathbf{X}^\top \mathbf{X}$ devient prohibitif. (La descente du gradient, stay tuned!)

Projection orthogonale sur l'espace des prédictions :

$$\mathbf{y} = \mathbf{X}\widehat{\boldsymbol{\beta}} + \boldsymbol{\epsilon}$$

La minimisation revient à projeter $\mathbf{y}$ sur le sous-espace engendré par les colonnes de $\mathbf{X}$:

$$\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = 0$$



Évaluation des performances

Mesures de performance :

- **Erreur quadratique moyenne (MSE) :**

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

- **Coefficient de corrélation r :**

$$r = \frac{\sum_{i=1}^{m}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{m}(y_i - \bar{y})^2 \sum_{i=1}^{m}(\hat{y}_i - \bar{\hat{y}})^2}}$$

Attention : Toujours évaluer sur un **jeu de test** indépendant.

Régularisation : Ridge Regression

Pour éviter le sur-apprentissage, on ajoute une pénalisation λ :

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y} \tag{10}$$

Effet de la régularisation :

- Réduit la variance du modèle.
- Implique un compromis biais-variance.
- Empêche la singularité de $\mathbf{X}^\top \mathbf{X}$.

Régression Lasso (L_1)

Principe :

- Ajoute une pénalisation L_1 à la régression linéaire :

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{d} |\beta_j|$$

- Contrairement à Ridge, qui réduit les coefficients sans les annuler, **Lasso force certains coefficients à zéro**.

Pourquoi utiliser Lasso ?

- Sélectionne automatiquement les variables importantes.
- Produit des modèles plus interprétables.
- Utile quand beaucoup de variables sont inutiles.

Comparaison Lasso vs Ridge

Effet des régularisations :

Méthode **Sélection de variables**	**Effet sur les coefficients**
Ridge (L_2)	Réduit l'amplitude mais ne les annule pas
Lasso (L_1)	Peut annuler certains coefficients

Qu'est-ce que la régression logistique ?

- La régression linéaire est utilisée pour une variable dépendante **continue**.
- On va s'intéresser à **la classification** au travers de la régression logistique = Le modèle → coût → entrainement.

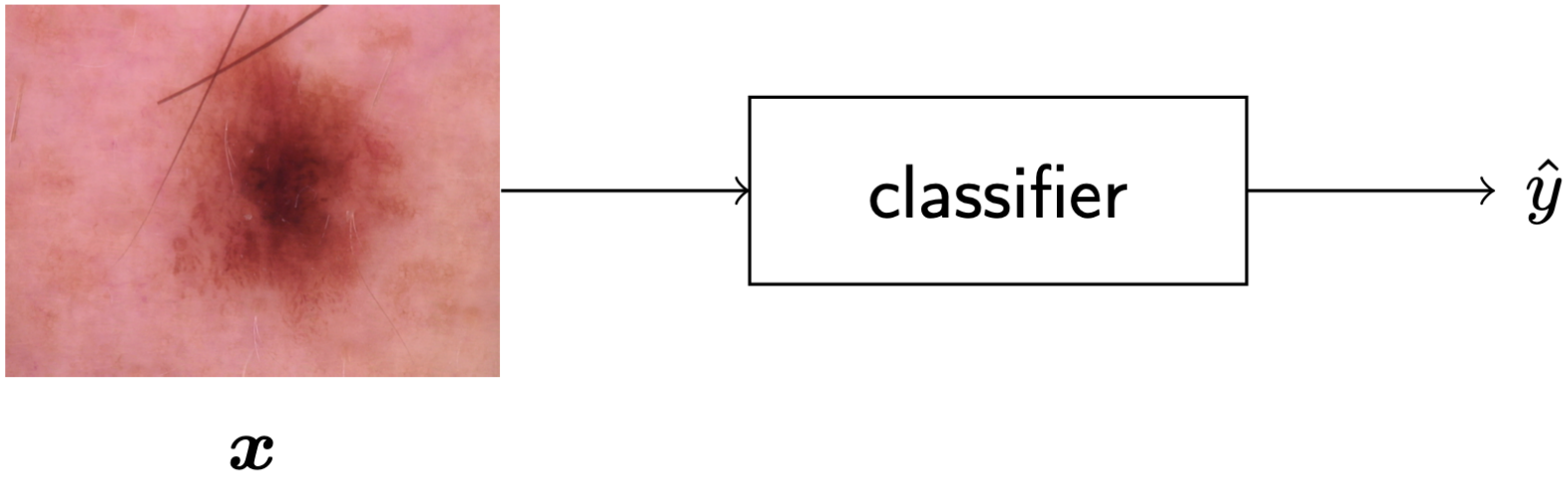


Figure: y and $\widehat{y} \in \{benign, malignant\}$

Classification simple

- On suppose que x est un nombre et $y \in \{0, 1\}$
- On suppose que les différentes valeurs de x sont linéairement séparables.

