Regression linéaire simple (MSE) y = W. x

Page 46 : ½ (on aime l’ajouter comme ça avec la descente du 2 en dérivant, le constant disparait)

**Regression simple**

**Y = W. x / w , y x**

**Regression multiple**

**Y = W. x / w , y x**

On peut avoir y € (sortie multiple) donc regression multiple

Les poids de la droite suivent une loi normale

Y = W. x + ~N (0, sigma\*\*2)

Trouver un modèle qui explique au moins mes données => maximiser la vraissemble (probabilité de l’ensemble des données)

Exemple : donnérs dist gaussienne

Vraisbmalcce fonction des paramètres (une fonction de mu, j fais chnager mon mu)

La vraissemeble sur ces données = probabilité qu’un modèle est generé c données

Sigma1

Sigma2

Mu2

Mu1

L (mu, sigma, D) = Produit pi(x\_i ; mu ; sigma) / en pradique on maximise le log de la vraissemblance (logvraissemblance)

i/X

vraissemblance conditionnelle L(w) = produit p(y/x ;w) page 48

Derniere formule page 48 on veut maximiser la deuxieme partie

Regression logistique (classification) modèle linéaire

Attention ;: utilise une fonction non linéaire « sigmoïde » »fonction d’apprentissage » mais la décision est linéaire

Fonction a seuil y [0 ; 1]

Fonction logistique y [0 ; 1] (la pente se module comme on veut, on peut meme la rendre à seuil).

Dans les réseaux de neurones on utilise la fonction TANH y [ -1 ; 1]

A compléter photo.

Formule photo

Cout pour la classification ?

1. L2
2. Entropie croisée (théoriquement c’est plus propre, en pratique les deux). C’est une distance. Avec la distribution calculer on veut approcher la distribution

Pour chaque valeur de x, on a une loi de Bernoulli

P (y =1/ x ; w) = (x), P (y = 0/ x ; w) = 1 - (x)

Maintenant la vraisemblance, le produit des couples x, y de ma base d’apprentissage

L(w) = formule page 50

Puis on passe à la log vraisemblance et on la maximise

Pour la maximiser on utilise l’algorithme d’optimisation de DG (dans le cours elle est présentée en BATCH) page 50

Généralisation avec P classes :

Passager d’un problème de 2 classes à p classes :

1. Autant de sortie que de classe (2 classes une sortie, p classes p sorties)

Règle de décision classe = argmax( (x))i / i = 1 … P

W1

1. Softmax

Les logiciels (Pytorch/Tensorflow) dont de la différentiation/dérivation automatique (calculer des dérivées complexes automatique) en utilisant des graphes de calcul (Page 53).

Dérivé de la fonction de cout par rapport a la prédhiction

Dérivé de la prédiction par rapport à S (1dim)

Dérivé de S par rapport à W (vecteur gradient)

Donc = page 55

Avec le graphe de calcul on peut : Quelle est le x qui à générer telle sortie.

Risque théorique [(y – h(x)\*\*)2]

La fonction optimale est une fonction de x

On veut optimiser R = (x)) \*\*2]

R = (x)) \*\*2] image

(x)) \*\*2] = E[y|x] + E[y|x] - ) \*\*2] % Développer le \*\*2

E[y|x] + E[y|x] - ) \*\*2] =

E[y|x]) \*\*2] + [E[y|x] - ] \*\*2 + 2 [(y - E[y|x]) \*E[y|x] - ]]

cte

Image

Perceptron multicouche :

La fonction d’activation : TANH ou Sigmoïde