

ARF

Régression linéaire Régression logistique Descente de gradient

Cours 2
ARF Master DAC

Nicolas Baskiotis

`nicolas.baskiotis@lip6.fr`

`http://webia.lip6.fr/~baskiotis`

équipe MLIA, Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie (UPMC)

S2 (2016-2017)

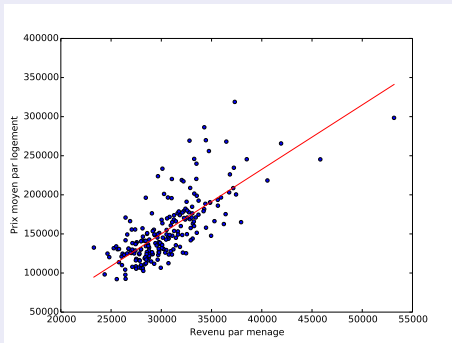
Plan

- 1 Régression linéaire
- 2 Régression logistique
- 3 Descente de gradient

Introduction

Régression linéaire

- Objectif : prédire une sortie continue réelle y à partir d'un nombre de variables d'entrée
- beaucoup d'applications, très utilisée un peu dans tous les domaines
- très flexible (transformation des entrées)



Formalisation

Objectif

Etant donné un ensemble $\{(\mathbf{x}^i, y^i)\} \in \mathbb{R}^d \times \mathbb{R}$,

- Hypothèse : variation linéaire de la sortie en fonction des entrées

$$\mathbb{E}(y|\mathbf{x}) = w_0 + \sum_{i=1}^d w_i x_i$$

⇒ On cherche :

- ▶ une fonction $f_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{i=1}^d w_i x_i$
- ▶ qui fait le moins d'erreurs : $f(\mathbf{x}^i)$ doit être proche de y^i
- ▶ sous la condition que l'erreur est indépendante de x , de variance σ^2 constante, suit une loi normale.

⇒ $y|x \sim \mathcal{N}(f(x), \sigma^2)$ (lien avec l'apprentissage bayésien)

- Notion d'erreur quadratique : $\ell(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$

Formalisation (2)

Objectif

- Minimiser :

$$\mathbb{E}(\ell(f(\mathbf{x}), y)) = \int_{\mathbf{x}, y} (y - f(\mathbf{x}))^2 p(\mathbf{x}, y) dx dy$$

- Soit trouver $\mathbf{w} \in \mathbb{R}^{d+1}$ qui minimise :

$$\sum_{j=1}^n \ell(f_{\mathbf{w}}(\mathbf{x}), y) = \sum_{j=1}^n (y^j - f(\mathbf{x}^j))^2 = \sum_{j=1}^n (y^j - w_0 - \sum_{i=1}^d w_i x_i^j)^2$$

Quelques notations et rappels

Convexité

- C ensemble convexe de \mathbb{R}^n : $\forall x, y \in C, \forall \lambda \in [0, 1], \lambda x + (1 - \lambda)y \in C$
- $\sum_i \lambda_i x_i$ est une combinaison convexe ssi $\forall i, \lambda_i \geq 0$ et $\sum_i \lambda_i = 1$
- Enveloppe convexe d'un ensemble fini $\{x_i\}, i = 1 \dots n$: toutes les combinaisons convexes de l'ensemble
- Fonction $f : X \rightarrow \mathbb{R}$ convexe ssi

$$\forall x, x' \in X, \forall \lambda \in [0, 1] \text{ tq } \lambda x + (1 - \lambda)x' \in X$$

$$\text{alors } f(\lambda x + (1 - \lambda)x') \leq \lambda f(x) + (1 - \lambda)f(x')$$

- si $\lambda_i \geq 0$ et $\sum_i \lambda_i = 1$, alors $f(\sum_i \lambda_i x_i) \leq \sum_i \lambda_i f(x_i)$ (inégalité de Jensen)

Quelques notations et rappels

Différentiabilité

- Si $f : X \rightarrow \mathbb{R}$ est convexe ssi $\forall x, x' \in X, f(x') \geq f(x) + \langle x' - x, \nabla f(x) \rangle$
- Si f convexe, alors sa matrice hessienne est définie semi-positive : $\nabla^2 f \geq 0$.

Minimum

- Si f atteint son minimum, alors les minimums forment un ensemble convexe.
- Si l'ensemble est strictement convexe, le minimum est un singleton.
- Si f est strictement convexe, son gradient ne s'annule que à son minimum local.

