TD 3 - Descente de gradient, Modèles linéaires

Exercice 1 – Apéro

Q 1.1 Parmi les fonctions suivantes, lesquelles sont convexes :
$f(x) = xcos(x)$, $g(x) = -log(x) + x^2$, $h(x) = x\sqrt{x}$, $t(x) = -log(x) - log(10 - x)$?

Toutes sauf $xcos(x)$
Pour $xcos(x)$, montrer que la fonction possède plusieurs points qui s'annulent. Puisque $cos(x)$ s'annule de manière périodique, $xcos(x)$ ne peut pas être convexe.
Pour les autres étudier la dérivée seconde : si la dérivée seconde > 0, alors fonction convexe : une fonction convexe est une fonction dont l'accelération ne fait qu'augmenter.
Somme de deux fonctions convexes est convexe (se déduit facilement de ce qu'on vient de dire précédemment)
Autre caractérisation issue de l'inégalité de Jensen : f est convexe ssi $f(\sum_i \lambda_i x_i) < \sum_i \lambda_i f(x_i)$ pour $\sum_i \lambda_i = 1, \lambda_i \geq 0$ (toutes les cordes sont supérieures à la fonction).
$\frac{\partial^2 g}{\partial x^2} = 2 + (1/x^2) > 0$
$\frac{\partial^2 h}{\partial x^2} = (3/(4\sqrt{x})) > 0$
$\frac{\partial^2 t}{\partial x^2} = 1/x^2 + 1/(10 - x)^2 > 0$

Q 1.2 Soit une application linéaire $f \in \mathbb{R}^n \to \mathbb{R}$; rappeler ce qu'est le gradient de f : $\nabla f(\mathbf{x})$. Donner le gradient de $f(\mathbf{x}) = 2x_1 + x_2^2 + x_2x_3$

$\nabla f(\mathbf{x}) = \left(\frac{\partial f(\mathbf{x})}{\partial x_i}\right)'_{i\in\{0;n-1\}}$
$\nabla f(\mathbf{x}) = (2, 2x_2 + x_3, x_2)$

Q 1.3 Exprimer $\nabla_{\mathbf{x}}(f(\mathbf{x}) + g(\mathbf{x}))$, $\nabla_{\mathbf{x}} tf(\mathbf{x})$.
Donner l'expression de $\nabla_{\mathbf{x}} b'\mathbf{x}$ avec $b \in \mathbb{R}^d$ et $\nabla_{\mathbf{x}} \mathbf{x}'A\mathbf{x}$ pour A symétrique.

$\nabla_{\mathbf{x}}(f(\mathbf{x}) + g(\mathbf{x})) = \nabla_{\mathbf{x}}(f(\mathbf{x})) + \nabla_{\mathbf{x}}(g(\mathbf{x})) = (\frac{\partial f(\mathbf{x})}{\partial x_i})_{i\in\{0;n-1\}} + (\frac{\partial g(\mathbf{x})}{\partial x_i})_{i\in\{0;n-1\}}$
$\nabla_{\mathbf{x}} tf(\mathbf{x}) = t\nabla_{\mathbf{x}} f(\mathbf{x}) = t(\frac{\partial f(\mathbf{x})}{\partial x_i})_{i\in\{0;n-1\}}$
$\nabla_{\mathbf{x}} b'\mathbf{x} = (\frac{\partial b_i x_i}{\partial x_i})_{i\in\{0;n-1\}} = (b_i)_{i\in\{0;n-1\}} = b$
$\nabla_x x'Ax = (\sum_j a_{i,j}x_j)_{i\in\{0..n-1\}} + (\sum_j a_{j,i}x_j)_{i\in\{0..n-1\}}$.
Et puisque B est symétrique alors $= 2(\sum_j a_{i,j}x_j)_{i\in\{0..n-1\}} = 2Ax$

Exercice 2 – Régression linéaire

Soit un ensemble de données d'apprentissage $\mathcal{D} = \{\mathbf{x^i}, y^i\}_{i=1,\ldots,N}, \mathbf{x^i} \in \mathbb{R}, y^i \in \mathbb{R}$.
Par convention que l'on suivra dans toute la suite du cours, la matrice de données sera notée X, où chaque ligne correspond à un exemple. La matrice Y des réponses est donc une matrice colonne ; la matrice W des poids également. L'erreur sur $\mathcal{D}$ sera notée $C(W)$.