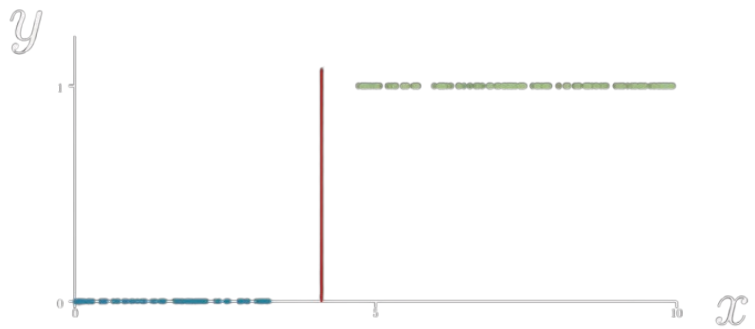
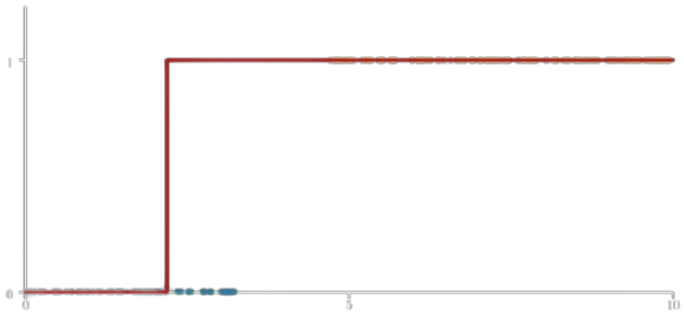


Figure: On choisit un seuil (ligne rouge) qu'on note s

$$\widehat{y_i} = \begin{cases} 0 \text{ si } s \times x \leq 0 \\ 1 \text{ sinon} \end{cases} \tag{11}$$

De manière générale

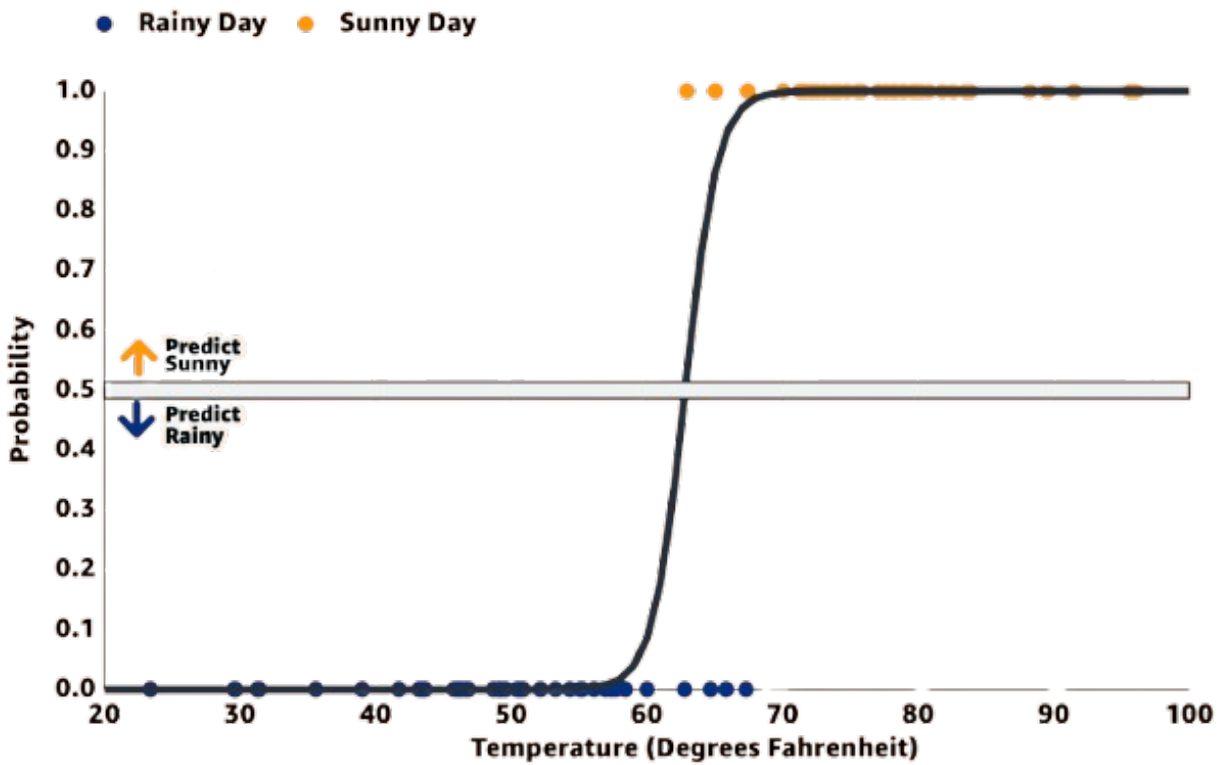
$$\widehat{y_i} = \begin{cases} 0 \text{ si } a \times x + b \leq 0 \\ 1 \text{ sinon} \end{cases} \tag{12}$$