Régression : solution analytique

Formalisation

- Minimiser :

$$\mathbb{E}(\ell(f(\mathbf{x}), y)) = \int_{\mathbf{x}, y} (y - f(\mathbf{x}))^2 p(\mathbf{x}, y) dx dy$$

- Soit trouver $\mathbf{w} \in \mathbb{R}^{d+1}$ qui minimise :

$$L(\mathbf{w}) = \sum_{j=1}^n \ell(f_{\mathbf{w}}(\mathbf{x}), y) = \sum_{j=1}^n (y^j - f(\mathbf{x}^j))^2 = \sum_{j=1}^n (y^j - w_0 - \sum_{i=1}^d w_i x_i^j)^2$$

- La fonction $L : \mathbb{R}^{d+1} \rightarrow \mathbb{R}$ est convexe
- ⇒ Solution analytique : annuler son gradient !
- ⇒ Trouver \mathbf{w}^* tq $\nabla_{\mathbf{w}} L(\mathbf{w}^*) = 0$

Dérivée matricielle

Ecriture pratique

- $X = \begin{pmatrix} x^1 \\ x^2 \\ \vdots \\ x^n \end{pmatrix} = \begin{pmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ & & \vdots & \\ x_1^n & x_2^n & \dots & x_d^n \end{pmatrix}, Y = \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{pmatrix}, W = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}$
- $L(\mathbf{w}) = (XW - Y)'(XW - Y)$
- $\nabla_{\mathbf{w}} L = 2X'(XW - Y), \mathbf{w} \text{ optimal} \Leftrightarrow 2X'(XW - Y) = 0$
- Solution : $(X'X)^{-1}X'Y$
- Et pour w_0 ?

Plan

- 1 Régression linéaire
- 2 Régression logistique**
- 3 Descente de gradient

Problématique

Classification binaire

- Deux classes : $Y = \{0, +1\}$, et un ensemble d'apprentissage $\{(\mathbf{x}^i, y^i) \in \mathbb{R}^d \times Y\}$
- Peut-on utiliser un coût quadratique dans ce cas ?
Cas 2D :

- ▶ $f_{\mathbf{w}}(x) = 1 \Leftrightarrow w_0 + x_1 w_1 + w_2 w_2 = 1$
- ▶ $f_{\mathbf{w}}(x) = 0 \Leftrightarrow w_0 + x_1 w_1 + w_2 w_2 = 0$

Problématique

Classification binaire

- Deux classes : $Y = \{0, +1\}$, et un ensemble d'apprentissage $\{(\mathbf{x}^i, y^i) \in \mathbb{R}^d \times Y\}$

- Peut-on utiliser un coût quadratique dans ce cas ?

Cas 2D :

$$\triangleright f_{\mathbf{w}}(x) = 1 \Leftrightarrow w_0 + x_1 w_1 + w_2 w_2 = 1$$

$$\triangleright f_{\mathbf{w}}(x) = 0 \Leftrightarrow w_0 + x_1 w_1 + w_2 w_2 = 0$$

- Et si $f_{\mathbf{w}}(x) \gg 1$?

⇒ le coût sera très grand !

- De même si $f_{\mathbf{w}}(x) \ll 0$

⇒ Le coût n'est pas adapté à la classification !

Adaptation du formalisme

Dans chaque région de l'espace :

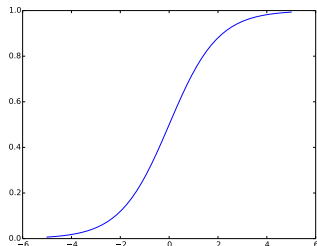
- le label +1 suit une loi de Bernoulli : $p(y = 1|\mathbf{x}) = \mu(\mathbf{x})$
 - le label 0 également : $p(y = 0|\mathbf{x}) = 1 - \mu(\mathbf{x})$
- ⇒ $p(y|x) = \mu(\mathbf{x})^y(1 - \mu(\mathbf{x}))^{1-y}$
- Comment représenter $\mu(\mathbf{x})$? Fonction linéaire ?

Adaptation du formalisme

Dans chaque région de l'espace :

- le label +1 suit une loi de Bernoulli : $p(y = 1|\mathbf{x}) = \mu(\mathbf{x})$
 - le label 0 également : $p(y = 0|\mathbf{x}) = 1 - \mu(\mathbf{x})$
- ⇒ $p(y|x) = \mu(\mathbf{x})^y (1 - \mu(\mathbf{x}))^{1-y}$
- Comment représenter $\mu(\mathbf{x})$? Fonction linéaire ?
- ⇒ **Problème ! pas entre 0 et 1 !**
- Transformation d'une fonction linéaire : la fonction sigmoïde.

$$\mu(\mathbf{x}) = \sigma(f_{\mathbf{w}}(\mathbf{x})) = \frac{1}{1 + e^{-f_{\mathbf{w}}(\mathbf{x})}}$$



Quelques remarques importantes

Que représente $f_{\mathbf{w}}(\mathbf{x})$?

