A2DI: Régression logistique

John Klein

Université de Lille - CRIStAL UMR CNRS 9189

Où en est-on dans notre problème d'apprentissage supervisé ?

- En théorie on veut minimiser Err_{esp}.
- En pratique on minimisera Err_{train} tout en s'assurant que Err_{train} ne dévie pas de Err_{esp}.
- Les solutions minimisant Err_{train} au sens de la théorie de la décision pour des fonctions de perte classiques tournent autour de solutions probabilistes reposant sur la connaissance de la distribution $p_{Y|X}$
- Si, on cherche directement $p_{Y|X}$, on parle de modèle discriminatif.

Où en est-on dans notre problème d'apprentissage supervisé ?

- En théorie on veut minimiser Err_{esp}.
- En pratique on minimisera Err_{train} tout en s'assurant que Err_{train} ne dévie pas de Err_{esp}.
- Les solutions minimisant Err_{train} au sens de la théorie de la décision pour des fonctions de perte classiques tournent autour de solutions probabilistes reposant sur la connaissance de la distribution $p_{Y|X}$
- Si, on cherche directement $p_{Y|X}$, on parle de modèle discriminatif.

La régression logistique offre une solution de cette nature pour un problème de classification.

Plan du chapitre

Reprenons un exemple habituel :

- Soit un problème de classification avec $\mathbb{Y} = \{0; 1\}$ où 0 représente l'hypothèse qu'un crédit n'est pas accordé et 1 qu'un crédit est accordé.

Reprenons un exemple habituel :

- Soit un problème de classification avec $\mathbb{Y} = \{0; 1\}$ où 0 représente l'hypothèse qu'un crédit n'est pas accordé et 1 qu'un crédit est accordé.

- Soit $\mathbb{X}$ l'espace des attributs brutes où chaque dimension x_j est un nombre réel représentant un information du style : *salaire*, *découvert max*, *endettement*, etc.

Reprenons un exemple habituel :

- Soit un problème de classification avec $\mathbb{Y} = \{0; 1\}$ où 0 représente l'hypothèse qu'un crédit n'est pas accordé et 1 qu'un crédit est accordé.

- Soit $\mathbb{X}$ l'espace des attributs brutes où chaque dimension x_j est un nombre réel représentant un information du style : *salaire*, *découvert max*, *endettement*, etc.

- On se doute qu'un score du style :

$$\text{score} = w_1 \times \textit{salaire} + w_2 \times \textit{decouvert} + w_3 \times \textit{endettement} + b$$

est pertinent pour décider de l'attribution du crédit..

Reprenons un exemple habituel :

- Soit un problème de classification avec $\mathbb{Y} = \{0; 1\}$ où 0 représente l'hypothèse qu'un crédit n'est pas accordé et 1 qu'un crédit est accordé.

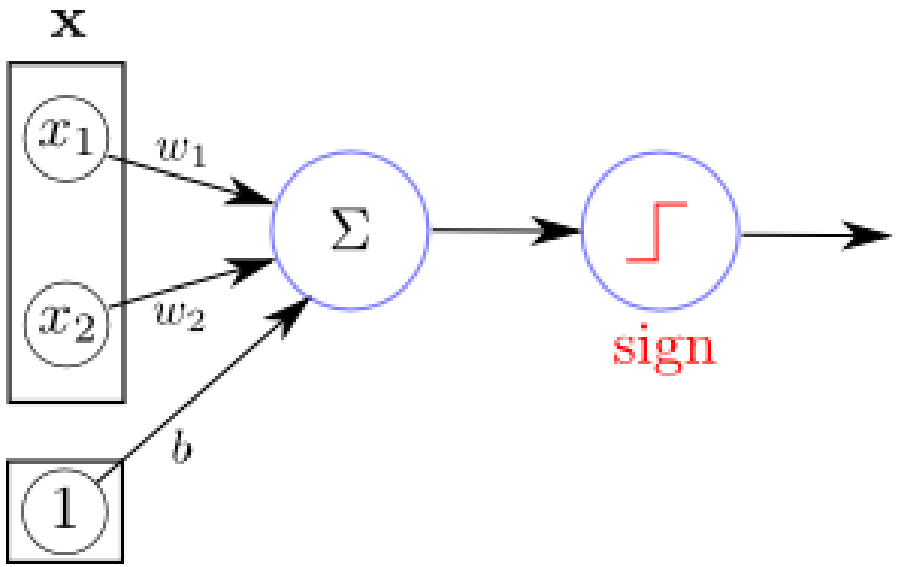
- Soit $\mathbb{X}$ l'espace des attributs brutes où chaque dimension x_j est un nombre réel représentant un information du style : *salaire*, *découvert max*, *endettement*, etc.

- On se doute qu'un score du style :

$$\text{score} = w_1 \times \textit{salaire} + w_2 \times \textit{decouvert} + w_3 \times \textit{endettement} + b$$

est pertinent pour décider de l'attribution du crédit.. à condition que les paramètres w_1, w_2, w_3, et b soient bien réglés !

- Partir sur cette idée revient à choisir un modèle linéaire.
- .. mais, le score n'est ni une classe, ni une probabilité d'appartenance à une classe.
- Avec le perceptron, on avait décidé d'appliquer la fonction sign au score :

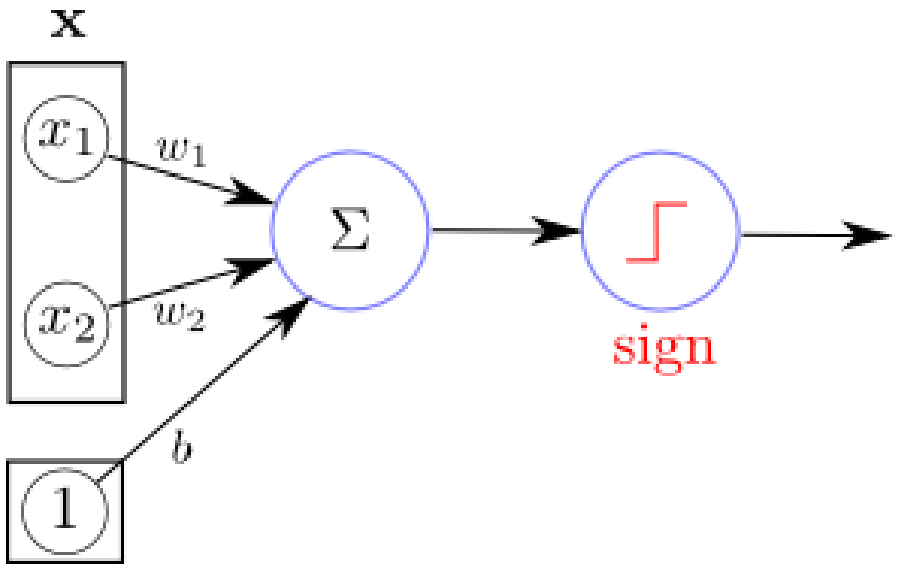


$$f_\theta(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

prédiction pour l'exemple $\mathbf{x}$

Le perceptron :

- fournit directement une classe sans passer par la case proba,
- repose sur l'hypothèse de séparabilité linéaire qui n'est pas réaliste en pratique.



$$f_\theta(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

prédiction pour l'exemple $\mathbf{x}$

Adoptons une solution plus réaliste :

- y n'est pas directement déductible de $\text{score}(\mathbf{x})$.
- On autorise donc des réponses bruitées et on cherche $p_{Y|\text{score}(\mathbf{x})}$.
- Comme $Y|\mathbf{X}=\mathbf{x}$ est binaire, on sait que $Y|\mathbf{X}=\mathbf{x} \sim \text{Ber}(\mu(\mathbf{x}))$ avec $\mu \in [0;1]$.
- Pourquoi ne pas faire en sorte que le score soit la probabilité d'appartenance à la classe 1 ?

$$\text{score}(\mathbf{x}) = \mu(\mathbf{x}). \tag{1}$$

- Il suffit de faire en sorte que $\text{score}(\mathbf{x}) \in [0;1]$!

- Une fonction qui permet d'obtenir cela est la fonction sigmoïde.

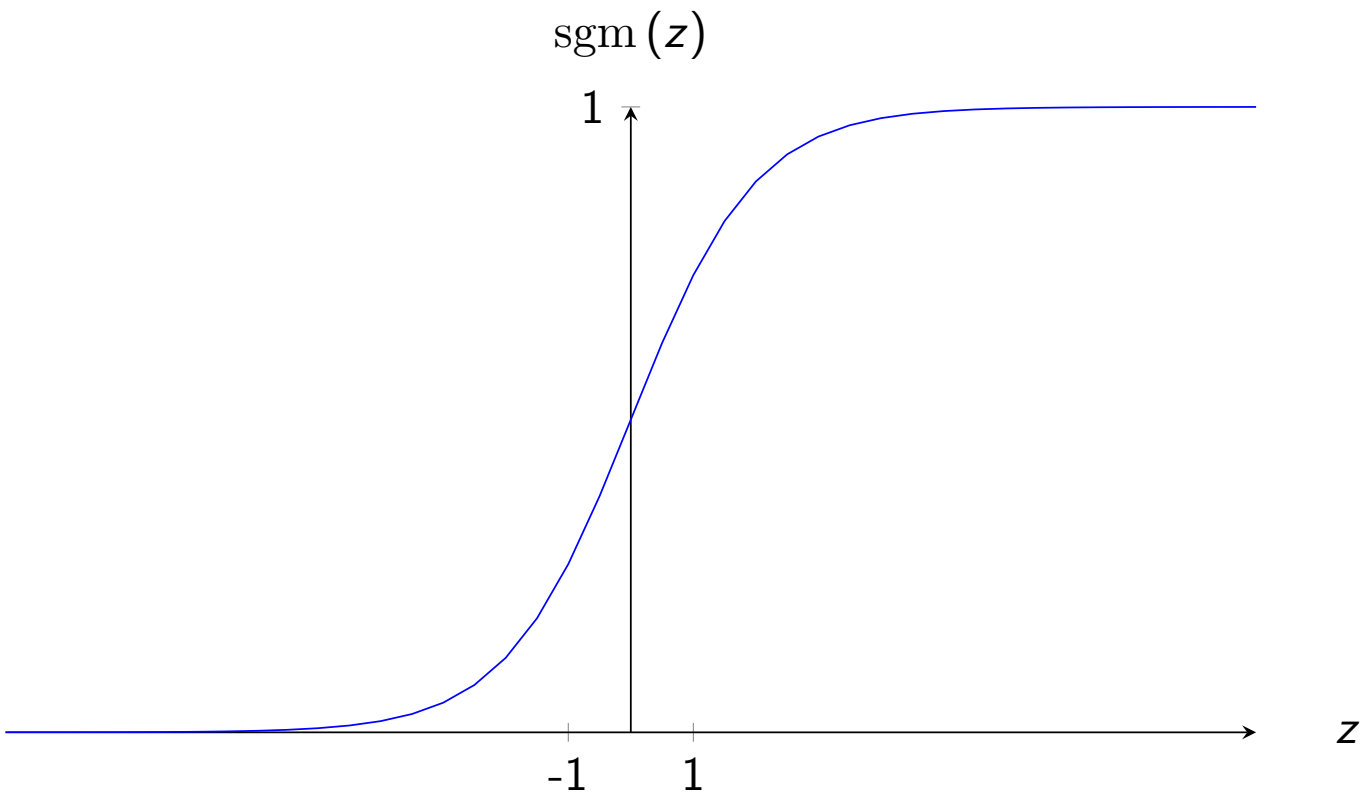
- Une fonction qui permet d'obtenir cela est la fonction sigmoïde.
- La fonction sigmoïde sgm est une fonction continue de $\mathbb{R}$ dans $[0;1]$, telle que

$$\mathrm{sgm}(z) = \frac{1}{1+e^{-z}}. \tag{2}$$

- Une fonction qui permet d'obtenir cela est la fonction sigmoïde.
- La fonction sigmoïde sgm est une fonction continue de $\mathbb{R}$ dans $[0;1]$, telle que