Q 2.1 Résolution analytique

Q 2.1.1 Rappeler le principe de la régression linéaire. Quelle fonction d'erreur $C(W)$ est utilisée ?

$C(W) = \sum_{i=1}^{N} (< \mathbf{x}^i, W > -y^i)^2$

Q 2.1.2 Quelles sont les dimensions des matrices X, W et Y ? Rappeler la formulation matricielle de l'erreur.

$C(W) = (XW - Y)'(XW - Y)$

Q 2.1.3 Trouver analytiquement la matrice W solution de la régression linéaire, qui minimise $C(W)$.

$\nabla_W C(W) = \nabla_W W'(X'X)W - Y'XW - YW'X' + Y'Y = 2X'(XW - Y) = 0 \Leftrightarrow X'XW = X'Y, \quad W = (X'X)^{-1}X'Y$
Pb : $(X'X)$ peut ne pas être inversible.
En pratique, afin de s'orienter vers une solution unique avec des propriétés désirables, on préfère considérer $C(W) = (XW - Y)'(XW - Y) + \lambda||W||^2$, qui donne $W = (X'X + \lambda I)^{-1}X'Y$ toujours calculable (avec solution régularisée L2).

Q 2.1.4 Même question si l'on considère maintenant une machine linéaire avec biais. Quelle est la valeur optimale du biais w_0 dans ce cas ?

$C(W) = (XW + W_0 - Y)'(XW + W_0 - Y)$ avec W_0 un vecteur colonne composé que de w_0 ($W_0 = w_0\mathbb{1}$)
Pour w_0, on utilise la dérivée : $\sum_i 2(x^i w + w_0 - y^i) = 0 \Leftrightarrow w_0 = \bar{y} - \bar{x}W$ (avec $\bar{y}$ la moyenne des y et $\bar{x}$ le vecteur x moyen).
Soit $\bar{X} = X - \bar{x}\mathbb{1}$ et $\bar{Y} = Y - \bar{y}\mathbb{1}$
$C(W) = (\bar{X}W - \bar{Y})'(\bar{X}W - \bar{Y})$
$\nabla_W C(W) = 2\bar{X}'(\bar{X}W - \bar{Y}) = 0 \Leftrightarrow \bar{X}'\bar{X}W = \bar{X}'\bar{Y}, \quad W = (\bar{X}'\bar{X})^{-1}\bar{X}'\bar{Y}$

Q 2.2 Rappeler le principe de l'algorithme de descente du gradient. Donner son application au cas de la régression linéaire.

On a calculé le gradient au dessus.

Q 2.3 On considère dans la suite un problème à 2 dimensions.

Q 2.3.1 Tracer l'espace des paramètres en 2D. Positionner arbitrairement les points $\mathbf{w}^0$, point initial, et $\mathbf{w}^*$, solution analytique du problème. Etant donnée la nature quadratique du coût, tracer les iso-contours de la fonction de coût dans l'espace des paramètres. Quelle est la forme de la fonction de coût $C(\mathbf{w}^0)$ dans l'espace des paramètres ?

Q 2.3.2 Dessiner le vecteur $\nabla C(\mathbf{w}^0)$. A quoi correspond ce vecteur géométriquement ?

forme elliptique autour de W^*, ∇C est orthogonal a la tangeante à l'ellipse, soit vers le centre de l'ellipse.

Exercice 3 – Régression logistique

Q 3.1 On considère un ensemble de données X muni d'étiquettes binaires $Y = \{0, 1\}$. En régression logistique, on considère que le log-rapport des probas conditionnelles $p(y|x)$ peut être modélisé par une application linéaire : $\log\left(\frac{p(y|x)}{(1-p(y|x))}\right) = \theta.x$.