- C'est le log-ratio des probabilités : $f_{\mathbf{w}}(x) = \log \left(\frac{P(+|x)}{P(-|x)} \right) = \log \left(\frac{\mu(x)}{1-\mu(x)} \right)$
 - qui est approximée par une fonction linéaire :
$$\ln \frac{P(+|x)}{P(-|x)} = w_0 + w_1 x_1 + w_2 x_2 \dots$$
 - $\mu(\mathbf{x}) = \frac{1}{1+e^{-f_{\mathbf{w}}(\mathbf{x})}}$ et $1 - \mu(\mathbf{x}) = \frac{1}{1+e^{f_{\mathbf{w}}(\mathbf{x})}}$
- $P(y|x, \mathbf{w}) = \sigma((2y - 1)f_{\mathbf{w}}(\mathbf{x}))$ (Vraisemblance du modèle pour un échantillon (\mathbf{x}, y))
- $\sigma'(x) = \sigma(x)(1 - \sigma(x))$
 - Quand est-ce que :
 - ▶ $p(+|x) = 0.5$?
 - ▶ $p(+|x) < 0.5$?
 - ▶ $p(+|x) > 0.5$?

Résolution

Maximum de vraisemblance

On cherche à maximiser :

$$P(y^1, \dots, y^n | \mathbf{x}^1, \dots, \mathbf{x}^n) = \prod_{i=1}^n P(y^i | \mathbf{x}^i)$$

$$\Leftrightarrow \text{maximiser } \log \prod_{i=1}^n P(y^i | \mathbf{x}^i)$$

$$\Leftrightarrow \text{maximiser } \sum_{i=1}^n \log P(y^i | \mathbf{x}^i)$$

$$\Leftrightarrow \text{minimiser } \frac{1}{n} \sum_{i=1}^n \log \frac{1}{P(y^i | \mathbf{x}^i)}$$

$$\Leftrightarrow \text{minimiser } \frac{1}{n} \sum_{i=1}^n \log \frac{1}{\sigma((2y^i - 1)f_{\mathbf{w}}(\mathbf{x}^i))}$$

$$\Leftrightarrow \text{On cherche } \mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^n \log [1 + \exp(-(2y^i - 1)f_{\mathbf{w}}(\mathbf{x}^i))]$$

Résolution

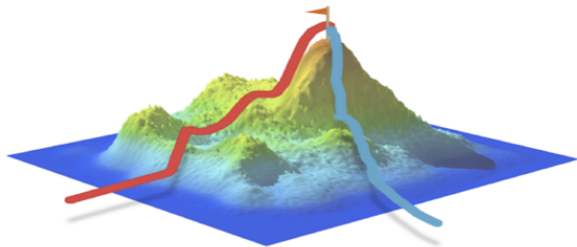
- Pas de solution analytique.
- Méthode d'optimisation numérique \rightarrow descente de gradient.

Plan

- 1 Régression linéaire
- 2 Régression logistique
- 3 Descente de gradient**

Principe

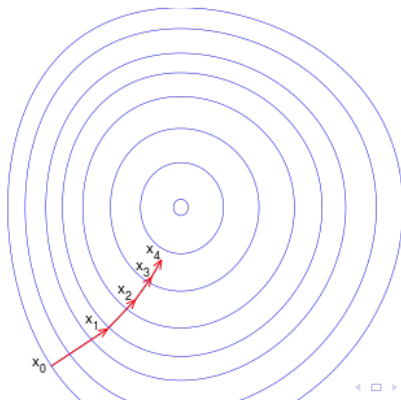
- Algorithme d'optimisation différentiable
- S'applique pour toute fonction différentiable
- Idée simple : améliorer de façon itérative la solution courante



Algorithme du gradient

Algorithme

- 1 Choisir un point x_0
- 2 Itérer :
 - ▶ Calculer $\nabla f(x_t)$
 - ▶ mettre à jour $x_{t+1} \leftarrow x_t - \alpha \nabla f(x_t)$



Pourquoi cela fonctionne ?

Développement de Taylor

- $f(\mathbf{x}) = f(\mathbf{x}_1) + \nabla f(\mathbf{x}_1) \times (\mathbf{x} - \mathbf{x}_1) + O(\|\mathbf{x} - \mathbf{x}_1\|^2)$
- On cherche à “bouger” dans une direction \mathbf{u} de façon à minimiser f :

$$f(\mathbf{x}_1 + h\mathbf{u}) - f(\mathbf{x}_1) = h\nabla f(\mathbf{x}_1)\mathbf{u} + h^2O(1)$$

- On doit donc minimiser $\nabla f(\mathbf{x}_1)\mathbf{u}$.
- On choisit le vecteur unité $\mathbf{u} = -\frac{\nabla f(\mathbf{x}_1)}{\|\nabla f(\mathbf{x}_1)\|}$

Plusieurs variantes :

- Hill climbing dans le cas discret
- Coordinate descent (line search selon les dimensions)
- Conjugate Gradient

Importance du pas de gradient

Algorithme

- 1 Choisir un point x_0
- 2 Itérer :
 - ▶ Calculer $\nabla f(x_t)$
 - ▶ mettre à jour $x_{t+1} \leftarrow x_t - \alpha \nabla f(x_t)$

Remarques

- Que se passe-t-il si :
 - ▶ α est choisi trop grand ?
 - ▶ trop petit ?
- Est-ce que l'on atteint toujours un minimum global ?
- Application à la régression logistique ?