

# A2DI: Régression logistique

John Klein

Lille1 Université - CRIStAL UMR CNRS 9189



**UFR IEEA**  
Informatique, Electronique  
Electrotechnique, Informatique

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

- En théorie on veut minimiser  $Err_{gen}$ .
- En pratique on minimisera  $Err_{train}$  tout en s'assurant que  $Err_{train}$  ne dévie pas de  $Err_{gen}$ .
- 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  $Err_{gen}$ .
- En pratique on minimisera  $Err_{train}$  tout en s'assurant que  $Err_{train}$  ne dévie pas de  $Err_{gen}$ .
- 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

- 1 Origines du modèle
- 2 Descente de gradient
- 3 Descente de gradient stochastique
- 4 Conclusions

Reprenons un **exemple** habituel :

- Soit un problème de classification avec  $\mathbb{Y} = \{0; 1\}$  où 0 représente l'hypothèse d'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 d'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 représente 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 d'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 représente un information du style : *salaire*, *découvert max*, *endettement*, etc.
- On se doute qu'un **score** du style :

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

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

Reprenons un **exemple** habituel :

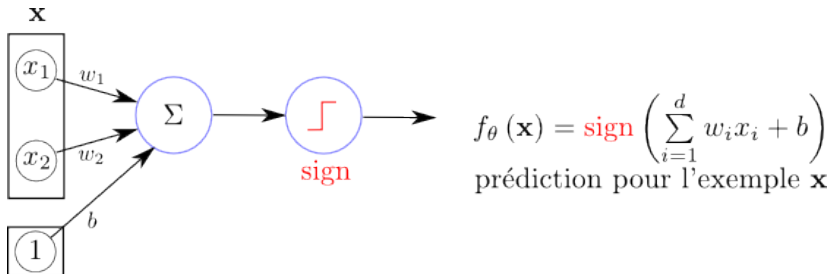
- Soit un problème de classification avec  $\mathbb{Y} = \{0; 1\}$  où 0 représente l'hypothèse d'un **crédit** n'est **pas accordé** et 1 qu'un crédit est **accordé**.
- Soit  $\mathbb{X}$  l'espace des attributs brutes où chaque **dimension**  $x_j$  est un nombre réel représente un information du style : *salaires*, *découvert max*, *endettement*, etc.
- On se doute qu'un **score** du style :

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

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 !

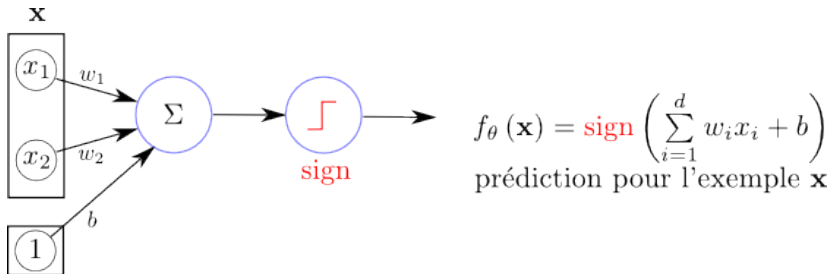


- Partir sur cette idée revient à choisir un **modèle linéaire**.
- .. mais, le **score** n'est ni une **classe**, ni une **probabilité** d'appartenance à une classe.
- Avec le perceptron, on avait décidé d'appliquer la fonction **sign** au **score** :



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.



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 ?

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

- 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**.

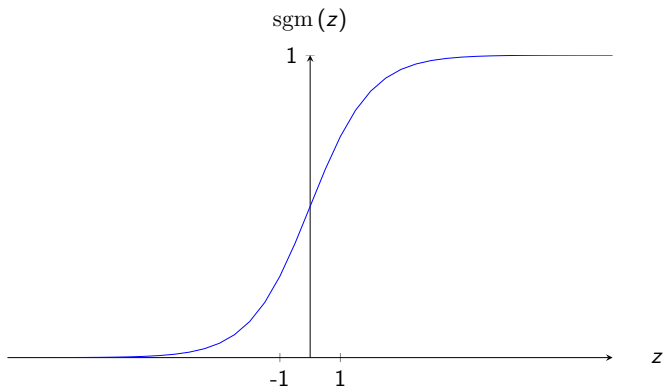
- Une fonction qui permet d'obtenir cela est la fonction **sigmoïde**.
- La fonction **sigmoïde** `sgm` est une fonction continue de  $\mathbb{R}$  dans  $[0; 1]$ , telle que