- Quel est le but ?
- Quelle étiquette prédire pour x si $\theta.x > 0$?
- Que vaut $p(y|x)$? Tracer la fonction $p(y|x)$ en fonction de $\theta.x$.
- En déduire le type de frontière que la regression logistique permet de déterminer

- Adaptation de la regression à la classification
- Si $\theta.x > 0$: prédire y, sinon 1-y
- Frontière linéaire
- $\log\left(\frac{p(y|x)}{(1-p(y|x))}\right) = \theta.x$ donc $\frac{p(y|x)}{(1-p(y|x))} = e^{\theta.x}$ et donc $p(y|x) = \frac{e^{\theta.x}}{1+e^{\theta.x}}$. En multipliant le numérateur et le dénominateur par $e^{-\theta.x}$, on obtient $p(y|x) = \frac{1}{1+e^{-x}}$.

Q 3.2 Pour une dimension x_i, quelle est l'influence de sa valeur pour $p(y|x)$? Quelle est la limite de la régression logistique ?

si le poids est positif, la variable contribue positivement, sinon negativement (cas x_i binaire : contribue ou pas, cas reel : plus elle est elevée plus sa contribution est forte si le poids est positif).
Limite : il faut que le log ratio des probas soit linéaire. Séparation linéaire des données

Q 3.3 Soit θ les paramètres recherchés. Quelle est l'expression de la vraissemblance conditionnelle de *theta* par rapport à un exemple (x, y) ? La log-vraissemblance ? Et sur un ensemble d'exemples $\mathcal{D}$?

Maximum de vraissemblance :
$\theta^* = \arg\max_\theta \sum_i^n y^i ln(\frac{1}{1+e^{-\theta.x^i}}) + (1-y^i) ln(1 - \frac{1}{1+e^{-\theta.x^i}}) = \arg\max_\theta - \sum_i ln(1 + e^{-(2y^i-1)<\theta.x^i>})$

Q 3.4 Proposer un algorithme pour résoudre le problème de la régression logistique.

Gradient : $\frac{\partial}{\partial w_j} = \sum_i ((2y^i - 1)x_j^i) e^{-(2y^i-1)<\theta.x^i>} \frac{1}{1+e^{-(2y^i-1)<\theta.x^i>}} = \sum_i (2y^i - 1)x_j^i \frac{1}{1+e^{(2y^i-1)<\theta.x^i>}}$.
Attention : maximisation du log vraissemblance, donc minimisation de l'opposé...

Exercice 4 – Optimisation d'un modèle gaussien par descente de gradient

Nous disposons ici d'un jeu de données non-étiquetées : $\mathcal{D} = \{\mathbf{x}_i\}_{i=1,\ldots,N}, \mathbf{x}_i \in \mathbb{R}^d$.
Nous souhaitons apprendre en mode non supervisé un modèle gaussien correspondant aux données de $\mathcal{D}$. Le modèle gaussien est défini par un ensemble de paramètres $\{\mu, \Sigma\}$

Q 4.1 Exprimez la log-vraisemblance en supposant les exemples de $\mathcal{D}$ statistiquement indépendants.

Q 4.2 Solution analytique

Q 4.2.1 Que vérifie la solution $W^\star$ du maximum de vraisemblance ? Montrez que la solution $W^\star$ du maximum de vraisemblance correspond à la moyenne et la covariance empirique des données $\mathcal{D}$ dans le cas où Σ est une matrice diagonale.

Q 4.3 Méthode de gradient

Q 4.3.1 Déterminez le gradient de la vraisemblance en un point W_0.