$$\mathrm{sgm}(z) = \frac{1}{1+e^{-z}}. \tag{2}$$

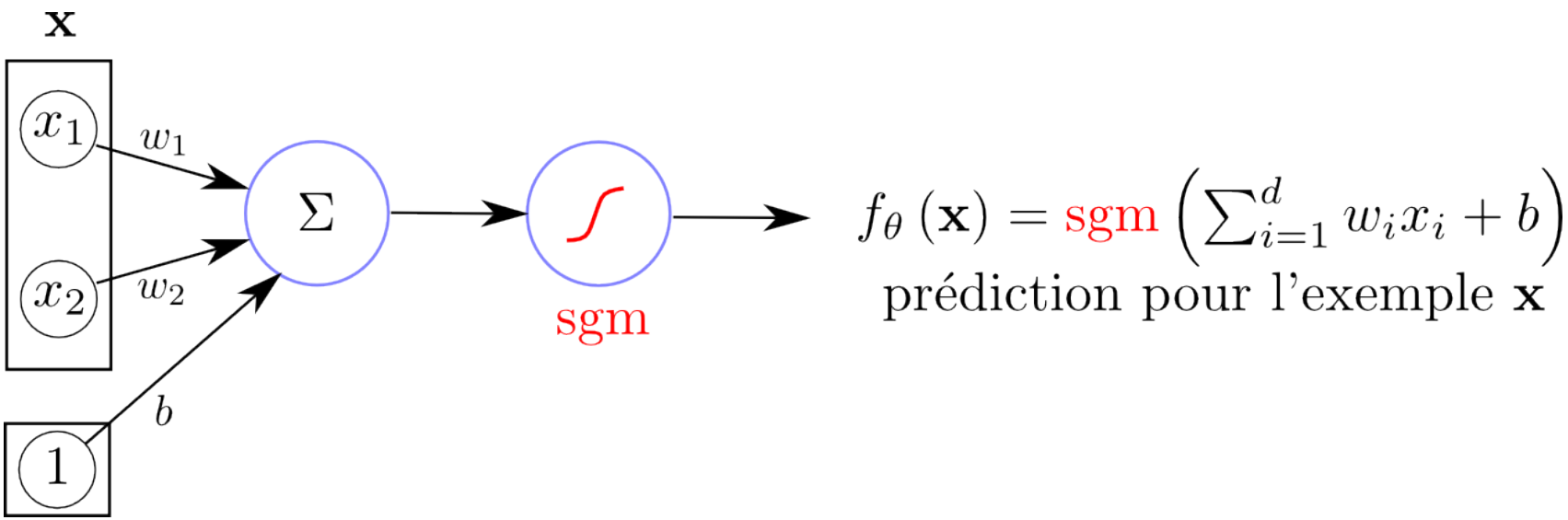
- Voici le graphe de la fonction sigmoïde :



Le modèle complet de la régression logistique est :

$$Y|\mathbf{X} = \mathbf{x} \sim \text{Ber}\left(\text{sgm}\left(\text{score}\left(\mathbf{x}\right)\right)\right), \tag{3}$$

$$\Leftrightarrow Y|\mathbf{X} = \mathbf{x} \sim \text{Ber}\left(\text{sgm}\left(\mathbf{w}^T \cdot \mathbf{x} + b\right)\right) \tag{4}$$



Soit θ le vecteur qui contient tous les paramètres du modèle :

$$\boldsymbol{\theta} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix} \tag{5}$$

Soit θ le vecteur qui contient tous les paramètres du modèle :

$$\boldsymbol{\theta} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix} \tag{5}$$

Selon le modèle de la régression logistique, quelle valeur de θ explique le mieux l'observation d'un couple $\left(\mathbf{x}^{(i)}, c^{(i)}\right)$?

Soit θ le vecteur qui contient tous les paramètres du modèle :

$$\boldsymbol{\theta} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix} \tag{5}$$

Selon le modèle de la régression logistique, quelle valeur de θ explique le mieux l'observation d'un couple $\left(\mathbf{x}^{(i)}, c^{(i)}\right)$?

$\longrightarrow$ Celle qui maximise

$$p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\boldsymbol{\theta}}\left(c^{(i)}\right) = \begin{cases} \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x} + b\right) & \text{si } c^{(i)} = 1 \\ 1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x} + b\right) & \text{si } c^{(i)} = 0 \end{cases}. \tag{6}$$

Reformulons sous une forme plus compacte :

$$p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\boldsymbol{\theta}}\left(c^{(i)}\right) = \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)^{c^{(i)}} \times \left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right)^{1-c^{(i)}}. \tag{7}$$

Reformulons sous une forme plus compacte :

$$p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\theta}\left(c^{(i)}\right) = \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)^{c^{(i)}} \times \left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right)^{1-c^{(i)}}. \tag{7}$$

Selon le modèle de la régression logistique, quelle valeur de θ explique le mieux tout mon ensemble d'apprentissage ?

Reformulons sous une forme plus compacte :

$$p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\theta}\left(c^{(i)}\right) = \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)^{c^{(i)}} \times \left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right)^{1-c^{(i)}}. \tag{7}$$

Selon le modèle de la régression logistique, quelle valeur de θ explique le mieux tout mon ensemble d'apprentissage ?
$\longrightarrow$ Celle qui maximise la vraisemblance :

Reformulons sous une forme plus compacte :

$$p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\theta}\left(c^{(i)}\right) = \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)^{c^{(i)}} \times \left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right)^{1-c^{(i)}}. \tag{7}$$

Selon le modèle de la régression logistique, quelle valeur de θ explique le mieux tout mon ensemble d'apprentissage ?
$\longrightarrow$ Celle qui maximise la vraisemblance :

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)^{c^{(i)}} \times \left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right)^{1-c^{(i)}} \tag{8}$$

Passons à la $\mathrm{NLL}(\boldsymbol{\theta}) =$

$$-\sum_{i=1}^{n} c^{(i)} \log \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right) + \left(1 - c^{(i)}\right) \log\left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right). \tag{9}$$

Passons à la $\mathrm{NLL}(\boldsymbol{\theta}) =$

$$-\sum_{i=1}^{n} c^{(i)} \log \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right) + \left(1 - c^{(i)}\right) \log\left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right). \tag{9}$$

Cette équation peut s'interprêter comme l'entropie croisée moyenne entre :

- la distribution empirique des classes pour le seul échantillon $c^{(i)}$: $\hat{p}^{(i)}(c) = \mathbb{I}_{c^{(i)}}(c)$ et

Passons à la $\mathrm{NLL}(\boldsymbol{\theta}) =$

$$-\sum_{i=1}^{n} c^{(i)} \log \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right) + \left(1 - c^{(i)}\right) \log\left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right). \tag{9}$$

Cette équation peut s'interprêter comme l'entropie croisée moyenne entre :

- la distribution empirique des classes pour le seul échantillon $c^{(i)}$: $\hat{p}^{(i)}(c) = \mathbb{I}_{c^{(i)}}(c)$ et
- la distribution prédite par notre modèle $p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\boldsymbol{\theta}}(c) = \mathrm{Ber}\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right)\right)$.

Passons à la $\mathrm{NLL}(\boldsymbol{\theta}) =$

$$-\sum_{i=1}^{n} c^{(i)} \log \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right) + \left(1 - c^{(i)}\right) \log\left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right). \tag{9}$$

Cette équation peut s'interprêter comme l'entropie croisée moyenne entre :

- la distribution empirique des classes pour le seul échantillon $c^{(i)}$: $\hat{p}^{(i)}(c) = \mathbb{I}_{c^{(i)}}(c)$ et
- la distribution prédite par notre modèle $p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\boldsymbol{\theta}}(c) = \mathrm{Ber}\left(f_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right)\right)$.

$$\mathrm{NLL}(\boldsymbol{\theta}) = \sum_{i=1}^{n} H\left(\hat{p}^{(i)},\ p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\boldsymbol{\theta}}\right), \tag{10}$$

$$= n \times Err_{train} \tag{11}$$

Passons à la $\mathrm{NLL}(\boldsymbol{\theta}) =$

$$-\sum_{i=1}^{n} c^{(i)} \log \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right) + \left(1 - c^{(i)}\right) \log\left(1 - \mathrm{sgm}\left(\mathbf{w}^T \cdot \mathbf{x}^{(i)} + b\right)\right). \tag{12}$$

Posons :

$$\mathbf{x}_+^{(i)} = \begin{bmatrix} x_1^{(i)} \\ \vdots \\ x_d^{(i)} \\ 1 \end{bmatrix} \tag{13}$$

On a $\mathrm{NLL}(\boldsymbol{\theta})$

$$= -\sum_{i=1}^{n} c^{(i)} \log\left(\frac{1}{1 + e^{-\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}}}\right) + \left(1 - c^{(i)}\right) \log\left(1 - \frac{1}{1 + e^{-\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}}}\right). \tag{14}$$

On continues et $\mathrm{NLL}(\boldsymbol{\theta}) =$

$$\sum_{i=1}^{n} c^{(i)} \log\left(1 + e^{-\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}}\right) - \left(1 - c^{(i)}\right) \log\left(\frac{e^{-\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}}}{1 + e^{-\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}}}\right). \tag{15}$$

Au final,

$$\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \sum_{i=1}^{n} \left(1 - c^{(i)}\right) \left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right) - \log \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right). \tag{16}$$

Au final,

$$\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \sum_{i=1}^{n} \left(1 - c^{(i)}\right) \left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right) - \log \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right). \tag{16}$$

Cette fonction est convexe mais il n'y pas de solution explicite au problème de minimisation :

$$\arg\min_{\boldsymbol{\theta}} \mathrm{NLL}\left(\boldsymbol{\theta}\right).$$

Au final,

$$\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \sum_{i=1}^{n} \left(1 - c^{(i)}\right) \left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right) - \log \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right). \tag{16}$$

Cette fonction est convexe mais il n'y pas de solution explicite au problème de minimisation :

$$\arg\min_{\boldsymbol{\theta}} \mathrm{NLL}\left(\boldsymbol{\theta}\right).$$

On va donc devoir utiliser un algorithme d'optimisation.

Modèles linéaires : bilan

Modèles linéaires : bilan

Régression Linéaire	**Perceptron**	**Régression Logistique**

Modèles linéaires : bilan

	Régression Linéaire	**Perceptron**	**Régression Logistique**
$f_\theta(\mathbf{x})$ Pred.	$\mathbf{w}^T \cdot \mathbf{x} + b$	$\text{sign}\left(\mathbf{w}^T \cdot \mathbf{x} + b\right)$	$\text{sgm}\left(\mathbf{w}^T v\mathbf{x} + b\right)$

Modèles linéaires : bilan

	Régression Linéaire	**Perceptron**	**Régression Logistique**
$f_\theta(\mathbf{x})$ Pred.	$\mathbf{w}^T \cdot \mathbf{x} + b$	$\text{sign}\left(\mathbf{w}^T \cdot \mathbf{x} + b\right)$	$\text{sgm}\left(\mathbf{w}^T v\mathbf{x} + b\right)$
Tâche	Régression	Classification	Classification

Modèles linéaires : bilan

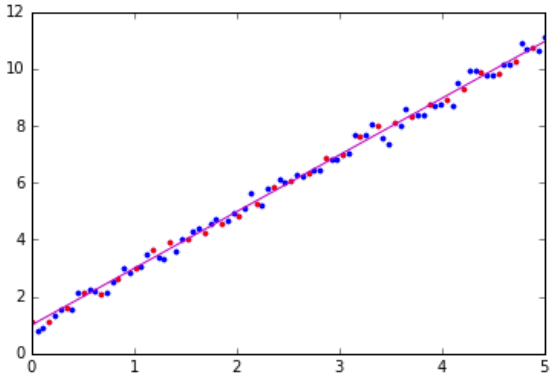
	Régression Linéaire	**Perceptron**	**Régression Logistique**
$f_\theta(\mathbf{x})$ Pred.	$\mathbf{w}^T \cdot \mathbf{x} + b$	$\text{sign}\left(\mathbf{w}^T \cdot \mathbf{x} + b\right)$	$\text{sgm}\left(\mathbf{w}^T v\mathbf{x} + b\right)$
Tâche	Régression	Classification	Classification
Sortie	valeur	classe	proba de classe