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

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

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

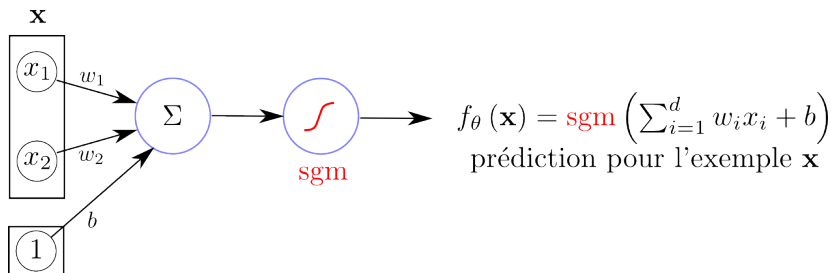
- Voici le graphe de la fonction **sigmoïde** :



Le modèle complet de la **régression logistique** est :

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

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



Soit  $\theta$  le vecteur qui contient tous les paramètres du modèle :

$$\theta = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix} \quad (5)$$



Soit  $\theta$  le vecteur qui contient tous les paramètres du modèle :

$$\theta = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix} \quad (5)$$

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

Soit  $\theta$  le vecteur qui contient **tous les paramètres** du modèle :

$$\theta = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix} \quad (5)$$

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

→ Celle qui maximise

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

Reformulons sous une forme plus compacte :

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

(7)

Reformulons sous une forme plus compacte :

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

Selon le modèle de la **régression logistique**, quelle valeur de  $\theta$  explique le mieux tout mon ensemble d'apprentissage ?

Reformulons sous une forme plus compacte :

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

Selon le modèle de la **régression logistique**, quelle valeur de  $\theta$  explique le mieux tout mon ensemble d'apprentissage ?

→ Celle qui maximise la vraisemblance :

Reformulons sous une forme plus compacte :

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

Selon le modèle de la **régression logistique**, quelle valeur de  $\boldsymbol{\theta}$  explique le mieux tout mon ensemble d'apprentissage ?

→ Celle qui maximise la vraisemblance :

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

Passons à la NLL ( $\theta$ )

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

Passons à la NLL ( $\theta$ )

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

Cette équation peut s'interpréter 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 NLL ( $\theta$ )

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

Cette équation peut s'interpréter 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  
 $p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\theta}(c) = \text{Ber}\left(f_{\theta}(\mathbf{x}^{(i)})\right).$

Passons à la NLL ( $\theta$ )

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

Cette équation peut s'interpréter 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  
 $p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\theta}(c) = \text{Ber}(f_{\theta}(\mathbf{x}^{(i)}))$ .

$$\text{NLL}(\theta) = - \sum_{i=1}^n H(\hat{p}^{(i)}, p_{Y|\mathbf{X}=\mathbf{x}^{(i)};\theta}), \quad (10)$$

$$= \text{Err}_{\text{train}} \quad (11)$$

Passons à la NLL ( $\theta$ )

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

Posons :

$$\mathbf{x}_+^{(i)} = \begin{bmatrix} x_{+,1}^{(i)} \\ \vdots \\ x_{+,d}^{(i)} \\ 1 \end{bmatrix} \quad (13)$$

On a NLL ( $\theta$ )

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

On continue et NLL( $\theta$ )

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

Au final,

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

Au final,

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

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

$$\arg \min_{\boldsymbol{\theta}} \text{NLL}(\boldsymbol{\theta}).$$

Au final,

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

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

$$\arg \min_{\boldsymbol{\theta}} \text{NLL}(\boldsymbol{\theta}).$$

On va donc devoir utiliser **algorithme d'optimisation**.