Q 4.3.2 Ecrire deux algorithmes de gradient batch et stochastique permettant d'apprendre une loi gaussienne à partir de $\mathcal{D}$.

$$logL = \log \Pi p(x|\theta) = \log \Pi_i \frac{1}{\alpha\sqrt{\det(\Sigma)}} \exp(-0.5(\mu - \mathbf{x}_i)^T \Sigma^{-1}(\mu - \mathbf{x}_i))$$

dans le cas d'une matrice de variance diagonale ($\Sigma = [\sigma_1^2, \ldots]$) :

$$logL = -0.5 \sum_i \sum_j \delta_{ij}^2 \sigma_j^{-2} - N \log(\alpha) - 0.5 \sum_i \log(\Pi_j \sigma_j^2)$$

avec $\delta_{ij} = (\mu_j - x_{ij})$

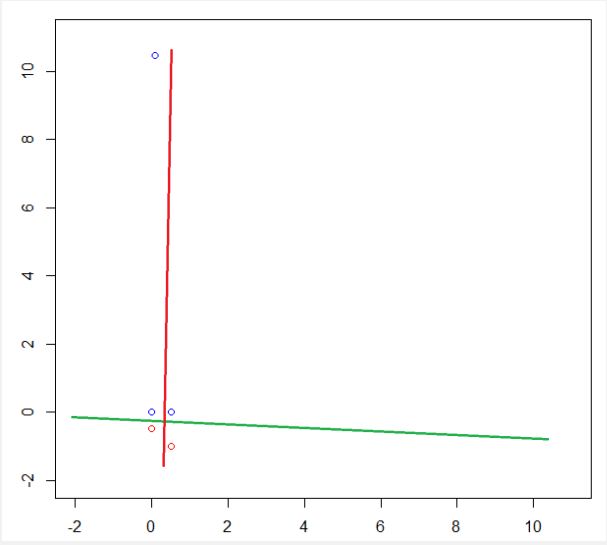
$$\frac{\partial logL}{\partial \mu_j} = -0.5 \sum_i 2(\mu_j - x_{ij})\sigma_j^{-2} = 0 \Leftrightarrow N\mu_j = \sum_i x_{ij}$$

$$\frac{\partial logL}{\partial \sigma_j} = -0.5 \sum_i -2\delta_{ij}^2 \sigma_j^{-3} - N\frac{\sigma_j}{\sigma_j^2}$$

$$\sigma_j^2 = 1/N \sum_i \delta_{ij}^2$$

Exercice 5 – Évaluation(s) de l'erreur

Q 5.1 Rappelez la fonction coût au sens des moindres carrés sur un problème d'apprentissage binaire. Proposer quelques exemples pour montrer que les échantillons correctement classés participent à la fonction coût.



ligne verte = hinge, ligne rouge = least squares

Q 5.2 En faisant appel à vos connaissances sur le perceptron, proposez une nouvelle fonction coût ne pénalisant que les points mals classés.

$$C = \sum_{i \in \mathcal{D}} (-f(\mathbf{x}_i) \times y_i)_+ = \begin{cases} 0 \text{ si pas d'erreur, cad } f(\mathbf{x}_i) \times y_i > 0 \\ -f(\mathbf{x}_i) \times y_i \text{ si err., cad } f(\mathbf{x}_i) \times y_i < 0 \end{cases}$$

Q 5.3 En imaginant une fonction f de complexité infinie (capable de modéliser n'importe quelle frontière de décision), tracez à la main la frontière de décision optimale au sens des coûts définis précédemment pour le deux problèmes jouets de la figure 1. Ces frontières sont-elles *intéressantes* ? Quels problèmes se posent ?

Sur- apprentissage dans le cas du percetron à degrés de liberté infinis
Frontière peu efficace même en train dans le cas des moindres carrés