Modèles linéaires : bilan

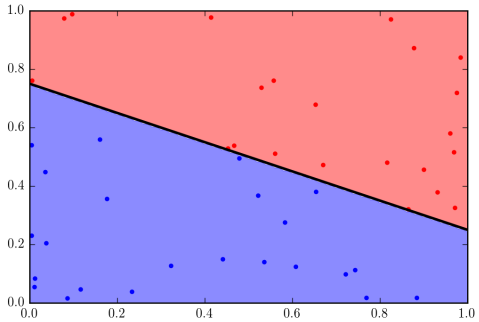
	Régression Linéaire	**Perceptron**	**Régression Logistique**
$f_\theta(\mathbf{x})$ Pred.	$\mathbf{w}^T \cdot \mathbf{x} + b$	$\text{sign}\left(\mathbf{w}^T \cdot \mathbf{x} + b\right)$	$\text{sgm}\left(\mathbf{w}^T v\mathbf{x} + b\right)$
Tâche	Régression	Classification	Classification
Sortie	valeur	classe	proba de classe
Perte	Quadratique	0-1	Entropie croisée

Modèles linéaires : bilan

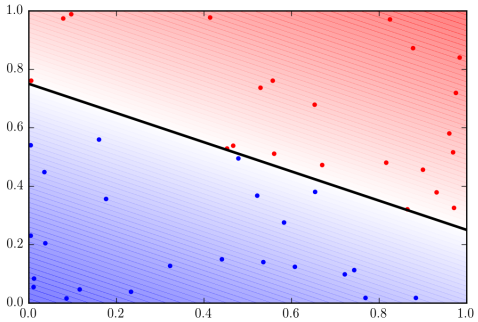
Régression Linéaire



Perceptron



Régression Logistique



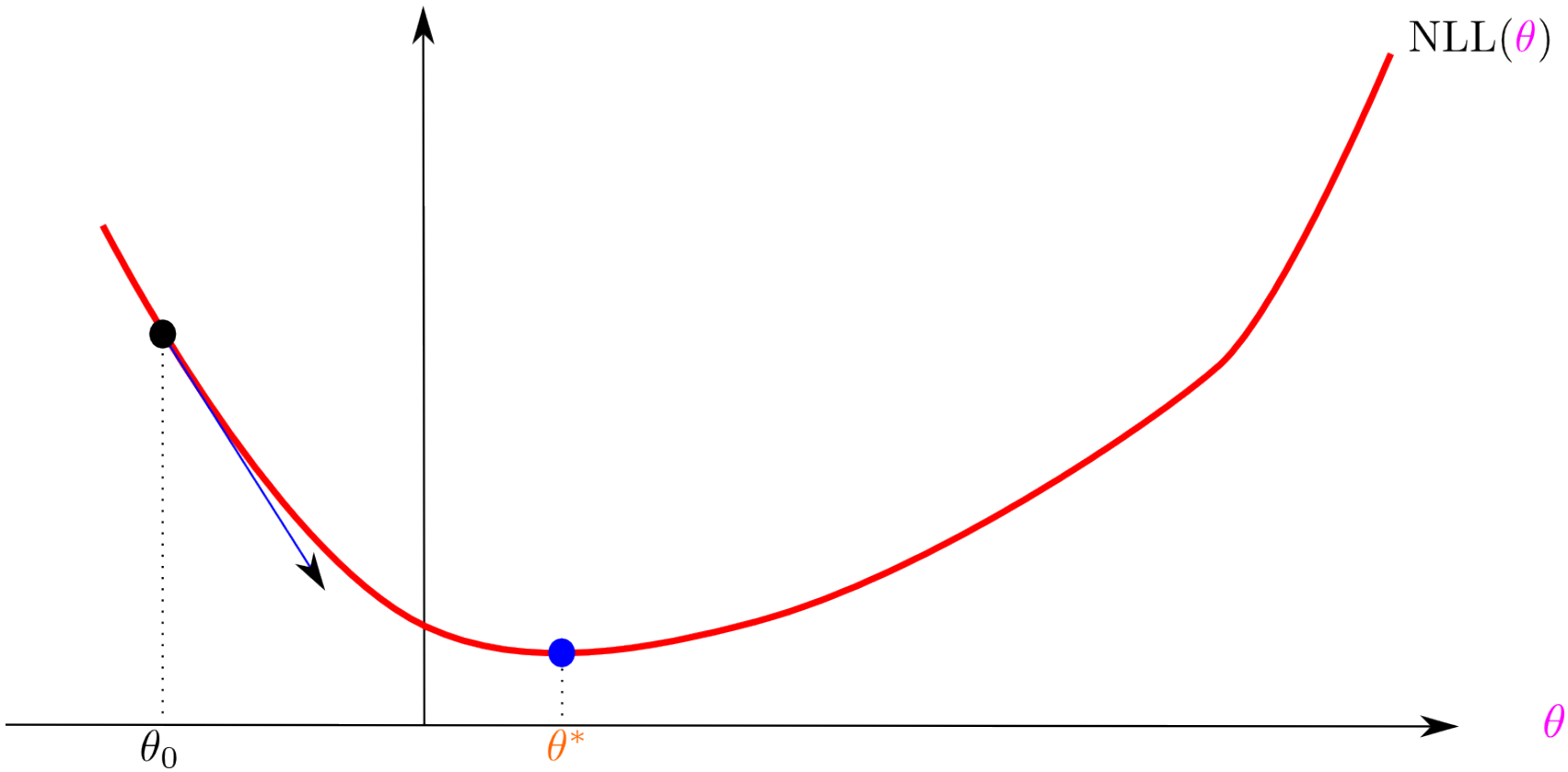
Plan du chapitre

Retour à la régression logistique :
Prochain défi : minimiser la NLL

- Impossible de calculer le MLE analytiquement.

Retour à la régression logistique :
Prochain défi : minimiser la NLL

- Impossible de calculer le MLE analytiquement.
- On va utiliser une descente de gradient.



Minimisation de la NLL par descente de gradient

Pause Optim !

Minimisation de la NLL par descente de gradient

Pause Optim !

- Cet algorithme repose sur une mise à jour itérative de $\boldsymbol{\theta}$ selon :

$$\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \eta \times \frac{d}{d\boldsymbol{\theta}} \mathrm{NLL}\left(\boldsymbol{\theta}\right),$$

Minimisation de la NLL par descente de gradient

Pause Optim !

- Cet algorithme repose sur une mise à jour itérative de $\boldsymbol{\theta}$ selon :

$$\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \eta \times \frac{d}{d\boldsymbol{\theta}} \mathrm{NLL}\left(\boldsymbol{\theta}\right),$$

avec η un paramètre de convergence, appelé *learning rate*.

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

Minimisation de la NLL par descente de gradient : principe graphique

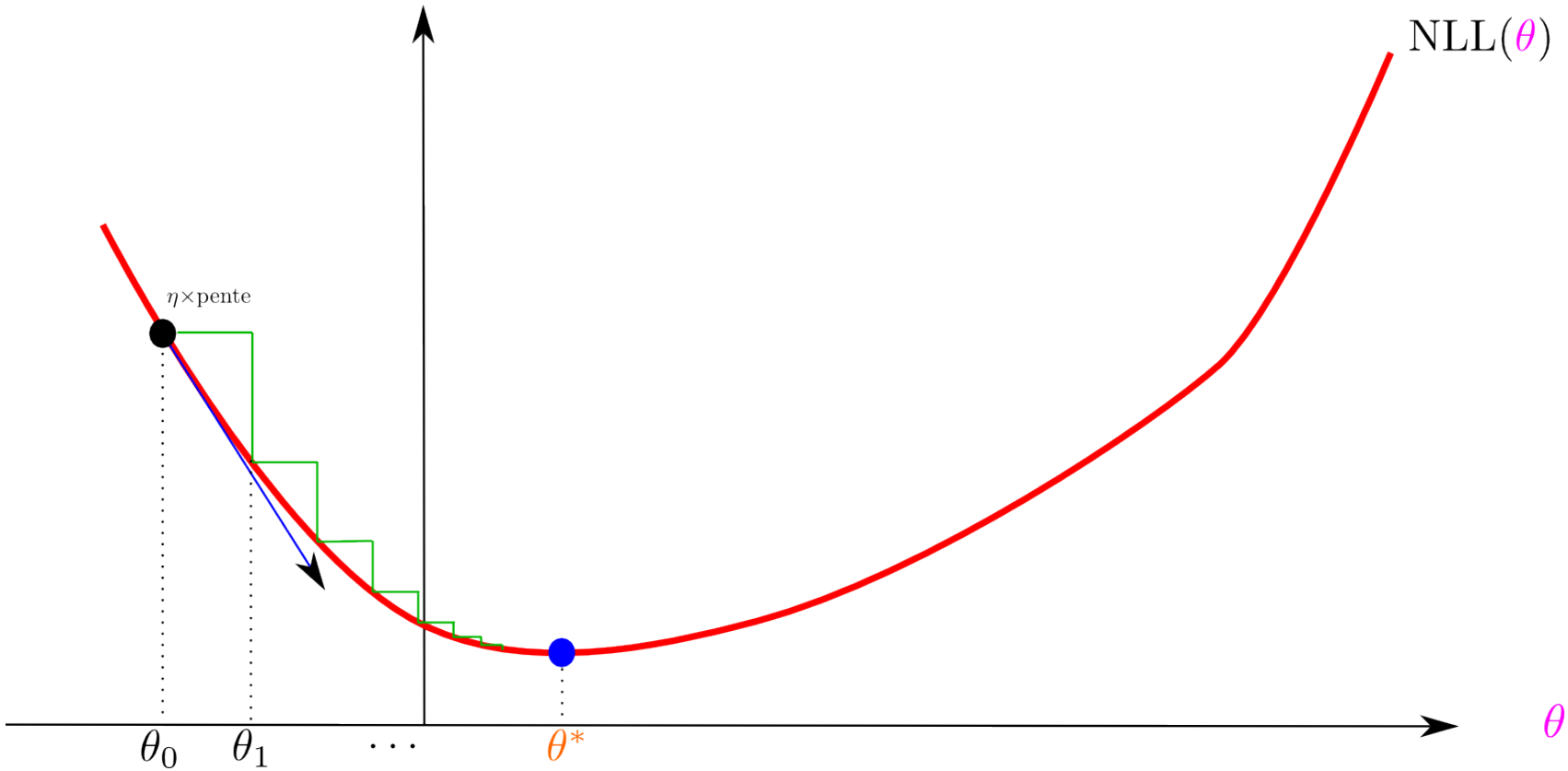
Pause Optim !