## Modèles linéaires : bilan



## Modèles linéaires : bilan

| Régression Linéaire | Perceptron | Régression Logistique

Modèles linéaires : bilan

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

Modèles linéaires : bilan

$f_{\theta}(\mathbf{x})$ Pred.	<b>Régression Linéaire</b> $\mathbf{w}^T \cdot \mathbf{x} + b$	<b>Perceptron</b> $\text{sign}(\mathbf{w}^T \cdot \mathbf{x} + b)$	<b>Régression Logistique</b> $\text{sgm}(\mathbf{w}^T \cdot \mathbf{x} + b)$
Tâche	Régression	Classification	Classification

## Modèles linéaires : bilan

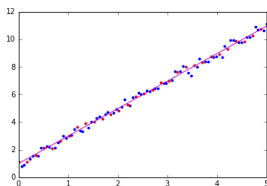
	Régression Linéaire	Perceptron	Régression Logistique
$f_{\theta}(\mathbf{x})$ Pred.	$\mathbf{w}^T \cdot \mathbf{x} + b$	$\text{sign}(\mathbf{w}^T \cdot \mathbf{x} + b)$	$\text{sgm}(\mathbf{w}^T \cdot \mathbf{x} + b)$
Tâche	Régression	Classification	Classification
Sortie	valeur	classe	proba de classe

## Modèles linéaires : bilan

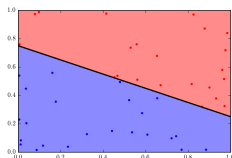
	Régression Linéaire	Perceptron	Régression Logistique
$f_{\theta}(\mathbf{x})$ Pred.	$\mathbf{w}^T \cdot \mathbf{x} + b$	$\text{sign}(\mathbf{w}^T \cdot \mathbf{x} + b)$	$\text{sgm}(\mathbf{w}^T \cdot \mathbf{x} + b)$
Tâche	Régression	Classification	Classification
Sortie	valeur	classe	proba de classe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

# Modèles linéaires : bilan

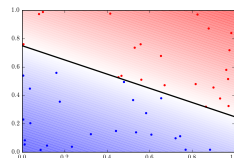
## Régression Linéaire



## Perceptron



## Régression Logistique



# Plan du chapitre

- 1 Origines du modèle
- 2 Descente de gradient**
- 3 Descente de gradient stochastique
- 4 Conclusions

Retour à la **régression logistique** :

Prochain défi : **minimiser** la NLL

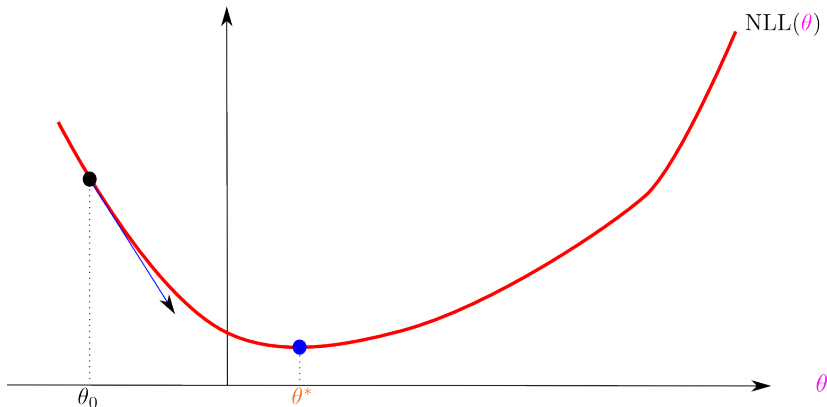
- Impossible de calculer le MLE analytiquement.



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  $\theta$  selon :

$$\theta_{t+1} \leftarrow \theta_t - \eta \times \frac{d}{d\theta} \text{NLL}(\theta),$$

## Minimisation de la NLL par descente de gradient



Pause Optim !

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

$$\theta_{t+1} \leftarrow \theta_t - \eta \times \frac{d}{d\theta} \text{NLL}(\theta),$$

avec  $\eta$  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 !

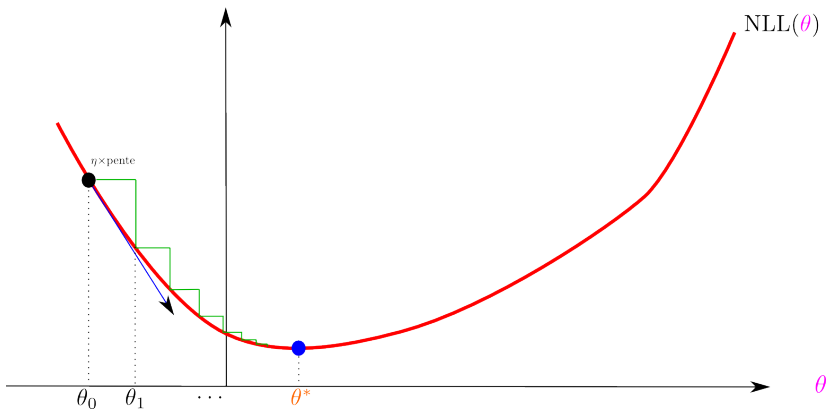
- Quand  $\eta$  est bien choisi, l'algorithme fonctionne ainsi :

## Minimisation de la NLL par descente de gradient : principe graphique



Pause Optim !

