Modélisation probabiliste

1 Les limites de la modélisation avec une fonction cible

Nous présentons des exemples qui nous montrent les limitations des techniques d'apprentissage "déterministe" que l'on a vu précédemment.

1.1 Densité discrète ou continue

La densité (ou vraisemblance) d'une variable Y est une fonction $L(y)$ qui vérifie :

- Quand $Y \in \mathbb{R}^p$ est continue :

$$\forall \varphi \qquad \mathbf{E}[\varphi(Y)] = \int_{\mathbb{R}^p} \varphi(y) L(y)\, dy$$

  Que l'on peut aussi écrire

$$\mathbf{P}[Y \in dy] = L(y)\, dy$$

- Quand $Y \in E$ est discret :

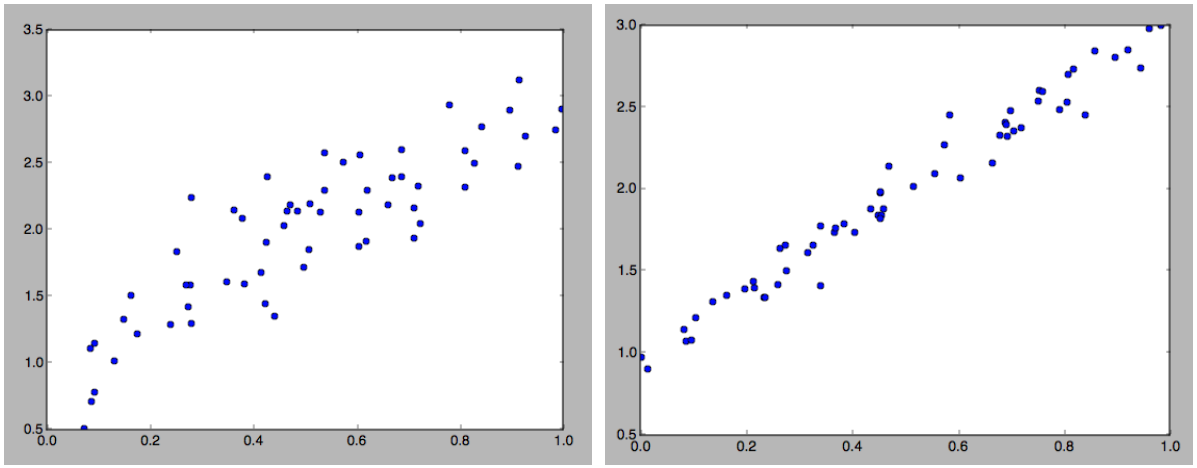
$$\forall \varphi \qquad \mathbf{E}[\varphi(Y)] = \sum_{y \in E} \varphi(y) L(y)$$

  Que l'on peut aussi écrire

$$\mathbf{P}[Y = y] = L(y)$$

1.2 Variation sur le bruit

Considérons ces deux jeux de donnés. Les X_i étant en abscisse, et les Y_i en ordonnées.



Dans les deux cas, il existe certainement un lien du type $Y = f^?(X) + Bruit$, et sans doute $f^?$ est une fonction affine. Ainsi, on choisit comme modèle :

$$f_w(x) = w_0 + w_1 x$$

Les techniques du cours précédent nous permettraient de trouver le meilleurs couple $(\widehat{w}_0, \widehat{w}_1)$. A vu d'oeil, ce couple serait le même pour deux jeux de données ci-dessus. Cependant, on voit qu'il y a une différence : le premier jeu de donnée est plus bruité que le second.

Il est souvent intéressant de mesurer la variance du bruit ; cela quantifie la qualité des observations.

