Chapitre 3 : Régression Linéaire

Ibrahima Sy

Institut Supérieur Informatique (**ISI**)
Master 2 Informatique Appliquée à la Gestion des Entreprises (**IAGE**)

August 12, 2021

Plan

Apprentissage supervisé, classification, régression

- L'apprentissage supervisé est lorsqu'on a une cible à prédire
  - **classification** : la cible est un indice de classe $t \in \{1, 2, ...K\}$
    - exemple : reconnaissance de caractères
    - $\mathbf{x}$: vecteur des intensités de tous les pixels de l'image
    - t : identité du caractère
  - **régression** : la cible est un nombre réel $t \in \mathbb{R}$
    - exemple : prédiction de la valeur d'une action à la bourse
    - $\mathbf{x}$: vecteur contenant l'information sur l'activité économique de la journée
    - t : valeur d'une action à la bourse le lendemain

Modèle , biais, poids

Le modèle de **régression linéaire** est de la forme :

$$\begin{aligned} y_{\mathbf{w}}(\mathbf{x}, \mathbf{w}) &= \omega_0 + \omega_1 x_1 + \cdots + \omega_D x_D \\ &= \mathbf{w}^T \mathbf{x}' \end{aligned}$$

ou $\mathbf{x} = (x_1, \ldots, x_D)$

La prédiction correspond donc à :

- Une **droite** pour $d = 1$
- Un **plan** pour $d = 2$
- Un **hyperplan** pour $D > 2$

Régression linéaire 1 D

$$y_{\mathbf{w}}(\mathbf{x}, \mathbf{w}) = \omega_0 + \omega_1 x_1$$

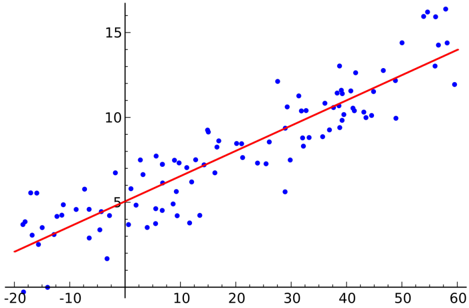


Figure 1: Régression en 1 D

Régression linéaire 2 D

$$y_{\mathbf{w}}(\mathbf{x}, \mathbf{w}) = \omega_0 + \omega_1 x_1 + \omega_2 x_2$$

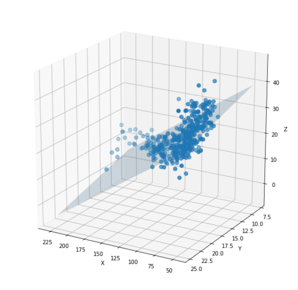
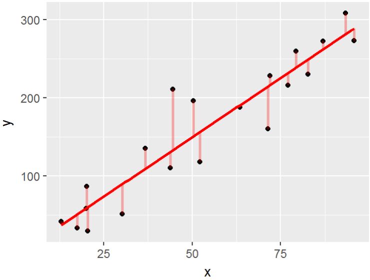


Figure 2: Régression en 2 D

- **Données** : $\mathcal{D} = \left\{ (\mathbf{x_1}, t_1), (\mathbf{x_2}, t_2), \ldots, (\mathbf{x_N}, t_N) \right\}$
- **Idéale** : Trouver un modéle $y_{\mathbf{w}}$ telque : $y_{\mathbf{w}}(\mathbf{w}, x_n) = t_n$
- **Objectif** : Comme on ne peut avoir un modèle qui prédit exactement la cible maintenant l'objectif est de trouver un modèle qui fait le moins d'erreurs possible.
- **Objectif** : $\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^{N} (y_{\mathbf{w}}(\mathbf{w}, x_n) - t_n)^2$



Maximum de vraisemblance

- Pour entraîner le modèle $y_{\mathbf{w}}(\mathbf{x}, \mathbf{w})$ nous passerons par une formulation probabiliste :

$$p(t|\mathbf{x}, \mathbf{w}, \beta^{-1}) = \mathcal{N}(t|y(\mathbf{x}, \omega), \beta^{-1})$$

- Équivaut à supposer que les cibles sont des versions bruitées du vrai modèle:

$$t_n = y_{\mathbf{w}}(\mathbf{x}, \omega) + \epsilon_n$$

ϵ : Gaussienne de moyenne 0 et de variance β^{-1}

Maximum de vraisemblance

- Soit notre ensemble d'entraînement $\mathcal{D} = \left\{ (\mathbf{x_1}, t_1), (\mathbf{x_2}, t_2), \ldots, (\mathbf{x_N}, t_N) \right\}$
- On peut aussi noter $\mathbf{X} = \{x_1, \ldots, x_N\}$ et $\mathbf{T} = \{t_1, \ldots, t_N\}^T$

- En faisant l'hypothèse i.i.d on a :

$$p(\mathbf{T}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^T\mathbf{x}_n, \beta^{-1})$$

Maximum de vraisemblance

- Lors de l'entraînement on cherche le $\mathbf{w}$ maximisant la (log-)probabilité des données d'entraînement

ou

$$\mathbb{E}_D = \frac{1}{2}\sum_{1}^{N}\{t_n - \mathbf{w}^T\mathbf{x}_n\}^2$$

Maximum de vraisemblance

Le « meilleur » **w** est celui pour lequel le **gradient est nul**:

$$\begin{aligned}\nabla \mathbb{E}_D &= \textstyle\sum_{n=1}^{N} \{t_n - \mathbf{w}^T \mathbf{x}_n\} \mathbf{x}_n^T \\ &= \textstyle\sum_{n=1}^{N} t_n \mathbf{x}_n^T - \mathbf{w}^T (\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T)\end{aligned}$$

$$\nabla \mathbb{E}_D = 0 \Longleftrightarrow \sum_{n=1}^{N} t_n \mathbf{x}_n^T - \mathbf{w}^T (\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^T) = 0$$