- Quand  $\eta$  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 !

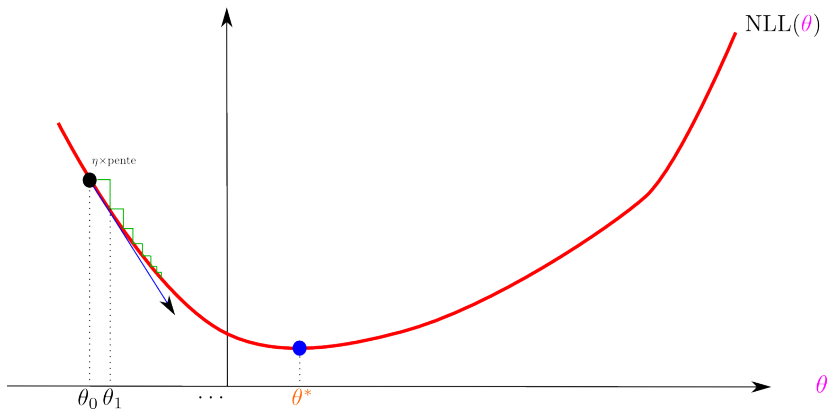
- Quand  $\eta$  est choisi trop petit :

# Minimisation de la NLL par descente de gradient : principe graphique



Pause Optim !

- Quand  $\eta$  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 !

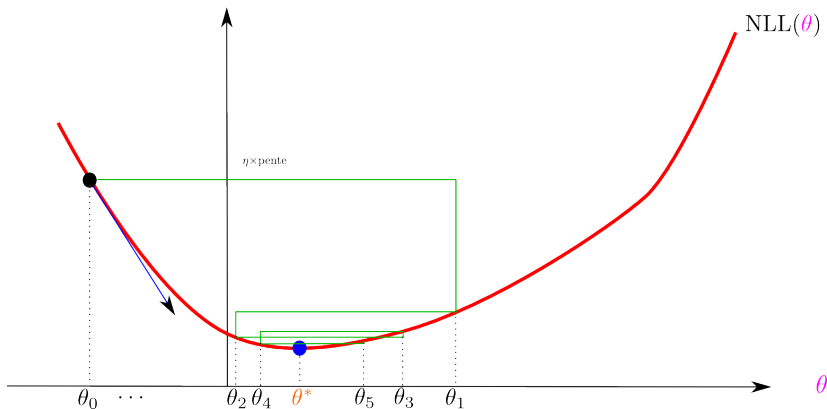
- Quand  $\eta$  est choisi trop grand :

## Minimisation de la NLL par descente de gradient : principe graphique



Pause Optim !

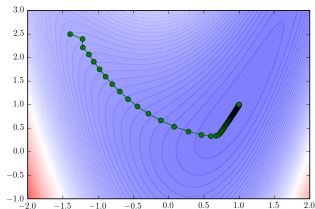
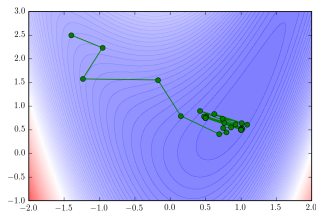
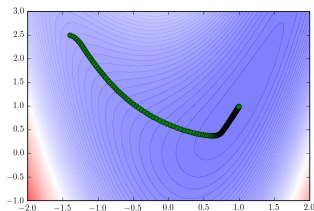
- Quand  $\eta$  est choisi trop grand :



# 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  $\eta$  ?



Minimisation de la NLL par descente de gradient : principe graphique



Pause Optim !