Pour cela, au lieu d'estimer un lien déterministe entre X et Y, on va plutôt estimer la loi de Y sachant $X = x$. Par exemple, on peut parier que :

$$Y = w_0 + w_1 X + \sigma\varepsilon \qquad \text{avec } \varepsilon \sim \mathcal{N}(0,1) \text{ et } \sigma > 0$$

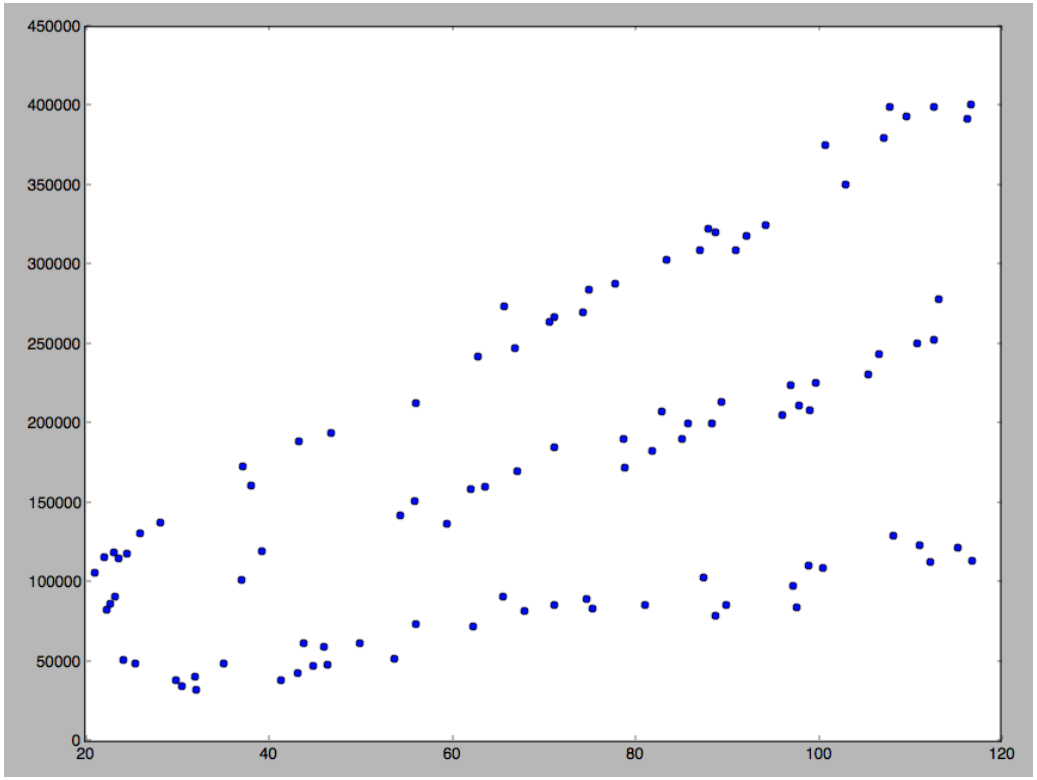
Autrement dit, la densité de Y sachant $X = x$ est donnée par

$$L_{w,\sigma}(y|x) = G_\sigma\Big(y - w_0 - w_1 x\Big) \qquad \text{avec } G_\sigma(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}$$

Il reste à trouver les meilleurs w et σ. À suivre.

1.3 Variable caché

Considérons (X, Y), avec X surface d'un appartement et Y prix d'un appartement. Nos observations sont les suivantes :



Question : Qu'est-ce qui pourrait expliquer de telles variations sur les prix ? L'appartenance à un quartier bien sûr !

Pour de telles données, on ne peut pas imaginer de fonction $f^?$ telle que $Y = f^?(X) + Bruit$. On pourrait par contre imaginer une fonction telle que $Y = f^?(X, Q) + Bruit$ où Q est le quartier, mais on ne nous nous a pas donnée Q (on parle de variable cachée).

Donc avec les données dont on dispose, peut modéliser Y ainsi :

$$Y = \begin{cases} w_0^0 + w_1^0 X + \sigma\varepsilon & \text{avec proba } \frac{1}{3} \\ w_0^1 + w_1^1 X + \sigma\varepsilon & \text{avec proba } \frac{1}{3} \\ w_0^2 + w_1^2 X + \sigma\varepsilon & \text{avec proba } \frac{1}{3} \end{cases}$$

Ainsi la densité de Y sachant $X = x$ serait :

$$L_{w,\sigma}(y|x) = \frac{1}{3}G_\sigma(y - w_0^0 - w_1^0 x) + \frac{1}{3}G_\sigma(y - w_0^1 - w_1^1 x) + \frac{1}{3}G_\sigma(y - w_0^2 - w_1^2 x)$$

Les paramètre inconnus sont ici σ et la matrice w de taille $[2, 3]$.