Maximum de vraisemblance

En isolant **w** , on obtient que:

$$\mathbf{W}_{MV} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{T}$$

$$X = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{1,d} \\ 1 & x_{2,1} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & \cdots & x_{N,d} \end{pmatrix} \qquad T = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

Régularisation, weight decay, régression de Ridge

Pour éviter l'overfitting on ajoute un terme de régulation :

$$\mathbb{E}_D = \frac{1}{2}\sum_{1}^{N}\{t_n - \mathbf{w}^T\mathbf{x}_n\}^2 + \frac{\lambda\mathbf{w}^T\mathbf{w}}{2}$$

- équivaut au maximum a posteriori dans la formulation probabiliste
- le terme de régularisation est souvent appelé **weight decay**
- la régression avec un terme de régularisation est aussi appelée **régression de Ridge**

Régularisation, weight decay, régression de Ridge

- On peut montrer que la solution (maximum a posteriori) est alors :

$$\mathbf{W}_{MAP} = (\lambda \mathbb{I} + \mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{T}$$

- dans le cas $\lambda = 0$, on retrouve la solution du maximum de vraisemblance
- si $\lambda > 0$, permet également d'avoir une solution plus stable numériquement(Si $\mathbf{X}^T\mathbf{X}$ n'est pas inversible)

Limites Régression Linéaire

- Un modèle linéaire est souvent pas assez flexible pour bien représenter les données

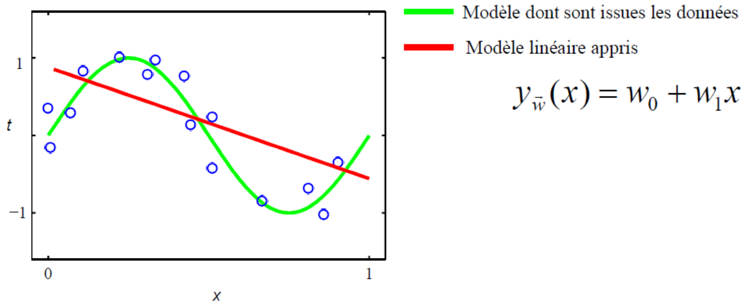


Figure 3: exemple de sous apprentissage avec LR

Solution

- on va projeter les donnée dans un espace plus grand, là où les données sont distribuées linéairement
- Ensuite on fait une régression sur des données de dimensions M au lieu de d dimensions$(M > d)$

$$\phi : \mathbb{R}^d \longrightarrow \mathbb{R}^M$$

Fonction de Base

- On transforme les entrées en définissant des fonctions de base (*Basis Function*)

$$y_{\mathbf{w}}(\mathbf{x}, \mathbf{w}) = \omega_0 + \sum_{n=1}^{M} \mathbf{w}_n \phi_i(\mathbf{x}_n)$$

- cas particulier(Régression linéaire) : $\phi_i(\mathbf{x}) = x_i$ et $M = d + 1$

Fontion de Base

Pour simplifier la notation, on va supposer que $\phi_0(\mathbf{x}) = 1$ afin d'inclure le biais dans la sommation

$$y_{\mathbf{w}}(\mathbf{x}, \mathbf{w}) = \sum_{n=0}^{M-1} \mathbf{w}_n \phi_i(\mathbf{x}_n)$$
$$= \mathbf{W}^T \phi(\mathbf{x})$$

- Avec $\mathbf{W} = (\omega_0, \ldots, \omega_{M-1})$
- Avec $\phi = (\phi_0(\mathbf{x}), \ldots, \phi_{M-1(\mathbf{x})})$
- Une des fonctions de base les plus fréquentes est la fonction polynomiale $\phi_i(\mathbf{x}) = x^i$

Formulation Probabiliste

Comme auparavant, on suppose que les données sont corrompues par un bruit gaussien

- **Estimateur du Maximum de vraisemblance**

$$\mathbf{W}_{MV} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{T}$$

- **Maximum a posteriori (MAP)**

$$\mathbf{W}_{MAP} = (\lambda\mathbb{I} + \mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{T}$$

Formulation Probabiliste

$$\Phi = \begin{pmatrix} \phi_0(\vec{x}_1) & \phi_1(\vec{x}_1) & \cdots & \phi_{M-1}(\vec{x}_1) \\ \phi_0(\vec{x}_2) & \phi_1(\vec{x}_2) & \cdots & \phi_{M-1}(\vec{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\vec{x}_N) & \phi_1(\vec{x}_N) & \cdots & \phi_{M-1}(\vec{x}_N) \end{pmatrix} \qquad T = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}$$

- $\mathbf{\Phi}$: s'appelle **design matrix**

References I

- **Hugo Larochelle**, Professeur associé, Université de Montréal, Google
- **Pierre-Marc Jodoin**, Professeur titulaire Université Sherbrooke
- Bayesian Reasoning and Machine Learning de David Barber
- The Elements of Statistical Learning de Trevor Hastie,
- Robert Tibshirani et Jerome Friedman
- Information Theory, Inference, and Learning Algorithms de David J.C. MacKay
- Convex Optimization de Stephen Boyd et Lieven Vandenberghe
- Natural Image Statistics de Aapo Hyvärinen, Jarmo Hurri et Patrik O. Hoyer
- The Quest for Artificial Intelligence - A History of Ideas and Achievements de Nils J. Nilsson
- Gaussian Processes for Machine Learning de Carl Edward Rasmussen et Christopher K. I. Williams
- Introduction to Information Retrieval de Christopher D. Manning, Prabhakar Raghavan et Hinrich Schütze