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

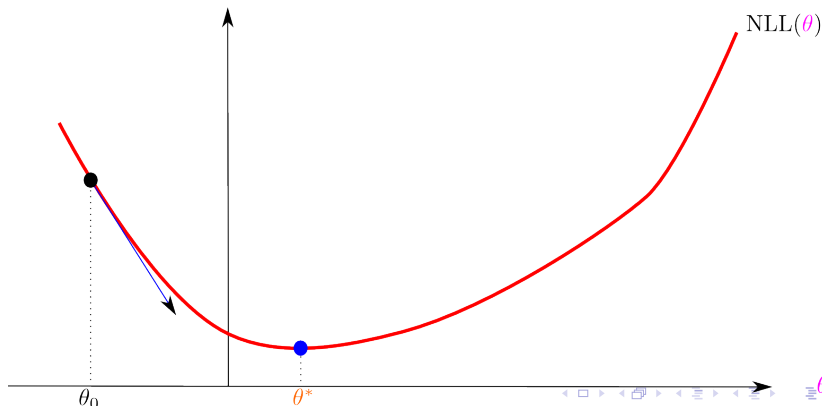
$$\eta_t = \frac{\eta}{\frac{d^2}{d\theta^2} \text{NLL}(\theta_t)} \quad (17)$$

Minimisation de la NLL par descente de gradient : principe graphique



Pause Optim !

$$\eta_t = \eta \times \left( \frac{d^2}{d\theta^2} \text{NLL}(\theta_t) \right)^{-1} \quad (18)$$



Minimisation de la NLL par descente de gradient : principe graphique



Pause Optim !

- Cette façon de calculer  $\eta$  est appelée méthode de Newton.

Minimisation de la NLL par descente de gradient : principe graphique



Pause Optim !

- Cette façon de calculer  $\eta$  est appelée méthode de Newton.
- 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  $\eta$  est appelée méthode de Newton.
- Elle peut converger vers un max si la fonction n'est pas convexe.
- Le calcul des dérivées secondes peut être couteux.

Minimisation de la NLL par descente de gradient : principe graphique



Pause Optim !

- Cette façon de calculer  $\eta$  est appelée méthode de Newton.
- Elle peut converger vers un max si la fonction n'est pas convexe.
- Le calcul des dérivées secondes peut être couteux.
- Ca peut déraiper 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.

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.
- 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

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

On montre alors que

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

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

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  $\mathcal{D}$  :

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

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  $\mathcal{D}$  :

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

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

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

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

$$\frac{d}{d\theta} \text{NLL}(\theta) = \mathbf{X}.\text{err} . \quad (24)$$

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

$$\frac{d}{d\theta} \text{NLL}(\theta) = \mathbf{X} \cdot \mathbf{err} . \quad (24)$$

Pour les **dérivées** 2ndes, on a :

$$\frac{d^2}{d\theta^2} \text{NLL}(\theta) = \mathbf{X} \cdot \mathbf{S} \cdot \mathbf{X}^T , \quad (25)$$

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

$$\frac{d}{d\theta} \text{NLL}(\theta) = \mathbf{X} \cdot \mathbf{err} . \quad (24)$$

Pour les **dérivées** 2ndes, on a :

$$\frac{d^2}{d\theta^2} \text{NLL}(\theta) = \mathbf{X} \cdot \mathbf{S} \cdot \mathbf{X}^T , \quad (25)$$

avec

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

Régression logistique : **minimisation** la NLL par la méthode de Newton

Bilan sous forme **matricielle** : la mäj se fait selon

$$\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \left(\mathbf{X}\mathbf{S}\mathbf{X}^T\right)^{-1} \mathbf{X}.\text{err} . \quad (27)$$



Régression logistique : **minimisation** la NLL par la méthode de Newton

Bilan sous forme **matricielle** : la màj se fait selon

$$\boldsymbol{\theta}_{t+1} \longleftarrow \boldsymbol{\theta}_t - \left(\mathbf{X}\mathbf{S}\mathbf{X}^T\right)^{-1} \mathbf{X}.\text{err} . \quad (27)$$

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 :  $\#\mathcal{C} = \ell \geq 3$ .

## Régression logistique : extension multi-classes

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

**Exemple** : autorisation de carte bancaire

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

## Régression logistique : extension multi-classes

- Supposons à présent un problème à plus de 2 classes :  $\#C = \ell \geq 3$ .

Exemple : autorisation de carte bancaire

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

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

## Régression logistique : extension multi-classes

- Supposons à présent un problème à plus de 2 classes :  $\#C = \ell \geq 3$ .