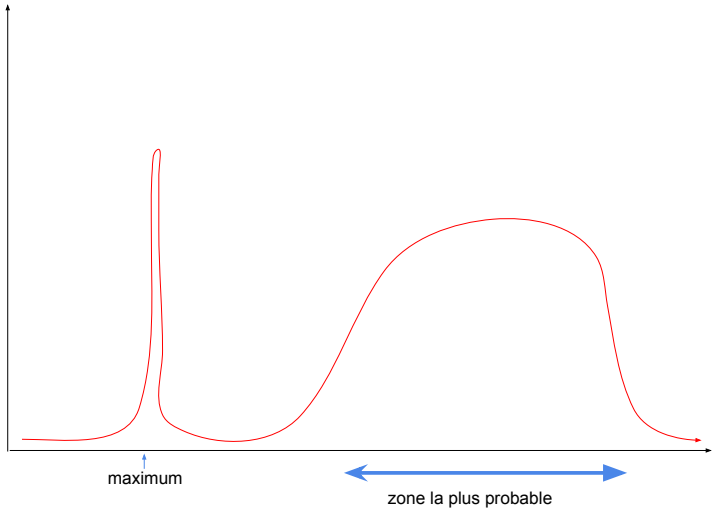
Remarques : puisqu'on ne nous a pas fournis une variable visiblement très importante (le quartier des appartements), on a mis cette variable cachée en paramètre inconnu (c'est l'exposant dans le tenseur w). On demande à l'algorithme d'optimisation de faire au mieux avec ces inconnues supplémentaires : le meilleur couple de paramétre sera appelé $\widehat{w}, \widehat{\sigma}$, et la densité estimée sera $\widehat{L} = L_{\widehat{w},\widehat{\sigma}}$.

Bien entendu, on pourrait augmenter la flexibilité du modèle en supposons que le nombre d'appartements par quartier n'est pas le même, ou en supposant que le bruit par quartier n'est pas le même. Mais il faut que cela soit vraiment nécessaire, car lorsqu'on a trop d'inconnue ...

1.4 Et si on veut quand même faire de la prédiction

Ainsi, notre nouvel objectif est de trouver un $\widehat{L}(y|x)$ qui décrive au mieux la loi de Y sachant $X = x$. Ensuite, on peut poser $\widehat{f}(x) = \text{argmax}_y \widehat{L}(y|x)$.

Cette technique repose sur l'hypothèse que l'endroit le plus probable d'apparition d'une v.a., c'est l'argmax de sa densité. Mais ce n'est pas vrai en général :



Heureusement, dans la nature, les densités ont des formes bien plus sages (ex: des gaussiennes).

Autre manière d'estimer: on peut prendre l'espérance:

$$\widehat{f}(x) = \int y \, \widehat{L}(y|x)$$

2 Distance cross-entropique

2.1 Trouver la meilleur densité via une distance

Notre problème est donc de trouver une densité conditionnelle ou vraisemblance $\hat{L}(y|x)$ qui représente au mieux la distribution de Y sachant $X = x$. On va rechercher cette densité dans une famille paramétrique $\mathcal{L} = \{L_\theta : \theta \in \Theta\}$.

Pour trouver le meilleur élément de cette famille on peut prendre:

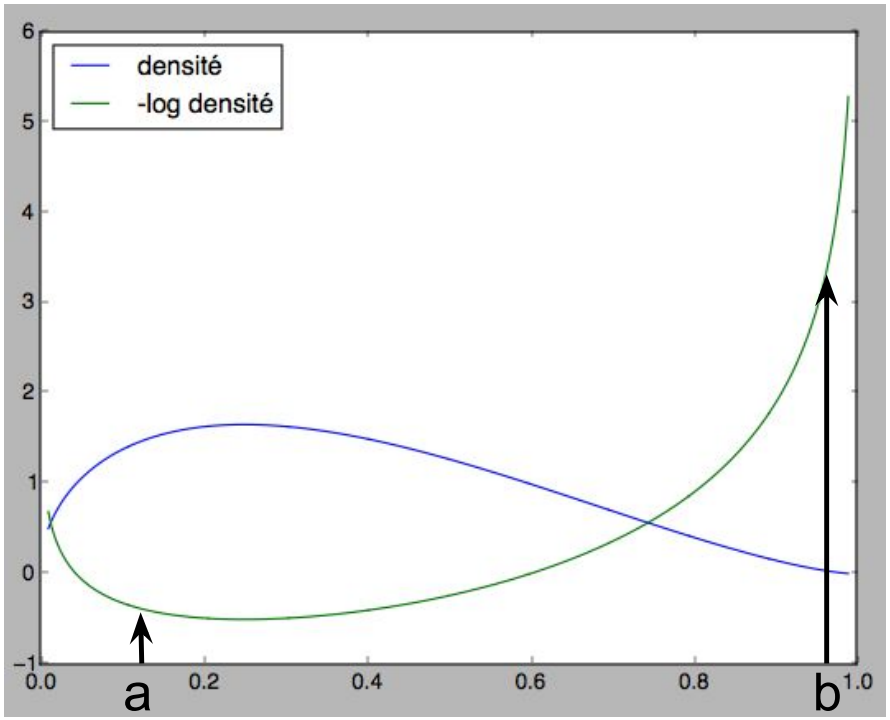
$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{Train} \operatorname{dist}\Big(L_\theta(\cdot|x_i), y_i\Big)$$

puis $\hat{L} = L_{\widehat{\theta}}$.

Attention : dist est une distance entre une densité L et une observation y_i. La distance sera petite quand la densité charge beaucoup l'observation.

La distance la plus connue, pour le cas discret et continu est

$$\operatorname{dist}\Big(L, y\Big) = H\Big(y, L\Big) = -\ln L(y)$$



$H(a, L)$ est petite alors que $H(b, L)$ est grande.

2.2 Cas discret

Dans le cas discret, quand p et q sont des probas, on écrit généralement

$$H(p, q) = -\sum_v p(v) \ln q(v)$$

Ainsi en notant δ_y la Dirac en y:

$$H(y, L) = H(\delta_y, L) = -\sum_v \delta_y(v) \ln L(v)$$

Attention, il y a deux notions proche: la divergence de Kullback–Leibler:

$$D_{KL}(p||q) = +\sum_v p(v) \ln \frac{q(v)}{p(v)}$$

et l'entropie:

$$H(p) = +\sum_v p(v) \ln p(v)$$

qui sont reliés par

$$H(p,q) = H(p) + D_{KL}(p||q)$$

2.3 Construire $\widehat{L}$ par maximum de vraisemblance

L'indépendance des observations, nous indique que la densité d'un échantillon $(Y_1, ..., Y_n)$ sachant $X_1 = x_1, ..., X_n = x_n$ est :

$$(y_1, ..., y_n) \to \prod_{Train} L_\theta(y_i|x_i)$$

Ainsi, quand on observe $Train = [(y_1, x_1), ..., (y_n, x_n)]$, le paramètre θ qui rend ces observations les plus vraisemblables est :

$$\widehat{\theta} = \underset{\theta \in \Theta}{\text{argmax}} \prod_{Train} L_\theta(y_i|x_i)$$

On doit estimer l'argmax d'un produit. Comme on préfère les sommes, on passe le tout au logarithme. C'est une fonction croissante, donc :

$$\widehat{\theta} = \underset{\theta \in \Theta}{\text{argmax}} \sum_{Train} \ln L_\theta(y_i|x_i)$$

En mettant un signe moins devant :

$$\widehat{\theta} = \operatorname*{argmin}_{\theta\in\Theta} \sum_{Train} -\ln L_\theta(y_i|x_i) = \operatorname*{argmin}_{\theta\in\Theta} \sum_{Train} H\big(y_i, L_\theta(\cdot|x_i)\big)$$

Conclusion: maximiser la vraissemblance ou minimiser la distance cross-entropique, c'est idem.