- Quand η est bien choisi, l'algorithme fonctionne ainsi :

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- Quand η est bien choisi, l'algorithme fonctionne ainsi :



Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

Minimisation de la NLL par descente de gradient : principe graphique

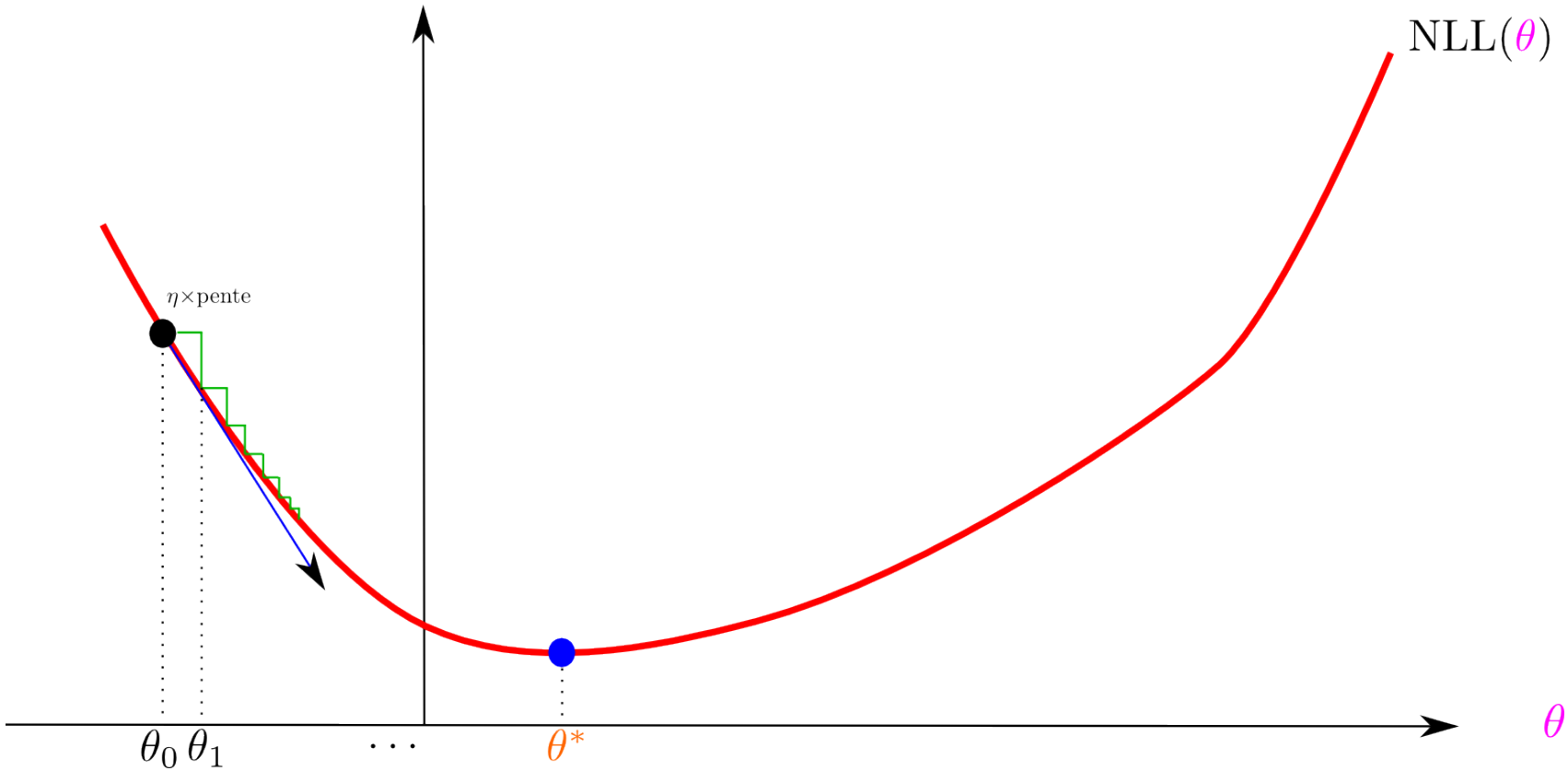
Pause Optim !

- Quand η est choisi trop petit :

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- Quand η est choisi trop petit :



Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

Minimisation de la NLL par descente de gradient : principe graphique

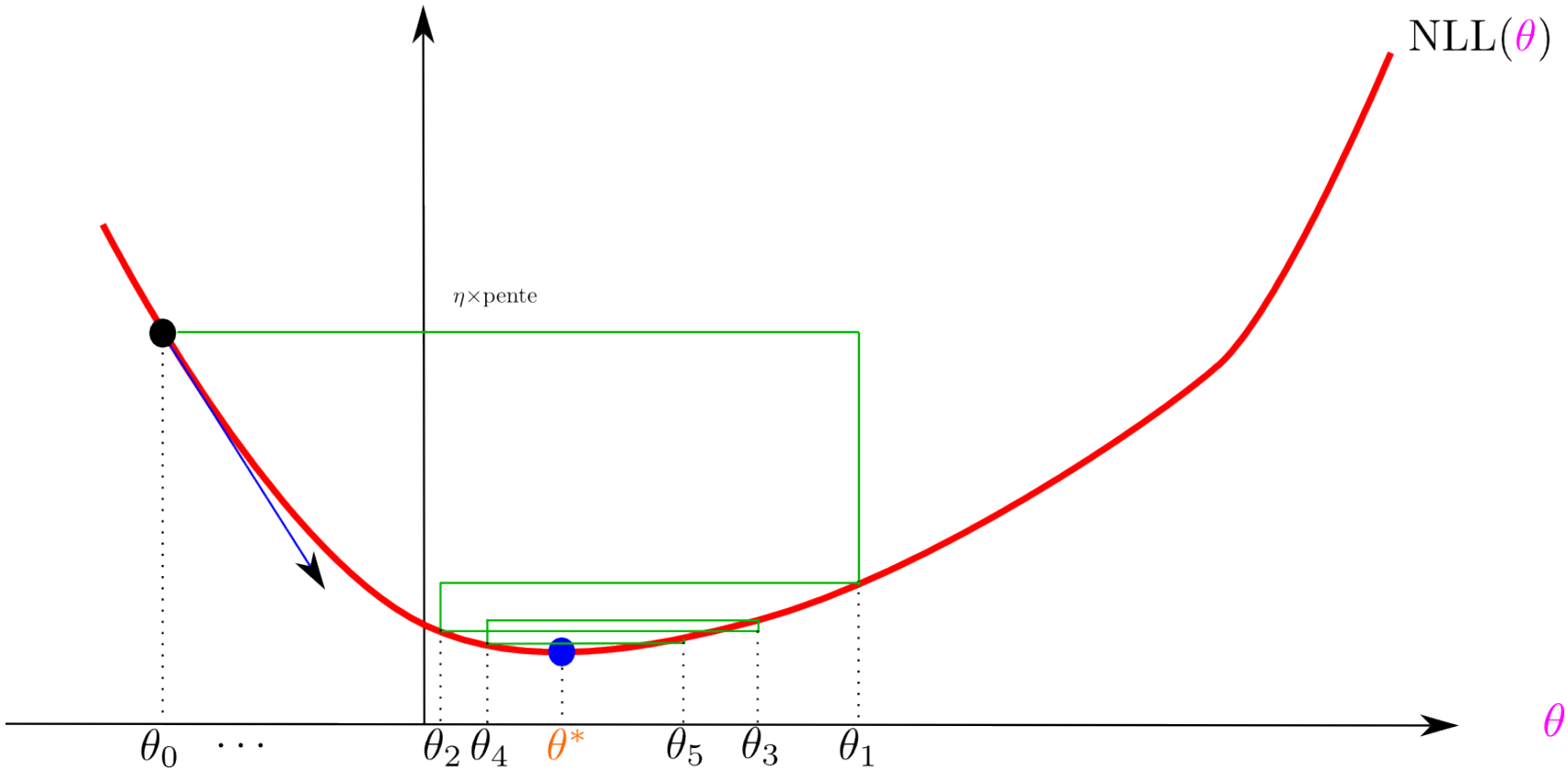
Pause Optim !

- Quand η est choisi trop grand :

Minimisation de la NLL par descente de gradient : principe graphique

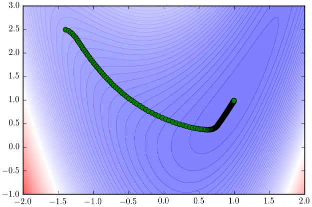
Pause Optim !

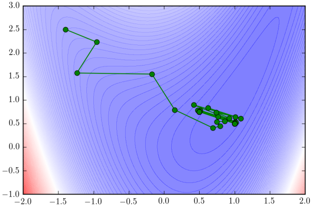
- Quand η est choisi trop grand :

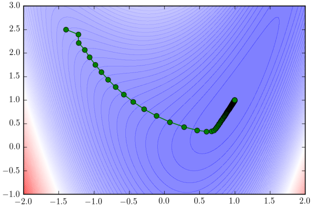


Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim ! Même explication en 2D







Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- N'y a-t-il pas un moyen intelligent de choisir η ?

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

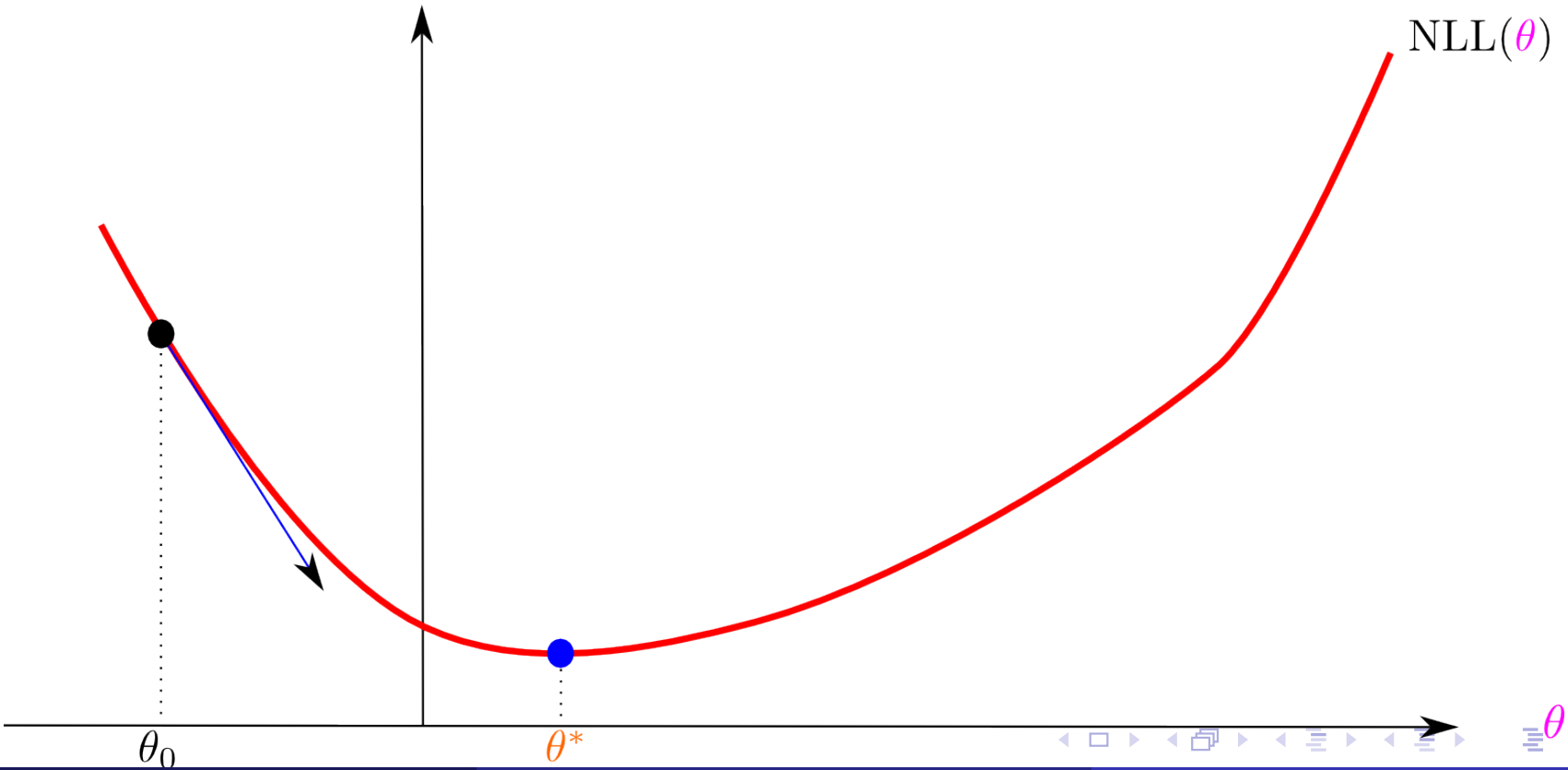
- N'y a-t-il pas un moyen intelligent de choisir η ?
- Oui, en le rendant itératif selon :

$$\eta_t = \frac{\eta}{\frac{d^2}{d\boldsymbol{\theta}^2}\mathrm{NLL}\left(\boldsymbol{\theta}_t\right)} \tag{17}$$

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

$$\eta_t = \eta \times \left(\frac{d^2}{d\boldsymbol{\theta}^2} \mathrm{NLL}\left(\boldsymbol{\theta}_t\right) \right)^{-1} \tag{18}$$



Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- Cette façon de calculer η_t est appelée méthode de Newton.

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- Cette façon de calculer η_t est appelée méthode de Newton.
- Le scalaire η est en général fixé à 1.

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- Cette façon de calculer η_t est appelée méthode de Newton.
- Le scalaire η est en général fixé à 1.
- $\frac{d^2}{d\boldsymbol{\theta}^2}\mathrm{NLL}\left(\boldsymbol{\theta}_t\right)$ est en général une matrice dite Hessienne.