**Exemple** : autorisation de carte bancaire

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

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

$$\boldsymbol{\pi}(\mathbf{x}) = \begin{bmatrix} \text{score}_1(\mathbf{x}) \\ \vdots \\ \text{score}_\ell(\mathbf{x}) \end{bmatrix} \quad (28)$$

## Régression logistique : extension multi-classes

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

**Exemple** : autorisation de carte bancaire

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

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

$$\boldsymbol{\pi}(\mathbf{x}) = \begin{bmatrix} \text{score}_1(\mathbf{x}) \\ \vdots \\ \text{score}_\ell(\mathbf{x}) \end{bmatrix} \quad (28)$$

et

$$\text{score}_i(\mathbf{x}) = f(\mathbf{w}_i^T \cdot \mathbf{x} + b_i). \quad (29)$$

## Régression logistique : extension multi-classes

- Quelle fonction  $f$  peut envoyer les combinaisons  $\mathbf{w}_i^T \cdot \mathbf{x} + b_i$  vers un vecteur  $\boldsymbol{\pi}$  contenant des probabilités ?

## Régression logistique : extension multi-classes

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

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



## Régression logistique : extension multi-classes

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

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

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

## Régression logistique : extension multi-classes

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

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

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

## Régression logistique : extension multi-classes

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

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

- $\Theta$  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 :

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

## Régression logistique : extension multi-classes

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

$$\mathbf{w}_1^T \cdot \mathbf{x} + b_1 = 0.1 \quad (31)$$

$$\mathbf{w}_2^T \cdot \mathbf{x} + b_2 = 0.1 \quad (32)$$

$$\mathbf{w}_3^T \cdot \mathbf{x} + b_3 = 100 \quad (33)$$

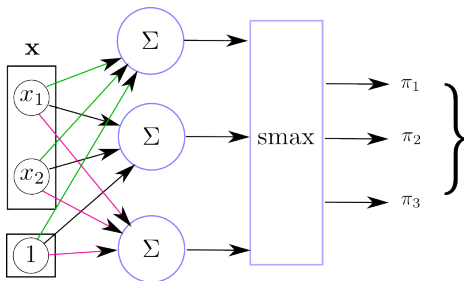
- On aurait alors :

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

- 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

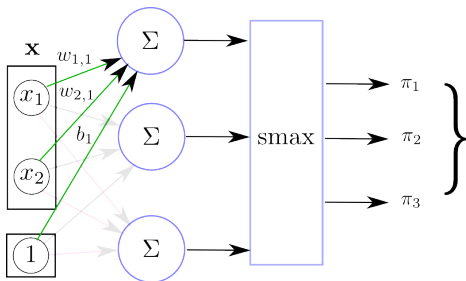


## Régression logistique : extension multi-classes

- Qu'advient-il des paramètres ?

## Régression logistique : extension multi-classes

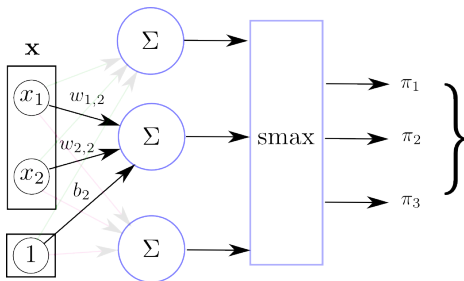
- Qu'advient-il des paramètres ?



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

## Régression logistique : extension multi-classes

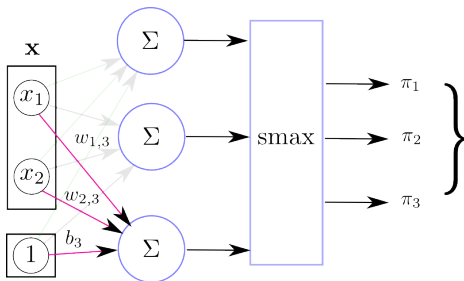
- Qu'advient-il des paramètres ?



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

## Régression logistique : extension multi-classes

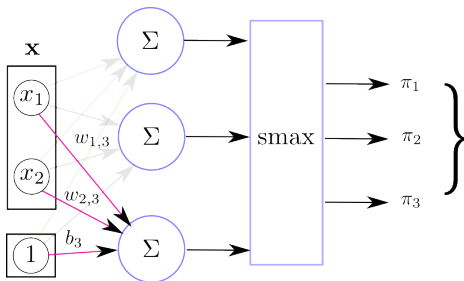
- Qu'advient-il des paramètres ?