2.4 Exercice

Considérez le modèle linéaire gaussien avec le paramètre $\sigma = 1$:

$$L_w(y|x) = G\Big(u - w_0 - w_1 x\Big) \qquad \text{avec } G(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}$$

Ecrire le problème de minimisation donné par le maximum de vraissemblance. Que constatez-vous ?

Plus dur: Refaites le même exo quand σ est inconnu :

$$L_{w,\sigma}(y|x) = G_\sigma\Big(u - w_0 - w_1 x\Big) \qquad \text{avec } G_\sigma(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}$$

Le problème de minimisation obtenu admet une solution explicite $(\widehat{w}, \widehat{\sigma})$, que l'on calcul en cherchant le lieu où la différentielle s'annule. Pour $\widehat{w}$ on tombe sur l'estimateur déjà vu, pour $\widehat{\sigma}$ on tombe sur une formule naturelle qui est implémentée dans la classe du TP. Si on n'aime pas calculer la différentielle, on peut aussi demander un algo d'optimisation de trouver $(\widehat{w}, \widehat{\sigma})$.

3 Classification

3.1 construction générique

On se spécialise maintenant dans le cas où l'output Y est quantitatif. Pour simplifier on appellera les classes $\{1, 2, ..., k\}$.

Une densité L sur $\{1, 2, ..., k\}$ est assimilée au vecteur $[L(1), ..., L(k)] \in \mathbb{R}^k$, vecteur positif dont la somme fait 1.

On considère les variables explicatives $X = (X^1, ..., X^p)$.

Définissons la fonction softmax:

$$\Big(\mathtt{SM}(V)\Big)_y = \frac{e^{V_y}}{\sum_u e^{V_u}}$$

qui transforme tout vecteur V en un vecteur de proba.

Pour construire un modèle paramètrique, commence par se donner une famille de fonction f_θ de $\mathbb{R}^p$ dans $\mathbb{R}^k$. Puis on définit:

$$L_\theta(y|x) = \Big(\mathtt{SM} \circ f_\theta(x)\Big)_y$$

La quasi-totalité des modèles de classification sont de la forme

Exo: caculer numériquement

$$\mathtt{SM}(1, -3, 10, 1)$$

$$(0, 0, 10, 0)$$

Vocabulaire: le résultat du modèle avant le softmax: $f_\theta(x)$ est souvent appelé les "logits".

3.2 Régression logistique (ou softmax)

C'est le modèle de classification le plus simple: le paramètre est $\theta = (w, b)$ avec w une matrice $p \times k$ et b un vecteur de taille k et on choisit:

$$f_{w,b}(x) = w \cdot x + b$$

puis

$$L_{w,b}(y|x) = \Big(\mathtt{SM} \circ f_{w,b}(x)\Big)_y$$

3.3 Ex: Réseau de neurone

Donnons une exemple de réseau de neurone classificateur dense (=fully-connected) à 2 couches (=1 couche d'entrée, 1 couche cachée, 1 couche de sortie).

On se donne une fonction non linéaire ℓ, par exemple

- $\ell(x) = x1_{x>0}$ la fonction relu.
- $\ell(x) = \tanh(x)$
- $\ell(x) = \frac{1}{1+e^{-x}}$ la sigmoïde.

Si V est un vecteur, on note $\ell(V)$ l'application de ℓ à toutes les composantes de V.

On prend comme paramètre $\theta = (w, b, w', b')$ composé de deux matrices et de vecteurs. Puis on choisit

$$f_\theta(x) = w' \cdot \ell(w \cdot x + b) + b'$$

puis

$$L_\theta(y|x) = \Big(\texttt{SM} \circ f_\theta(x)\Big)_y$$

DESSIN

$$f_\theta(x) = w'' \cdot \ell\Big(w' \cdot \ell(w \cdot x + b) + b'\Big) + b''$$

puis

$$L_\theta(y|x) = \Big(\texttt{SM} \circ f_\theta(x)\Big)_y$$

3.4 Classifier

ça y est: vous avez trouvé une bonne vraisemblance $L_{\widehat{w}}$ pour décrire vos données.