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- Cette façon de calculer η_t est appelée méthode de Newton.
- Le scalaire η est en général fixé à 1.
- $\frac{d^2}{d\boldsymbol{\theta}^2}\mathrm{NLL}\left(\boldsymbol{\theta}_t\right)$ est en général une matrice dite Hessienne.
- Elle peut converger vers un max si la fonction n'est pas convexe.

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- Cette façon de calculer η_t est appelée méthode de Newton.
- Le scalaire η est en général fixé à 1.
- $\frac{d^2}{d\boldsymbol{\theta}^2}\mathrm{NLL}\left(\boldsymbol{\theta}_t\right)$ est en général une matrice dite Hessienne.
- Elle peut converger vers un max si la fonction n'est pas convexe.
- L'inversion et le calcul du Hessien peut être couteux.

Minimisation de la NLL par descente de gradient : principe graphique

Pause Optim !

- Cette façon de calculer η_t est appelée méthode de Newton.
- Le scalaire η est en général fixé à 1.
- $\frac{d^2}{d\boldsymbol{\theta}^2}\mathrm{NLL}\left(\boldsymbol{\theta}_t\right)$ est en général une matrice dite Hessienne.
- Elle peut converger vers un max si la fonction n'est pas convexe.
- L'inversion et le calcul du Hessien peut être couteux.
- Ca peut déraper si les dérivées secondes sont trop approximatives (matrice non définie).

Retour à la régression logistique : minimisation la NLL par la méthode de Newton

La logreg est le cas idéal pour Newton :

- Les dérivées 1ères et 2ndes sont faciles à calculer. item La NLL est convexe donc la convergence est assurée.

Régression logistique : minimisation la NLL par la méthode de Newton
Calcul des dérivées 1ères (gradient) :
On avait obtenu

$$\mathrm{NLL}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \left(1 - c^{(i)}\right) \left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right) - \log \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right). \tag{19}$$

On montre alors que

$$\frac{\partial}{\partial w_j}\mathrm{NLL}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} x_j^{(i)} \left(c^{(i)} - \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right)\right), \tag{20}$$

$$\frac{\partial}{\partial b}\mathrm{NLL}(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \left(c^{(i)} - \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(i)}\right)\right) \tag{21}$$

Régression logistique : minimisation la NLL par la méthode de Newton
Calcul des dérivées 1ères (gradient) sous forme matricielle :

Régression logistique : minimisation la NLL par la méthode de Newton
Calcul des dérivées 1ères (gradient) sous forme matricielle :
Soit **err** le vecteur rassemblant les écarts entre prédiction et réalité pour tous les éléments de $\mathcal{D}$:

$$\mathbf{err} = \begin{pmatrix} \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(1)}\right) - c^{(1)} \\ \vdots \\ \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(n)}\right) - c^{(n)} \end{pmatrix}. \tag{22}$$

Régression logistique : minimisation la NLL par la méthode de Newton
Calcul des dérivées 1ères (gradient) sous forme matricielle :
Soit **err** le vecteur rassemblant les écarts entre prédiction et réalité pour tous les éléments de $\mathcal{D}$:

$$\mathbf{err} = \begin{pmatrix} \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(1)}\right) - c^{(1)} \\ \vdots \\ \mathrm{sgm}\left(\boldsymbol{\theta}^T \cdot \mathbf{x}_+^{(n)}\right) - c^{(n)} \end{pmatrix}. \tag{22}$$

Introduisons également une matrice $\mathbf{X}$ qui aggrège tous les vecteurs d'exemple d'apprentissage en y ajoutant une ligne de "1" à la fin :

$$\mathbf{X} = \begin{pmatrix} \begin{bmatrix} \mathbf{x}^{(1)} \\ \end{bmatrix} & \cdots & \begin{bmatrix} \mathbf{x}^{(n)} \\ \end{bmatrix} \\ 1 & \cdots & 1 \end{pmatrix}. \tag{23}$$

Régression logistique : minimisation la NLL par la méthode de Newton
Calcul des dérivées 1ères (gradient) sous forme matricielle :

$$\frac{d}{d\theta}\mathrm{NLL}\left(\theta\right) = \mathbf{X} \cdot \mathbf{err}\ . \tag{24}$$

Régression logistique : minimisation la NLL par la méthode de Newton
Calcul des dérivées 1ères (gradient) sous forme matricielle :

$$\frac{d}{d\boldsymbol{\theta}}\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \mathbf{X}\cdot\mathbf{err}\,. \tag{24}$$

Pour les dérivées 2ndes, on a :

$$\frac{d^2}{d\boldsymbol{\theta}^2}\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \mathbf{X}\cdot\mathbf{S}\cdot\mathbf{X}^T\,, \tag{25}$$

Régression logistique : minimisation la NLL par la méthode de Newton
Calcul des dérivées 1ères (gradient) sous forme matricielle :

$$\frac{d}{d\boldsymbol{\theta}}\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \mathbf{X}\cdot\mathbf{err}\,. \tag{24}$$

Pour les dérivées 2ndes, on a :

$$\frac{d^2}{d\boldsymbol{\theta}^2}\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \mathbf{X}\cdot\mathbf{S}\cdot\mathbf{X}^T\,, \tag{25}$$

avec

$$\mathbf{S} = \begin{bmatrix} f_{\boldsymbol{\theta}}\left(\mathbf{x}^{(1)}\right)\left(1-f_{\boldsymbol{\theta}}\left(\mathbf{x}^{(1)}\right)\right) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & f_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right)\left(1-f_{\boldsymbol{\theta}}\left(\mathbf{x}^{(n)}\right)\right) \end{bmatrix} \tag{26}$$

Régression logistique : minimisation la NLL par la méthode de Newton

Bilan sous forme matricielle : la màj se fait selon

$$\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \left(\mathbf{XSX}^T\right)^{-1} \mathbf{X} \cdot \mathbf{err}\,. \tag{27}$$

Régression logistique : minimisation la NLL par la méthode de Newton

Bilan sous forme matricielle : la màj se fait selon

$$\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \left(\mathbf{XSX}^T\right)^{-1} \mathbf{X} \cdot \mathbf{err}\,. \tag{27}$$

Cette approche est équivalente à une autre appelée IRLS (*iteratively reweighted least squares*).

Régression logistique : extension multi-classes

- Supposons à présent un problème à plus de 2 classes : $\sharp\mathcal{C} = \ell \geq 3$.

Régression logistique : extension multi-classes

- Supposons à présent un problème à plus de 2 classes : $\sharp\mathcal{C} = \ell \geq 3$.
  Exemple : autorisation de carte bancaire

$$\mathcal{C} = \{\mathit{refus}, \mathit{electron}, \mathit{standard}, \mathit{gold}\}$$

Régression logistique : extension multi-classes

- Supposons à présent un problème à plus de 2 classes : $\sharp\mathcal{C} = \ell \geq 3$.
  Exemple : autorisation de carte bancaire

$$\mathcal{C} = \{\textit{refus}, \textit{electron}, \textit{standard}, \textit{gold}\}$$

- Le modèle $Y|X = \mathbf{x} \sim \mathrm{Ber}\left(\mathrm{score}\left(\mathbf{x}\right)\right)$ ne convient plus !

Régression logistique : extension multi-classes

- Supposons à présent un problème à plus de 2 classes : $\sharp\mathcal{C} = \ell \geq 3$.
  Exemple : autorisation de carte bancaire

$$\mathcal{C} = \{\textit{refus}, \textit{electron}, \textit{standard}, \textit{gold}\}$$

- Le modèle $Y|X = \mathbf{x} \sim \mathrm{Ber}\left(\mathrm{score}\left(\mathbf{x}\right)\right)$ ne convient plus !
- On doit passer à $Y|X = \mathbf{x} \sim \mathrm{Cat}\left(\boldsymbol{\pi}\left(\mathbf{x}\right)\right)$ avec

$$\boldsymbol{\pi}\left(\mathbf{x}\right) = F\left(\begin{bmatrix} \mathrm{score}_1\left(\mathbf{x}\right) \\ \vdots \\ \mathrm{score}_\ell\left(\mathbf{x}\right) \end{bmatrix}\right) \tag{28}$$

Régression logistique : extension multi-classes

- Supposons à présent un problème à plus de 2 classes : $\sharp\mathcal{C} = \ell \geq 3$.
  Exemple : autorisation de carte bancaire

$$\mathcal{C} = \{refus, electron, standard, gold\}$$

- Le modèle $Y|X = \mathbf{x} \sim \mathrm{Ber}\left(\mathrm{score}\left(\mathbf{x}\right)\right)$ ne convient plus !
- On doit passer à $Y|X = \mathbf{x} \sim \mathrm{Cat}\left(\boldsymbol{\pi}\left(\mathbf{x}\right)\right)$ avec

$$\boldsymbol{\pi}\left(\mathbf{x}\right) = F\left(\begin{bmatrix} \mathrm{score}_1\left(\mathbf{x}\right) \\ \vdots \\ \mathrm{score}_\ell\left(\mathbf{x}\right) \end{bmatrix}\right) \tag{28}$$

et

$$\mathrm{score}_i\left(\mathbf{x}\right) = \mathbf{w}_i^T \cdot \mathbf{x} + b_i. \tag{29}$$

Régression logistique : extension multi-classes

- Quelle fonction F peut envoyer les scores $\mathbf{w}_i^T \cdot \mathbf{x} + b_i$ vers un vecteur π contenant des probabilités ?

Régression logistique : extension multi-classes

- Quelle fonction F peut envoyer les scores $\mathbf{w}_i^T \cdot \mathbf{x} + b_i$ vers un vecteur π contenant des probabilités ?
- $\rightarrow$ la fonction smax *softmax* peut faire cela :

$$\mathrm{smax}\left(\mathbf{x}; \boldsymbol{\Theta}\right) = \begin{bmatrix} \frac{\exp \mathbf{w}_1^T \cdot \mathbf{x} + b_1}{\sum_{k=1}^{\ell} \exp\left(\mathbf{w}_k^T \cdot \mathbf{x} + b_k\right)} \\ \vdots \\ \frac{\exp \mathbf{w}_\ell^T \cdot \mathbf{x} + b_\ell}{\sum_{k=1}^{\ell} \exp\left(\mathbf{w}_k^T \cdot \mathbf{x} + b_k\right)} \end{bmatrix} \tag{30}$$

Régression logistique : extension multi-classes

- Quelle fonction F peut envoyer les scores $\mathbf{w}_i^T \cdot \mathbf{x} + b_i$ vers un vecteur π contenant des probabilités ?
- $\rightarrow$ la fonction smax *softmax* peut faire cela :

$$\mathrm{smax}\left(\mathbf{x}; \boldsymbol{\Theta}\right) = \begin{bmatrix} \frac{\exp \mathbf{w}_1^T \cdot \mathbf{x} + b_1}{\sum_{k=1}^{\ell} \exp\left(\mathbf{w}_k^T \cdot \mathbf{x} + b_k\right)} \\ \vdots \\ \frac{\exp \mathbf{w}_\ell^T \cdot \mathbf{x} + b_\ell}{\sum_{k=1}^{\ell} \exp\left(\mathbf{w}_k^T \cdot \mathbf{x} + b_k\right)} \end{bmatrix} \tag{30}$$

- $\boldsymbol{\Theta}$ est une matrice rassemblant tous les paramètres.

Régression logistique : extension multi-classes

- Quelle fonction F peut envoyer les scores $\mathbf{w}_i^T \cdot \mathbf{x} + b_i$ vers un vecteur π contenant des probabilités ?
- $\rightarrow$ la fonction smax *softmax* peut faire cela :

$$\mathrm{smax}\left(\mathbf{x}; \boldsymbol{\Theta}\right) = \begin{bmatrix} \frac{\exp \mathbf{w}_1^T \cdot \mathbf{x} + b_1}{\sum_{k=1}^{\ell} \exp\left(\mathbf{w}_k^T \cdot \mathbf{x} + b_k\right)} \\ \vdots \\ \frac{\exp \mathbf{w}_\ell^T \cdot \mathbf{x} + b_\ell}{\sum_{k=1}^{\ell} \exp\left(\mathbf{w}_k^T \cdot \mathbf{x} + b_k\right)} \end{bmatrix} \tag{30}$$

- $\boldsymbol{\Theta}$ est une matrice rassemblant tous les paramètres.
- Elle généralise la sigmoïde.