- On veut interpréter $\widehat{y}_i$ comme **une probabilité** On introduit la fonction **sigmoide**:

$$\sigma(x) = \frac{1}{1 + \exp^{-x}} \tag{13}$$

$$\widehat{y}_i = p(y = 1|x) = \sigma(ax + b) \in [0, 1]. \tag{14}$$



Fonction coût: L'entropie croisée

On définit le coût de l'entropie pour une donnée i comme:

$$J^i = -\log p_\beta(y = y_i|x_i) \tag{15}$$

- Pour une classification **binaire** on a 2 cas, $y_i = 0$ ou $y_i = 1$

$$J^i = \begin{cases} -\log p_\beta(y = 1|x_i), & y_i = 1 \\ -\log p_\beta(y = 0|x_i), & y_i = 0 \end{cases} \tag{16}$$

En notant $\widehat{y}_i = p(y = 1|x)$ (c'est un choix arbitraire, on peut choisir $\widehat{y}_i = p(y = 0|x)$, on peut réecrire J^i comme:

$$J^i = -y_i \log(\widehat{y}_i) - (1 - y_i) \log(1 - \widehat{y}_i) \tag{17}$$

Comment optimiser cette fonction coût ?

On peut utiliser **l'algorithme de descente du gradient** pour trouver les paramètres a et b optimaux.

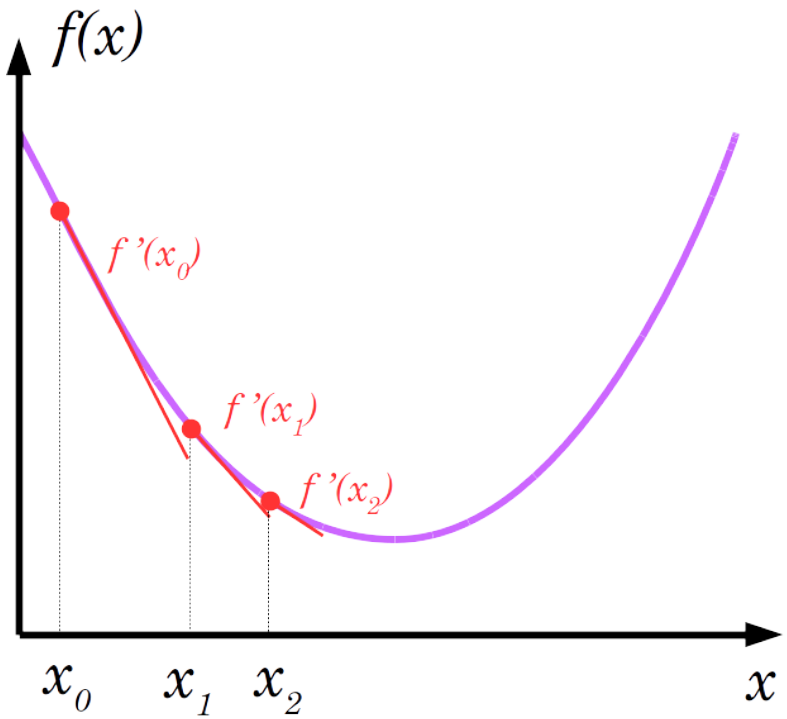


Figure: Descente du gradient

	Prédit Positif	**Prédit Négatif**
Réel Positif	VP (Vrais Positifs)	FN (Faux Négatifs)
Réel Négatif	FP (Faux Positifs)	VN (Vrais Négatifs)

Table: Matrice de confusion

$$\text{Précision} = \frac{VP}{VP + FP} \tag{18}$$

Interprétation : Proportion des prédictions positives qui sont réellement positives.

$$\text{Rappel} = \frac{VP}{VP + FN} \tag{19}$$

Interprétation : Proportion des cas positifs correctement identifiés.

Formule :

$$\text{Spécificité} = \frac{VN}{VN + FP} \tag{20}$$

Interprétation : Proportion des cas négatifs bien classés.

Formule :

$$F1 = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \tag{21}$$

Interprétation : Mesure équilibrée entre précision et rappel.

Quelques remarques

- Ce modèle peut être étendu au cas multi-classe, où $y \in \{1, \ldots, K\}$, ce qui donne lieu à la régression logistique multinomiale (on utilise plus la fonction sigmoide mais plutôt la fonction **softmax**).
- Bien que ce modèle soit utilisé pour la classification, on l'appelle un "régresseur" car la fonction $\sigma(ax + b)$ produit des valeurs réelles comprises entre 0 et 1. C'est l'interprétation de ces valeurs (par exemple, en appliquant un seuil à 0.5) qui permet de prendre des décisions de classification.
- On peut également utiliser la régularisation comme dans la régression linéaire pour éviter le sur-apprentissage.