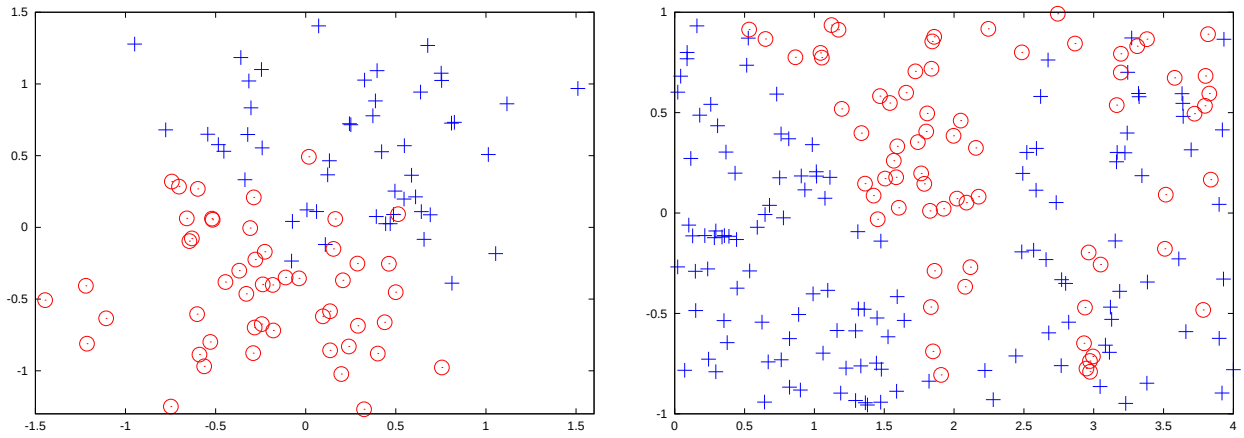


FIGURE 1 – Gaussiennes non séparables linéairement

Exercice 6 – Perceptron

Q 6.1 Soit $\mathbf{w} = (2, 1)$ le vecteur de poids d'une séparatrice linéaire. Dessinez cette séparatrice dans le plan. Précisez sur le dessin les quantités $< \mathbf{w}, \mathbf{x} >$ par rapport à un exemple $\mathbf{x}$ bien classé et mal classé. Que se passe-t-il pour le produit scalaire dans le cas d'un exemple mal classé avec la mise-à-jour $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$?

La séparatrice correspond à la droite $y = -2x$
Exemples $+1$ bien classés : (1,1) ou (-1,5), Exemple $+1$ mal classé :(-1,1)
La frontière se rapproche de l'exemple mal classé après mise à jour (ici ça l'amène même à être bien classé)

Q 6.2 Comment sont les classifieurs suivants par rapport à celui de la question précédente : $w^1 = (1, 0.5)$, $w^2 = (200, 100)$, $w^3 = (-2, -1)$?

Même frontière
Seulement une différence d'échelle mais ne modifie pas la classification

Q 6.3 Montrez que l'algorithme du perceptron correspond à une descente de gradient. La solution est-elle unique ?

$C(w) = \sum_{i=1}^{n} max(0, -y* < w, x >)$

Non la solution n'est pas unique plusieurs frontières possibles, plusieurs scales

Q 6.4 Quel problème peut-il se poser pour certaines valeurs de w ? Comment y rémédier ?

Si w nul => gradient nul
Hinge Loss en discutant que marge de 1 c'est pareil que marge de 0.0001, 100000 ou n'importe quel réel positif dans ce cas, ça revient à faire une inégalité non stricte pour pénaliser lorsque nul

Q 6.5 Donner un perceptron qui permet de réaliser le `AND` logique entre les entrées binaires x_1 et x_2 (positif si les deux sont à 1, négatif sinon) et un autre pour le `OR` logique.

$and(x) = 1 * x_1 + 1 * x_2 - 1.5$
$or(x) = 1 * x + 1 * x_2 - 0.5$

Q 6.6 Nous allons augmenter l'expressivité du modèle en étendant l'espace de représentation initial dans le cas 2D : $\mathbf{x} = [x_1, x_2]$. Soit la transformation ϕ suivante : $\phi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2, x_1x_2]$, considérons le modèle linéaire $f(\mathbf{x}_i) = \sum_j \phi_j(\mathbf{x}_i)w_j$.