Régression logistique : extension multi-classes

- Quelle fonction F peut envoyer les scores $\mathbf{w}_i^T \cdot \mathbf{x} + b_i$ vers un vecteur π contenant des probabilités ?
- $\rightarrow$ la fonction smax *softmax* peut faire cela :

$$\mathrm{smax}\left(\mathbf{x}; \boldsymbol{\Theta}\right) = \begin{bmatrix} \frac{\exp \mathbf{w}_1^T \cdot \mathbf{x} + b_1}{\sum_{k=1}^{\ell} \exp\left(\mathbf{w}_k^T \cdot \mathbf{x} + b_k\right)} \\ \vdots \\ \frac{\exp \mathbf{w}_\ell^T \cdot \mathbf{x} + b_\ell}{\sum_{k=1}^{\ell} \exp\left(\mathbf{w}_k^T \cdot \mathbf{x} + b_k\right)} \end{bmatrix} \tag{30}$$

- $\boldsymbol{\Theta}$ est une matrice rassemblant tous les paramètres.
- Elle généralise la sigmoïde.
- Elle appartient à la famille des distributions de Boltzmann.

Régression logistique : extension multi-classes

- D'où vient le nom *softmax* ?

Régression logistique : extension multi-classes

- D'où vient le nom *softmax* ?
- Supposons $\ell = 3$ et

$$\mathbf{w}_1^T \cdot \mathbf{x} + b_1 = 0.1 \quad (31)$$
$$\mathbf{w}_2^T \cdot \mathbf{x} + b_2 = 0.1 \quad (32)$$
$$\mathbf{w}_3^T \cdot \mathbf{x} + b_3 = 100 \quad (33)$$

Régression logistique : extension multi-classes

- D'où vient le nom *softmax* ?
- Supposons $\ell = 3$ et

$$\mathbf{w}_1^T \cdot \mathbf{x} + b_1 = 0.1 \quad (31)$$
$$\mathbf{w}_2^T \cdot \mathbf{x} + b_2 = 0.1 \quad (32)$$
$$\mathbf{w}_3^T \cdot \mathbf{x} + b_3 = 100 \quad (33)$$

- On aurait alors :

$$\mathrm{smax}\left(\mathbf{x}; \mathbf{W}\right) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (34)$$

Régression logistique : extension multi-classes

- D'où vient le nom *softmax* ?
- Supposons $\ell = 3$ et

$$\mathbf{w}_1^T \cdot \mathbf{x} + b_1 = 0.1 \quad (31)$$
$$\mathbf{w}_2^T \cdot \mathbf{x} + b_2 = 0.1 \quad (32)$$
$$\mathbf{w}_3^T \cdot \mathbf{x} + b_3 = 100 \quad (33)$$

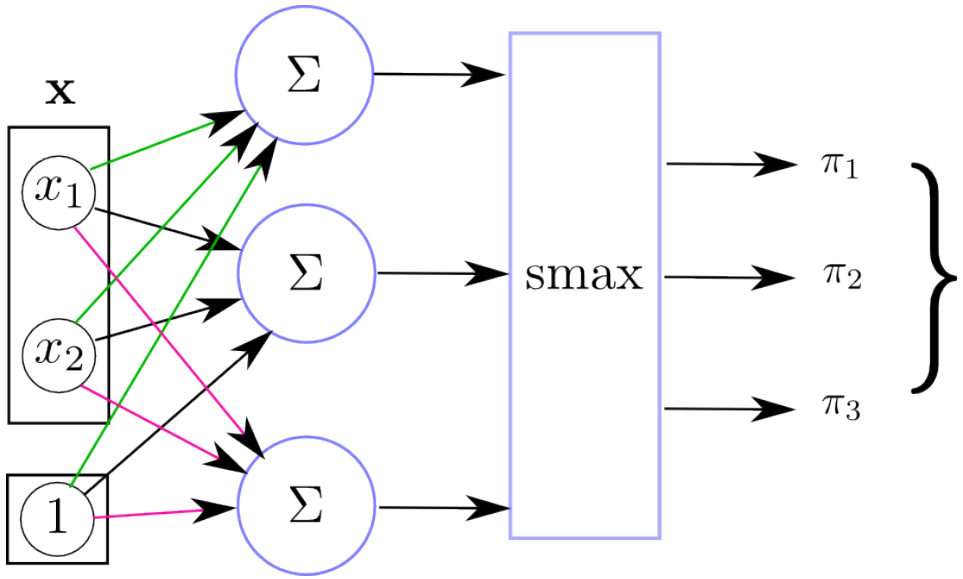
- On aurait alors :

$$\text{smax}\left(\mathbf{x}; \mathbf{W}\right) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (34)$$

- En conclusion, quand une classe i a un gros score pour l'exemple $\mathbf{x}$ la proba que $y = i$ est proche de 1.

Régression logistique : extension multi-classes

- Schématiquement, ça donne :



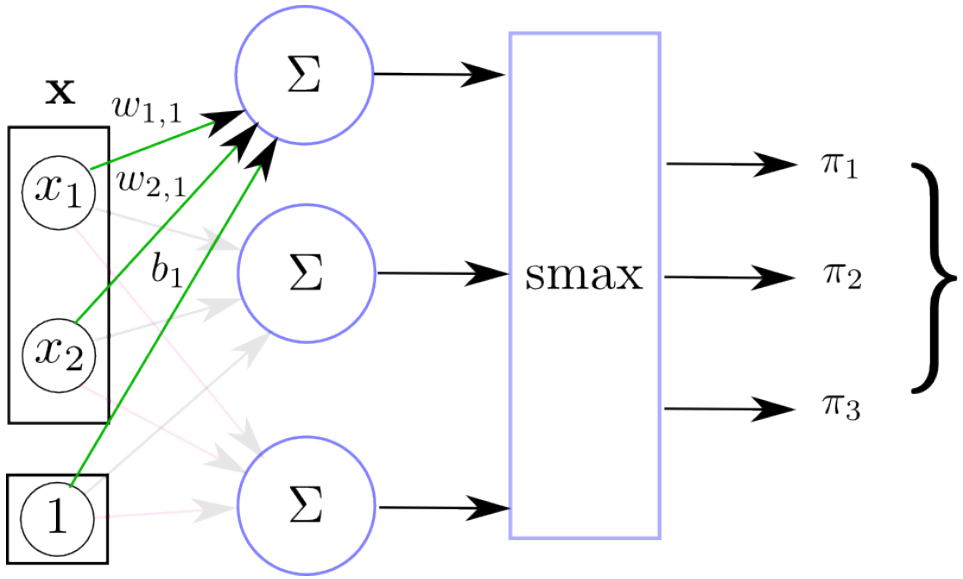
$f_\Theta(\mathbf{x})$
prédiction pour l'exemple $\mathbf{x}$:
probabilités

Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ?

Régression logistique : extension multi-classes

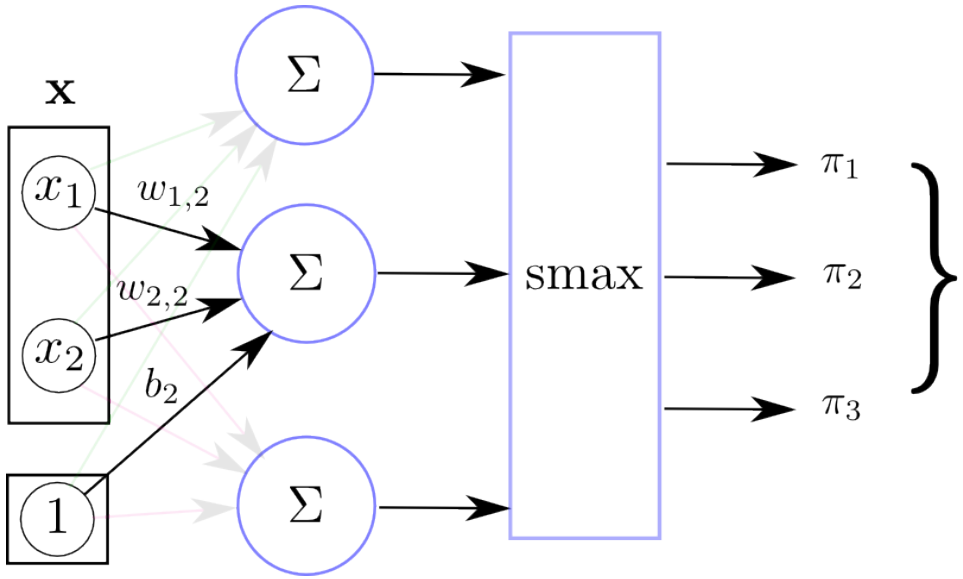
- Qu'advient-il des paramètres ?



$f_\Theta(\mathbf{x})$
prédiction pour l'exemple $\mathbf{x}$:
probabilités

Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ?



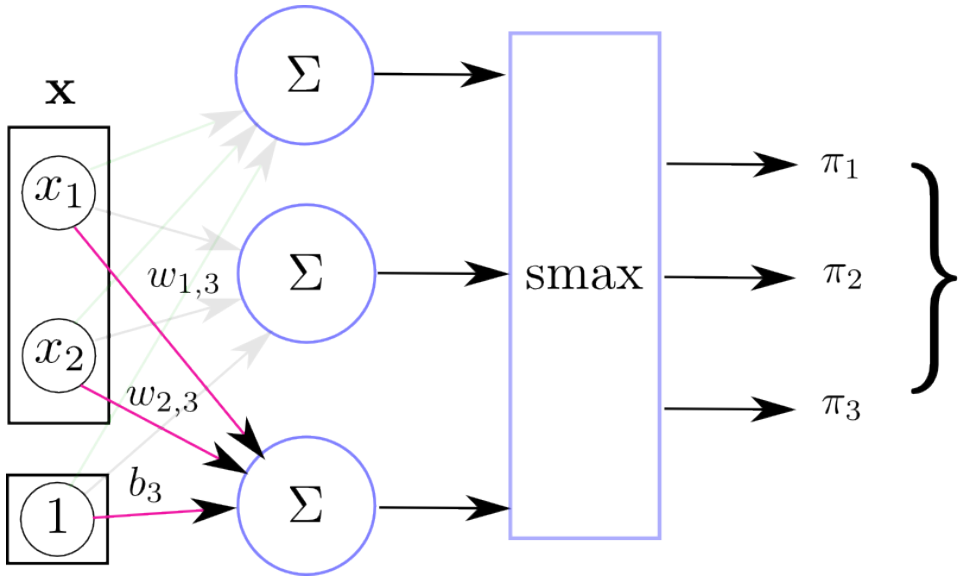
$f_\Theta(\mathbf{x})$
prédiction pour l'exemple $\mathbf{x}$:
probabilités

Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ?

Régression logistique : extension multi-classes

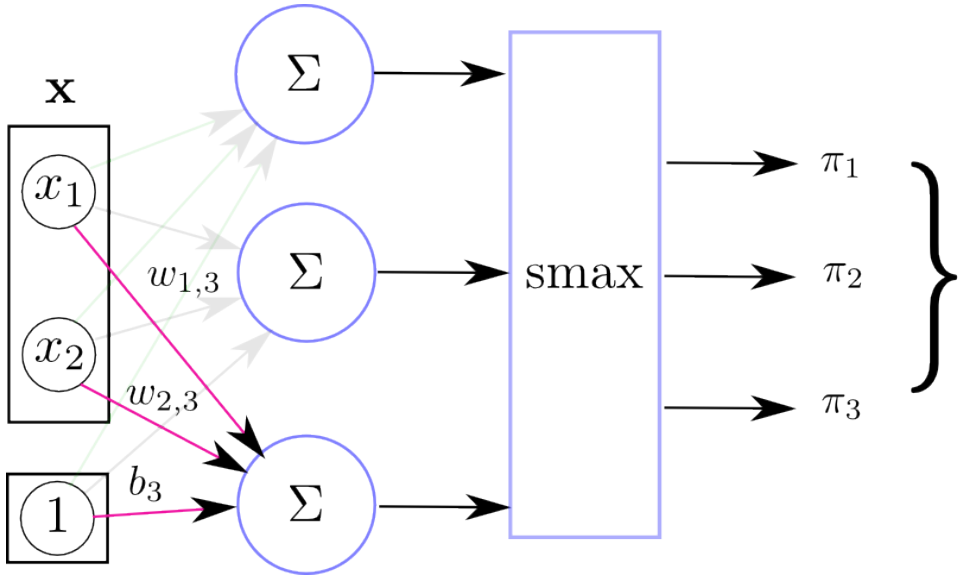
- Qu'advient-il des paramètres ?