$f_{\Theta}(\mathbf{x})$   
 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

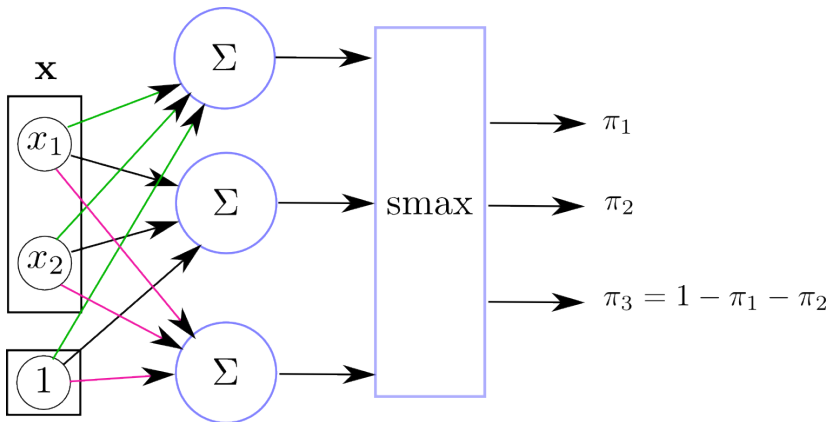
Donc  $\Theta =$

## Régression logistique : extension multi-classes

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

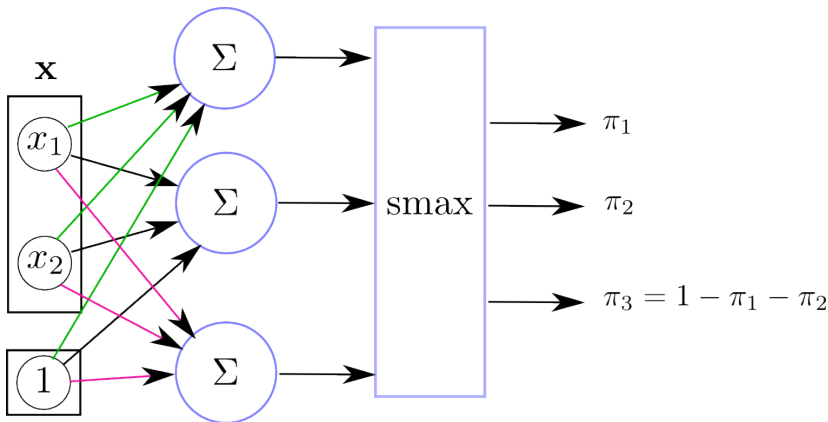
## Régression logistique : extension multi-classes

- 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* se servait à rien !



## Régression logistique : extension multi-classes

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

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

## Régression logistique : extension multi-classes

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

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

- Par convention, on choisit  $\psi = \theta_\ell$  et on supprime la dernière colonne de  $\Theta$  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 séparément pour chaque vecteur  $\theta_i$  selon :

$$\frac{\partial}{\partial \theta_i} \text{NLL}(\Theta) = - \sum_{j=1}^n \mathbf{x}^{(j)} \left( \mathbb{I}_i(y^{(j)}) - \pi_i \right) \quad (36)$$

# Plan du chapitre

- 1 Origines du modèle
- 2 Descente de gradient
- 3 Descente de gradient stochastique**
- 4 Conclusions

# 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 :

$$\frac{d}{d\theta} \text{NLL}(\theta) = \mathbf{X}.\text{err} . \quad (37)$$

## 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 :

$$\frac{d}{d\theta} \text{NLL}(\theta) = \mathbf{X}.\text{err} . \quad (37)$$

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

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

## Régression logistique et Big Data

- Reformulons ce calcul matriciel :

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

## Régression logistique et Big Data

- Reformulons ce calcul matriciel :

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

- 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 :

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

- 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  $\theta$  ?