Maintenant on vous donne une nouvelle entrée x_o. Pour prédire la sortie correspondante, le plus naturelle, c'est de prendre la classe qui a la plus grande probabilité:

$$y_o = \operatorname*{argmax}_y L_{\widehat{w}}(y|x_o)$$

Cependant toutes les classes n'ont pas la même importance. Imaginons que nous cherchons à repérer des malades en vue de les soigner. La classe 1 (ou positif) étant "Malade", et la classe 0 (ou négatif) étant "sain". Supposons que

$$L_{\widehat{w}}(\cdot|x_o) = [0.52, 0.48]$$

Dans ce cas là, il vaut peut-être mieux classer le sujet "malade" pour le soigner par précaution.

En classification binaire (2 classes), on a inventer de nombreux outils pour choisir le bon seuils de probabilité: courbe ROC, score AUC, score F1.

En classification multi-classe, on analyse surtout la matrice de confusion.

On verra tout cela dans les T.P.

3.5 Exo

Dans cette partie, nous nous plaçons dans le cas le plus simple où il y a deux descripteurs $p = 2$ et deux classes $k = 2$. On considère un modèle logistique que l'on a entrainée avec des données $Train$. On a obtenu des paramètres $(\hat{w}, \hat{b})$ optimaux.

On choisit ensuite de classifier un individu par la méthode la plus simple:

$$\hat{y}_o = 1 \;\Leftrightarrow\; L_{\widehat{w}}(1|x_o) > L_{\widehat{w}}(0|x_o) \;\Leftrightarrow\; L_{\widehat{w}}(1|x_o) > 0.5$$

A quoi ressemble la frontière de décision c.à.d la limite entre les $x \in \mathbb{R}^2$ dont la classe prédite est 0 et ceux dont la classe prédite est 1.

Même question si maintenant $k = 3$.

4 Variantes

4.1 Classification binaire

On se place dans le cas de deux classes: $k = 2$.

Nous notons les logits $f_\theta(x) = [f_{\theta,0}(x), f_{\theta,1}(x)] = [f_0, f_1]$.

Posons $g = f_1 - f_0$. La loss peut se réécrire:

$$\begin{aligned} L &= -1_{y=0} \log\left(\frac{e^{f_0}}{e^{f_0}+e^{f_1}}\right) - 1_{y=1} \log\left(\frac{e^{f_1}}{e^{f_0}+e^{f_1}}\right) \\ &= -(1-y)\log\left(1 - \frac{e^{f_1}}{e^{f_0}+e^{f_1}}\right) - y\log\left(\frac{e^{f_1}}{e^{f_0}+e^{f_1}}\right) \\ &= -(1-y)\log\left(1 - \frac{1}{1+e^{f_0-f_1}}\right) - y\log\left(\frac{1}{1+e^{f_0-f_1}}\right) \\ &= -(1-y)\log\Big(1-\sigma(g)\Big) - y\log\Big(\sigma(g)\Big) \end{aligned}$$

où $\sigma(g) = \frac{1}{1+e^{-g}}$ est la fonction sigmoide.

On voit que la seule chose qui importe au final, c'est la différence $f_1 - f_0$. On n'a donc des paramètres redondant dans notre modèle.

Ainsi la variante couramment utilisée est la suivante: je prend une fonction g de $\mathbb{R}^p$ dans $\mathbb{R}$. Puis prend comme loss pour une observation (x, y) la "binary-cross-entropy":

$$loss_\theta(x, y) = -(1 - y) \log \Big(1 - \sigma\big(g_\theta(x)\big)\Big) - y \log \Big(\sigma\big(g_\theta(x)\big)\Big)$$

4.2 SVM

$$loss_\theta(x, y) = \sum_{u \neq y} \max(0, f_{\theta,u}(x) - f_{\theta,y}(x) + \Delta)$$

Pour avoir une loss de zéro, il faut que le logit de la bonne classe f_y dépasse d'au moins Δ le score des logits des autres classes f_u.

Cette loss est aussi appelée 'hinge'. On utilise parfois de carré de la hinge pour pénaliser violemment les mauvais logits.