$f_\Theta(\mathbf{x})$
prédiction pour l'exemple $\mathbf{x}$:
probabilités

Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ?



$f_\Theta(\mathbf{x})$
prédiction pour l'exemple $\mathbf{x}$:
probabilités

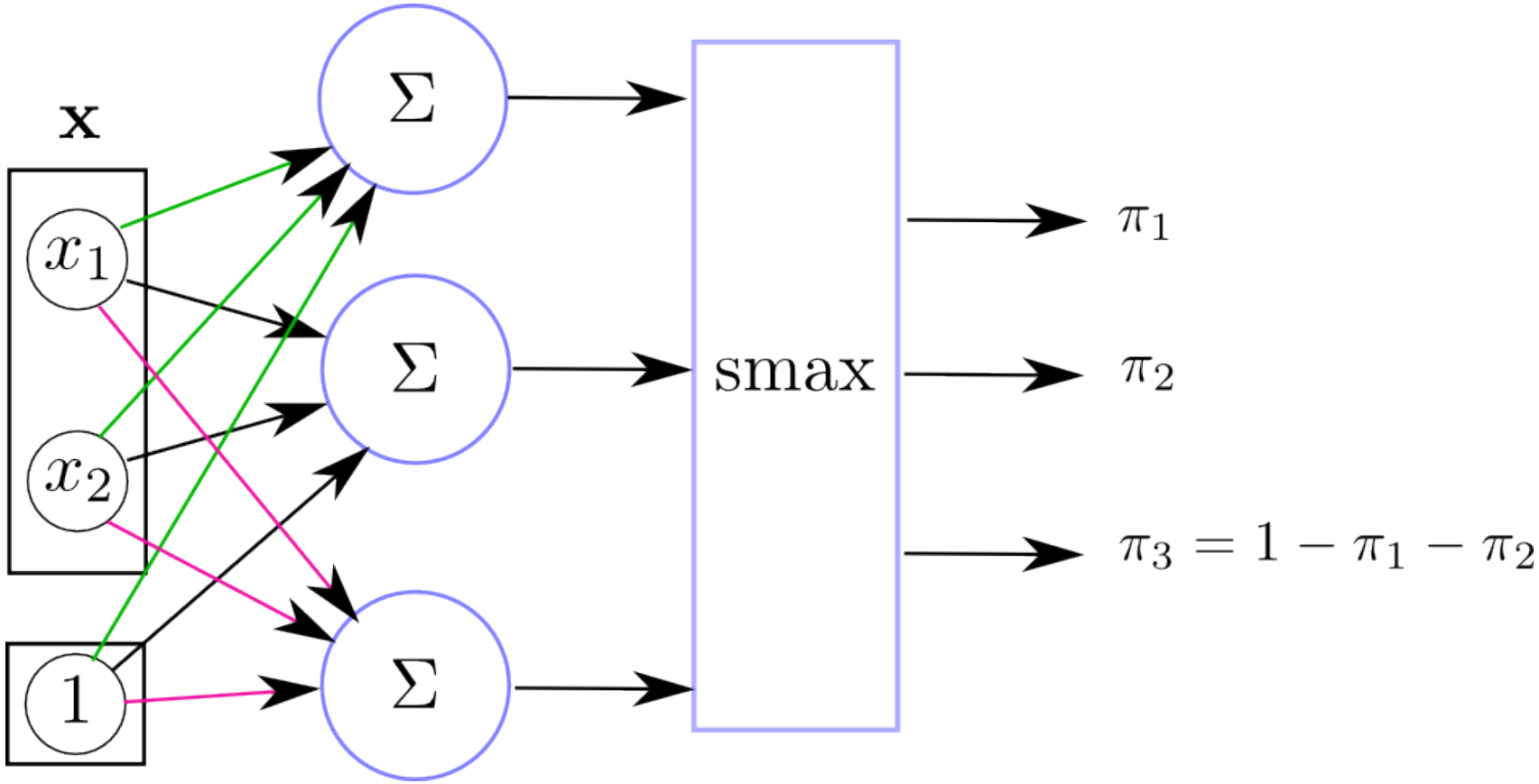
Donc $\boldsymbol{\Theta} =$

Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ? Regardons de plus près ...

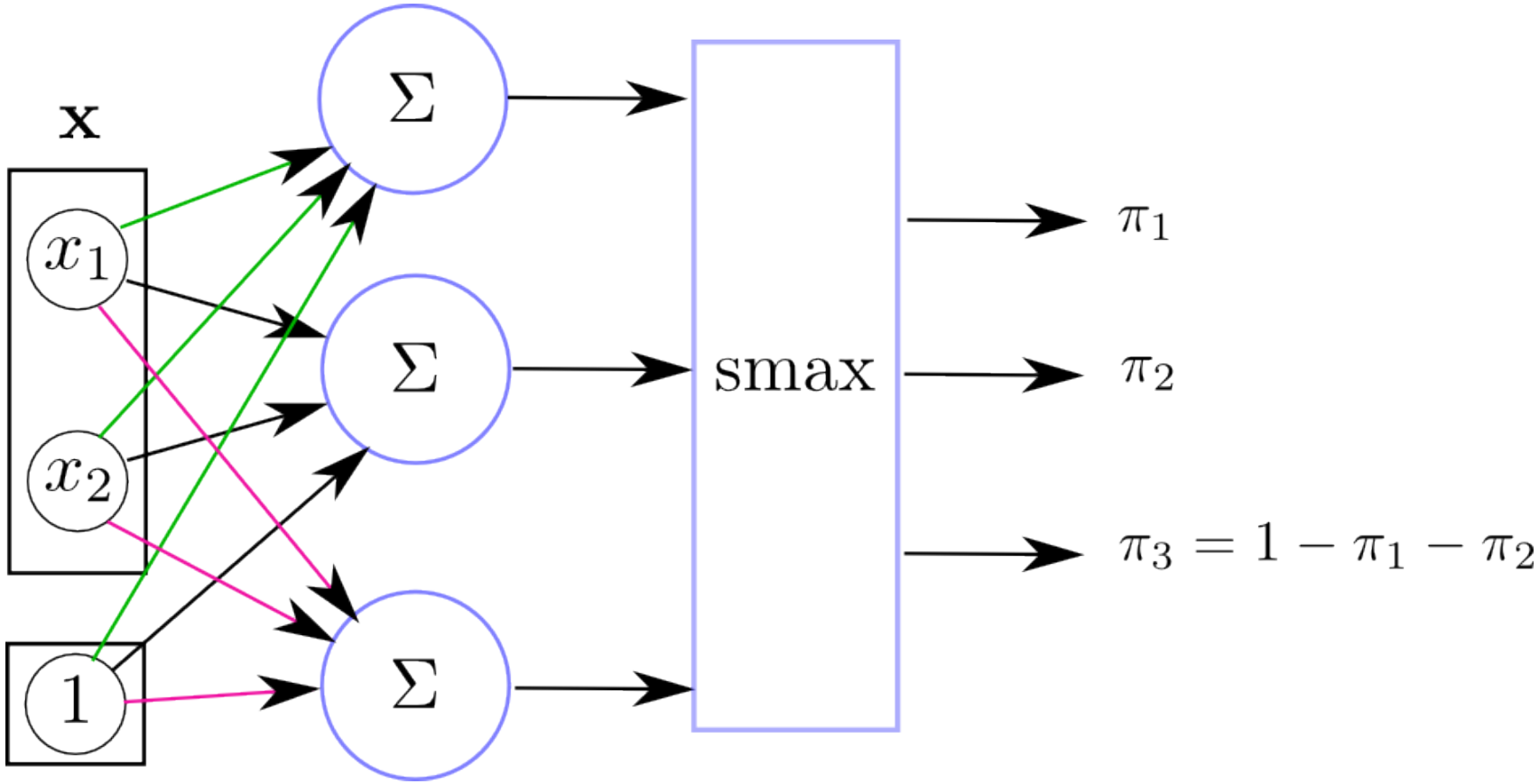
Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ? Regardons de plus près ...



Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ? Regardons de plus près ...



C'est comme si une des sorties du *softmax* de servait à rien !

Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ? Regardons de plus près ...
- On peut montrer que pour tout vecteur ψ de même taille que les θ_i, si la matrice Θ^* minimise la NLL alors, la matrice suivant aussi :

$$\Theta^* - \left(\begin{bmatrix} \\ \psi \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \psi \\ \\ \end{bmatrix} \right) \tag{35}$$

- Par convention, on choisit $\psi = \theta_\ell$ et on supprime la dernière colonne de Θ dans le problème de minimisation de la NLL.

Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ? Regardons de plus près ...
- On peut montrer que pour tout vecteur ψ de même taille que les θ_i, si la matrice Θ^* minimise la NLL alors, la matrice suivant aussi :

$$\Theta^* - \left(\begin{bmatrix} \\ \psi \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \psi \\ \\ \end{bmatrix} \right) \tag{35}$$

- Par convention, on choisit $\psi = \theta_\ell$ et on supprime la dernière colonne de Θ dans le problème de minimisation de la NLL.

Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ? Regardons de plus près ...
- On peut montrer que pour tout vecteur ψ de même taille que les $\boldsymbol{\theta}_i$, si la matrice $\boldsymbol{\Theta}^*$ minimise la NLL alors, la matrice suivant aussi :

$$\boldsymbol{\Theta}^* - \left(\begin{bmatrix} \\ \psi \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \psi \\ \\ \end{bmatrix} \right) \tag{35}$$

- Par convention, on choisit $\psi = \boldsymbol{\theta}_\ell$ et on supprime la dernière colonne de $\boldsymbol{\Theta}$ dans le problème de minimisation de la NLL.

Régression logistique : extension multi-classes

- Qu'advient-il de la descente de gradient ?

Régression logistique : extension multi-classes

- Qu'advient-il de la descente de gradient ?
- La même approche reste valable, mais les dérivées sont plus nombreuses.

Régression logistique : extension multi-classes

- Qu'advient-il de la descente de gradient ?
- La même approche reste valable, mais les dérivées sont plus nombreuses.
- Le plus simple est de calculer à chaque itération t, les dérivées séparemment pour chaque vecteur $\boldsymbol{\theta}_i$ selon :

$$\frac{\partial}{\partial \boldsymbol{\theta}_i} \mathrm{NLL}\left(\boldsymbol{\Theta}\right) = -\sum_{j=1}^{n} \mathbf{x}^{(j)} \left(\mathbb{I}_i \left(y^{(j)} \right) - \pi_i \right) \tag{36}$$

Plan du chapitre

Régression logistique et Big Data

Régression logistique et Big Data

- Imaginons que $n = 10^9$ (ex : nombre d'utilisateurs de Facebook).

Régression logistique et Big Data

- Imaginons que $n = 10^9$ (ex : nombre d'utilisateurs de Facebook).
- Pour une itération de la descente de gradient, je dois calculer :

$$\frac{d}{d\theta}\mathrm{NLL}(\theta) = \mathbf{X} \cdot \mathbf{err} \,. \tag{37}$$

Régression logistique et Big Data

- Imaginons que $n = 10^9$ (ex : nombre d'utilisateurs de Facebook).
- Pour une itération de la descente de gradient, je dois calculer :

$$\frac{d}{d\theta}\mathrm{NLL}(\theta) = \mathbf{X} \cdot \mathbf{err} \,. \tag{37}$$

- Problème : la matrice $\mathbf{X}$ ne tient pas sur ma mémoire :

$$\mathbf{X} = \left(\boxed{\begin{bmatrix} \\ \mathbf{x}^{(1)} \\ \\ \end{bmatrix}} \cdots \quad \cdots \cdots \begin{bmatrix} \\ \mathbf{x}^{(10^9)} \\ \\ \end{bmatrix} \right) \tag{38}$$

Régression logistique et Big Data

- Reformulons ce calcul matriciel :

$$\frac{d}{d\theta}\mathrm{NLL}(\theta) = \mathbf{X} \cdot \mathbf{err} = \sum_{i=1}^{n} err_i \times \mathbf{x}^{(i)}. \tag{39}$$

Régression logistique et Big Data

- Reformulons ce calcul matriciel :

$$\frac{d}{d\theta}\mathrm{NLL}(\theta) = \mathbf{X} \cdot \mathbf{err} = \sum_{i=1}^{n} err_i \times \mathbf{x}^{(i)}. \tag{39}$$

- Bonne nouvelle : je n'ai pas besoin de charger en mémoire tous mes exemples d'un coup !