- Quelle est la dimension du vecteur $\mathbf{w}$ dans ce cas ?
- A quoi correspond la projection *phi* ?
- Retracer les frontières de décision optimales sur la figure en utilisant cette nouvelle représentation.
- Pouvons nous retrouver les frontières linéaires de la question précédente dans ce nouvel espace ? Dans l'affirmative, donner les coefficients w_j associés.

Dimension de w=5
phi : projection quadratique => $\phi(x) = (1, x1, x2)'(1, x1, x2)$ Donc pour données en parabole dans l'espace de départ => linéaire dans l'espace d'arrivée
w1 et w2 identiques aux modèles précédents, 0 partout ailleurs

Q 6.7 Les frontières sont-elles plus *intéressantes* en utilisant la première ou la seconde représentation des données ? Pouvez vous comparer grossièrement l'amplitude de la fonction coût (au sens des moindres carrés par exemple) dans les cas linéaires et quadratiques ? Qu'en déduire ? Sur quel élément vous basez vous pour mesurer la qualité du modèle créé ?

Frontières plus intéressantes après projection
Il est très probable que le coût de la fonction quadratique soit plus élevé que le coût de la fonction linéaire... Ces coûts ne sont en fait pas vraiment comparables !
L'intérêt et la qualité sont intuitivement basés sur le pourcentage de bonne classification....

Q 6.8 Afin d'augmenter l'expressivité de notre classe de séparateur, nous nous tournons vers les représentations gaussiennes. Nous créons une grille de points $\mathbf{p}^{i,j}$ sur l'espace 2d, puis nous mesurons la similarité gaussienne du point $\mathbf{x}$ par rapport à chaque point de la grille : $s(\mathbf{x}, \mathbf{p}^{i,j}) = Ke^{-\frac{||\mathbf{x}-\mathbf{p}^{\mathbf{i},\mathbf{j}}||^2}{\sigma}}$. La nouvelle représentation de l'exemple est le vecteur contenant pour chaque dimension la similarité de l'exemple à un point de la grille.

- Quelle est la dimension du vecteur $\mathbf{w}$?
- Donnez l'expression littérale de la fonction de décision.
- Quel rôle joue le paramètre σ ?

Avec l'espace découpé en 10 sur chaque dimension :
w est de dimension 100 car chaque échantillon est représenté en dimension 100

$$f(\mathbf{x}) = \sum_{i=1}^{100} w_i s(\mathbf{x}, \mathbf{x}_i), \quad \mathbf{x}_i \text{ sont les points de la grille}$$

σ règle la zone d'influence des points. Moins on a d'échantillons, plus on augmente cette zone

On peut discuter de la similtude avec la taille de la fenêtre de Parzen ou le nombre de voisins du KNN.

Q 6.9 Introduction (très) pragmatique aux noyaux

- Que se passe-t-il en dimension 3 si nous souhaitons conserver la résolution spatiale du maillage ?
- Afin de palier ce problème, nous proposons d'utiliser la base d'apprentissage à la place de la grille : les points servant de support à la projection seront ceux de l'ensemble d'apprentissage. Exprimer la forme littérale de la fonction de décision dans ce nouveau cadre. Quelle est la nouvelle dimension du paramètre **w** ?
- Que se passe-t-il lorsque σ tend vers 0 ? vers l'infini ? A-t-on besoin de toutes les dimensions de w ou est-il possible de retrouver la même frontière de decision en limitant le nombre de données d'apprentissage ? A quoi cela correspond-il pour $||\mathbf{w}||$?

Explosion du nombre de points :
Avec l'espace découpé en 10 sur chaque dimension :
w est de dimension 1000 car chaque échantillon est représenté en dimension 1000
$f(x) = \sum_j w_j s(x, x_j)$
Dimension = nombre d'échantillons
Quand σ tend vers 0 : on ne peut prédire que pour les points observés
Quand σ tend vers l'infini : toutes les similarités tendent vers 1 : classe majoritaire predite pour tout le monde
On peut se concentrer sur des points supports, les plus proches de la frontière optimale = Minimisation de $||\mathbf{w}||$