## Régression logistique et Big Data

- Reformulons ce calcul matriciel :

$$\frac{d}{d\theta} \text{NLL}(\theta) = \mathbf{X}.\text{err} = \sum_{i=1}^n \text{err}_i \times \mathbf{x}^{(i)}. \quad (39)$$

- 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  $\theta$  ?
- Peut on faire la m à j exemple par exemple selon :

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \times \text{err}_i \times \mathbf{x}^{(i)} ? \quad (40)$$

## Régression logistique et Big Data

- Reformulons ce calcul matriciel :

$$\frac{d}{d\theta} \text{NLL}(\theta) = \mathbf{X}.\text{err} = \sum_{i=1}^n \text{err}_i \times \mathbf{x}^{(i)}. \quad (39)$$

- 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  $\theta$  ?
- Peut on faire la m à j exemple par exemple selon :

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \times \text{err}_i \times \mathbf{x}^{(i)} ? \quad (40)$$

→ 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

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

## 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

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

- Prenons l'espérance sous la loi de  $U$  :

$$\mathbb{E}_U[g] = \sum_{i=1}^n err_u \times \mathbf{x}^{(u)} p_U(u), \quad (42)$$

$$= \mathbf{x}^{(u)} \quad (43)$$

## 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

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

- Prenons l'espérance sous la loi de  $U$  :

$$\mathbb{E}_U[g] = \sum_{i=1}^n err_u \times \mathbf{x}^{(u)} p_U(u), \quad (42)$$

$$= \mathbf{x}^{(u)} \quad (43)$$

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

## Régression logistique et Big Data : gradient stochastique

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

## Régression logistique et Big Data : gradient stochastique

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

$$\mathbb{E}[\text{màj}] = \text{cible} \quad (44)$$

$$\Leftrightarrow \mathbb{E}_{\mathbf{u}} \left[ \text{gradient pour } \mathbf{x}^{(\mathbf{u})} \text{ seul} \right] = \frac{d}{d\theta} \text{NLL}(\theta). \quad (45)$$

## Régression logistique et Big Data : gradient stochastique

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

$$\mathbb{E}[\text{màj}] = \text{cible} \quad (44)$$

$$\Leftrightarrow \mathbb{E}_{\mathbf{x}^{(U)}} [\text{gradient pour } \mathbf{x}^{(U)} \text{ seul}] = \frac{d}{d\theta} \text{NLL}(\theta). \quad (45)$$

- Je peux alors invoquer le résultat suivant :

### Conditions de Robbins-Monroe

Si la suite  $\eta_t$  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  $\theta^*$ .

## Régression logistique et Big Data : gradient stochastique / Procédure

## SGD

Initialiser  $\theta_0$  et  $\eta_0$ .

**while** pas convergé **do**

Mélanger les données aléatoirement (*Shuffle*).

**for**  $i$  de 1 à  $n$  **do**

Calculer le gradient sur l'exemple  $\mathbf{x}^{(i)}$  :  $\mathbf{g} \leftarrow err_i \times \mathbf{x}^{(i)}$ .

Mettre à jour les paramètres :  $\theta_{t+1} \leftarrow \theta_t - \eta_t \times \mathbf{g}$ .

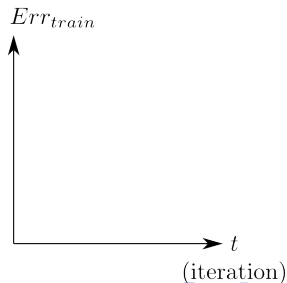
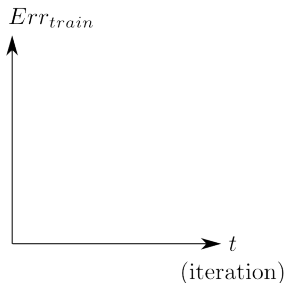
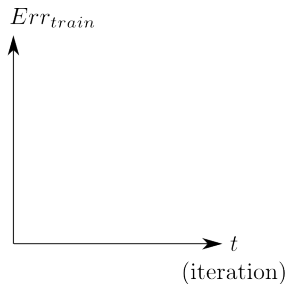
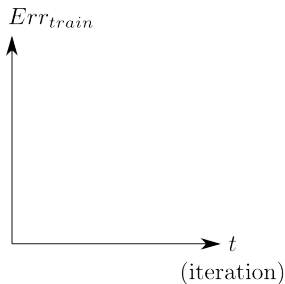
Mettre à jour le *learning rate* :  $\eta_{t+1} \leftarrow \frac{a}{t+b}$

**end for**

**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  $\theta$  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**.