Régression logistique et Big Data

- Reformulons ce calcul matriciel :

$$\frac{d}{d\boldsymbol{\theta}}\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \mathbf{X}\cdot\mathbf{err} = \sum_{i=1}^{n} err_i \times \mathbf{x}^{(i)}. \tag{39}$$

- Bonne nouvelle : je n'ai pas besoin de charger en mémoire tous mes exemples d'un coup !
- mais finalement... pourquoi attendre d'avoir vu tous mes exemples pour mettre à jour $\boldsymbol{\theta}$?

Régression logistique et Big Data

- Reformulons ce calcul matriciel :

$$\frac{d}{d\boldsymbol{\theta}}\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \mathbf{X}\cdot\mathbf{err} = \sum_{i=1}^{n} err_i \times \mathbf{x}^{(i)}. \tag{39}$$

- Bonne nouvelle : je n'ai pas besoin de charger en mémoire tous mes exemples d'un coup !
- mais finalement... pourquoi attendre d'avoir vu tous mes exemples pour mettre à jour $\boldsymbol{\theta}$?
- Peut on faire la màj exemple par exemple selon :

$$\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \eta_t \times err_i \times \mathbf{x}^{(i)} \ ? \tag{40}$$

Régression logistique et Big Data

- Reformulons ce calcul matriciel :

$$\frac{d}{d\boldsymbol{\theta}}\mathrm{NLL}\left(\boldsymbol{\theta}\right) = \mathbf{X}\cdot\mathbf{err} = \sum_{i=1}^{n} err_i \times \mathbf{x}^{(i)}. \tag{39}$$

- Bonne nouvelle : je n'ai pas besoin de charger en mémoire tous mes exemples d'un coup !
- mais finalement... pourquoi attendre d'avoir vu tous mes exemples pour mettre à jour $\boldsymbol{\theta}$?
- Peut on faire la màj exemple par exemple selon :

$$\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \eta_t \times err_i \times \mathbf{x}^{(i)} \ ? \tag{40}$$

$\rightarrow$ Oui, ça s'appelle faire une descente de gradient stochastique.

Régression logistique et Big Data : gradient stochastique

- Pourquoi stochastique ?

Régression logistique et Big Data : gradient stochastique

- Pourquoi stochastique ?
- Imaginons tirer $U \sim \mathcal{U}_{\{1;n\}}$ uniformément afin de choisir au hasard une de nos données $\mathbf{x}^{(U)}$.

Régression logistique et Big Data : gradient stochastique

- Pourquoi stochastique ?
- Imaginons tirer $U \sim \mathcal{U}_{\{1;n\}}$ uniformément afin de choisir au hasard une de nos données $\mathbf{x}^{(U)}$.
- Le gradient de l'erreur pour cet exemple est donc

$$g(U) = err_U \times \mathbf{x}^{(U)}. \tag{41}$$

Régression logistique et Big Data : gradient stochastique

- Pourquoi stochastique ?
- Imaginons tirer $U \sim \mathcal{U}_{\{1;n\}}$ uniformément afin de choisir au hasard une de nos données $\mathbf{x}^{(U)}$.
- Le gradient de l'erreur pour cet exemple est donc

$$g(U) = err_U \times \mathbf{x}^{(U)}. \tag{41}$$

- Prenons l'espérance sous la loi de U :

$$\begin{aligned} \mathbb{E}_U[g] &= \sum_{u=1}^{n} err_u \times \mathbf{x}^{(u)} p_U(u), \\ &= \\ &= \end{aligned} \tag{42}$$

Régression logistique et Big Data : gradient stochastique

- Pourquoi stochastique ?
- Imaginons tirer $U \sim \mathcal{U}_{\{1;n\}}$ uniformément afin de choisir au hasard une de nos données $\mathbf{x}^{(U)}$.
- Le gradient de l'erreur pour cet exemple est donc

$$g(U) = err_U \times \mathbf{x}^{(U)}. \tag{41}$$

- Prenons l'espérance sous la loi de U :

$$\begin{aligned} \mathbb{E}_U[g] &= \sum_{u=1}^{n} err_u \times \mathbf{x}^{(u)} p_U(u), \\ &= \\ &= \end{aligned} \tag{42}$$

- Le gradient qu'on cherche est l'espérance de celui pris sur un exemple au hasard.

Régression logistique et Big Data : gradient stochastique

- Est-ce que ça converge vers $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathrm{NLL}(\boldsymbol{\theta})$?

Régression logistique et Big Data : gradient stochastique

- Est-ce que ça converge vers $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathrm{NLL}(\boldsymbol{\theta})$?
- J'ai :

$$\mathbb{E}[\text{màj}] = \text{cible} \tag{43}$$

$$\Leftrightarrow \mathbb{E}_U\left[\text{gradient pour } \mathbf{x}^{(U)} \text{ seul}\right] = \frac{d}{d\boldsymbol{\theta}}\mathrm{NLL}(\boldsymbol{\theta}). \tag{44}$$

Régression logistique et Big Data : gradient stochastique

- Est-ce que ça converge vers $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathrm{NLL}(\boldsymbol{\theta})$?
- J'ai :

$$\mathbb{E}[\text{màj}] = \text{cible} \tag{43}$$

$$\Leftrightarrow \mathbb{E}_U\left[\text{gradient pour } \mathbf{x}^{(U)} \text{ seul}\right] = \frac{d}{d\boldsymbol{\theta}}\mathrm{NLL}(\boldsymbol{\theta}). \tag{44}$$

- Je peux alors invoquer le résultat suivant :

Conditions de Robbins-Monroe

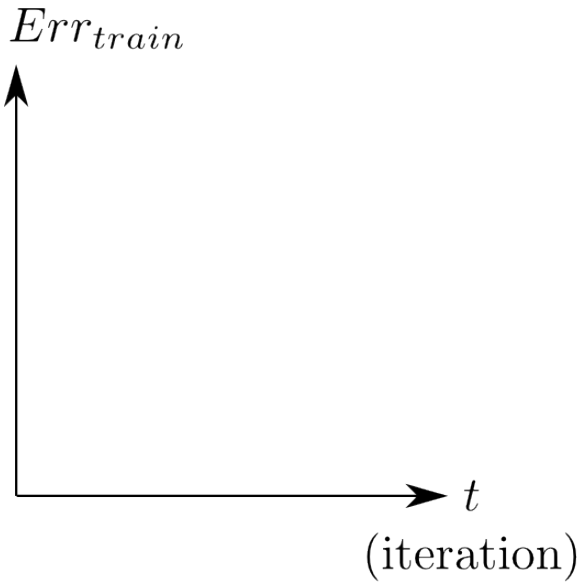
Si la suite η_t des *learning rates* est telle que $\sum_{t\geq 0} \eta_t = \infty$ et $\sum_{t\geq 0} \eta_t^2 < \infty$ alors l'algorithme du gradient stochastique converge vers l'optimum $\boldsymbol{\theta}^*$.

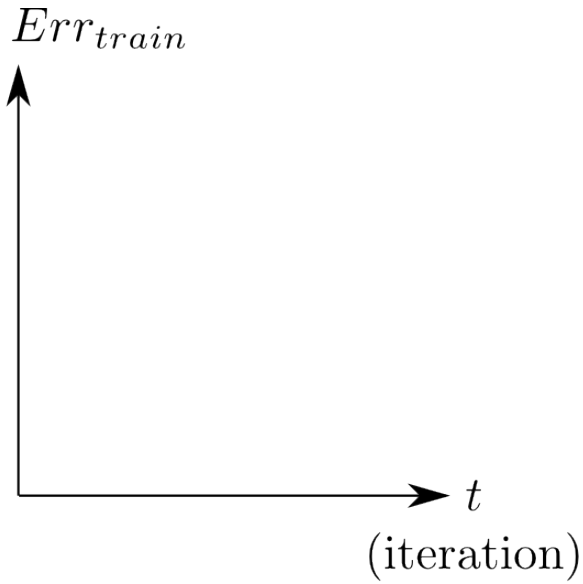
Régression logistique et Big Data : gradient stochastique / Procédure

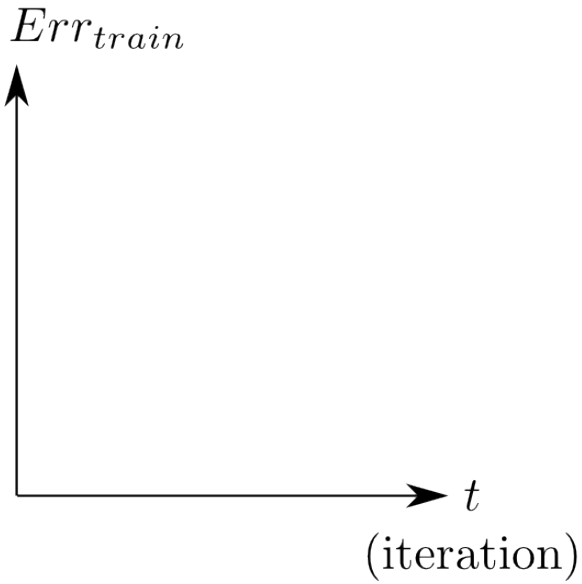
SGD

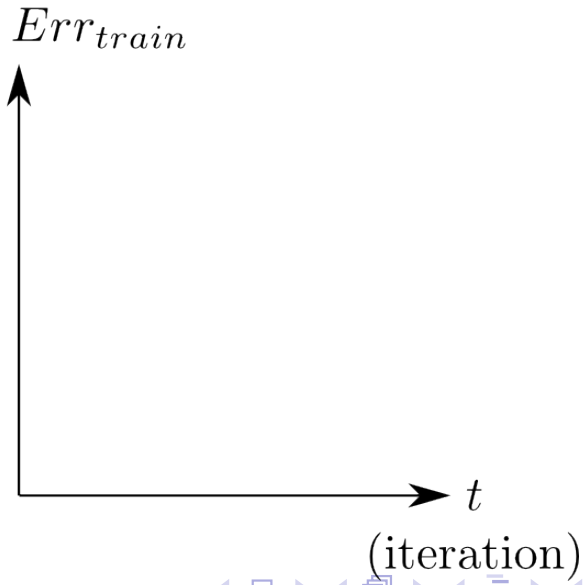
Initialiser $\boldsymbol{\theta}_0$ et η_0.
while pas convergé **do**
  if (i multiple de n) Mélanger les données aléatoirement (*Shuffle*).
  Calculer le gradient sur l'exemple $\mathbf{x}^{(i)}$: $\mathbf{g} \leftarrow err_i \times \mathbf{x}^{(i)}$.
  Mettre à jour les paramètres : $\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \eta_t \times \mathbf{g}$.
  Mettre à jour le *learning rate* : $\eta_{t+1} \longleftarrow \frac{a}{t+b}$
  $i \leftarrow i + 1$
end while

Régression logistique et Big Data : grad. stoch. / 4 situations en pratique









Messages importants du chapitre :

- La régression logistique offre une estimation de $p_{Y|X;\theta}$ à partir des données. C'est donc un modèle discriminatif.
- Elle fonctionne pour une perte L appelée entropie croisée.
- Avec cette perte, la NLL de la régression logistique est l'erreur Err_{train}.
- La NLL doit être minimisée par rapport aux paramètres θ du modèle de la régression logistique.
- Puisqu'il n'y a pas de solution analytique, on utilise la descente de gradient.
- Quand le dataset est grand, on préfère la descente de gradient stochastique.