Chap8: Classification 1: utiliser régression linéaire en convertissant classes en chiffres(ex : homme 0, femme 1). 0/1 fonctionne pour classifieurs binaires. Positif 🡪1, négatif 🡪0. Trjs donner valeur 1 à la variable d’intérêt. Droite régression coupe ces classes, même s’il existe séparateur dû à la minimisation de l’erreur carrée. **Régression logistique**  1ere méthode pr trouver fonction de séparation de 2 classes. Vise à convertir valeur continue en valeur de probabilité ∈ [0, 1]. Utilise fonction d’erreur différente de celle de régression linéaire🡪l’entropie croisée. 1)Fonction sigmoïde : donne la proba de *Xi* d’appartenir à une classe particulière.Proba d’appartenir à la classe 1. F(0) = ½, F()=1, F()=0. quand juste 1 attribut Une image contenant texte

Description générée automatiquement2)Fonction d’erreur - entropie croisée : fonction convexe ! Trouver meilleur séparateur entre 2 classes avec méthode gradient(trouver minimum global). Erreur est 0 slm si classes sont linéairement séparables(si données peuvent etre séparés par un plan, alors ce plan sera trouvé par régression logistique). Faut ajouter colonnes non linéaires dans X pour séparer classes non linéairement séparables(). *y* est binaire. 1 seul des termes de la somme est actif pr chaque enregistrement. Une image contenant texte

Description générée automatiquement Classification-Problèmes : 1) Ensemble d’entraînement déséquilibré : (ex :identification d’un terroriste) Considérons droite séparation optimale pour classes très déséquilibrées, 1 positif(etre terroriste) vs 1,000,000 négatifs(pas etre terroriste). Meilleure droite trouvée par régress logistique essaiera d’être très loin du grand groupe des gens non terroristes au lieu de se placer entre les classes. Présence des faux négatifs! Mm classer tt le monde comme non terroriste aura peu d’impact pour l’entropie croisée. Solution🡪utiliser le même nb d’exemples + et - . Travailler + fort pour trouver des membres de la classe minoritaire. Supprimer des éléments de la classe majoritaire(problème, on perd des données qui peuvent etre utiles pr notre prédiction). Peser + lourdement données de la classe minoritaire, mais méfiez-vous du surapprentissage. Répliquer membres de la classe minoritaire, idéalement avec une perturbation aléatoire[échantilloner avec valeurs gaussiennes](ex : on a 10 terroristes pr 10 millions de non terroristes).2) Classification multi-classe : Tâches de classification ne sont pas trjs binaires. Un film est-il une comédie, drame,sci-fi, documentaire? Sélectionner classe de proba la + élevée comme étiquette prédite. Classific multi-classe devient + difficile à mesure que le nb de classes augmente(prédire si chat, chien, plante et prédire leurs sous-classes race, couleur). 3)Fonctions de partition: normaliser valeurs de sorties pr obtenir des probas. Classifieurs binaires indépendants ne produisent pas vraies probas (somme ne vaut pas 1). Régression multinomiale combine les classifieurs au niveau de l’entraînement.Apprentissage machine : regression linéaire et logistique + les algos qui apprenent à partir d’exemples. TT les algos qui résoudent des problemes complexes. Limitations de la régression : Besoin attributs non linéaires (ex. x1 x2) pour que régression puisse bien séparer données non linéairement séparables. Implique bcp de combinaisons d’attributs à tester. Apprentissage profond(Deep learning) : Couches cachées créent compositions de fonctions non linéaires ⇒ + grand pouvoir de représentation. Faire apprendre le réseau signifie définir les valeurs des coefficients *w*. + il y a de connexions entre neurones, + il y a des paramètres à apprendre. Analyser l’ensemble de données d’entraînement étiquetées et ajuster coefficients pour que nœuds de sortie génèrent qqc proche de *yi* lorsque réseau est alimenté avec *X*i . Pour de nombreuses tâches, la complexité des réseaux n’est pas nécessaire. Réseaux de neurones fonctionnet par surapprentissage, trouvant moyen d’utiliser millions d’exemples pour s’adapter à des millions de paramètres. **Surapprentissage** : Faible erreur de prédiction. Variance élevée. Uniquement lié à la compléxité, flexibilité du modèle. Modèle trop complexe, si nouvelles données arrivent, modèle va faire des erreurs. Fonction apprise doit avoir une bonne performance de généralisation. Cela est vérifié sur **un ensemble de test**. Trop flexible, il va avoir une fonction compliqué qui va séparer les donnees et qui va etre specifique à nos donnees. **Sous-apprentissage** :Erreur de prédiction élevé, variance faible. Le modèle est pas assez complexe pr resoudre probleme, pas capable d’apprendre la fonction pr apprendre. Apprentissage profond : évite le pire comportement du surapprentissage en utilisant des moyens - précis **d’encoder** la connaissance(arreter apprentissage un peu d’avance pr eviter le surapprentissage quand modele devient complexe). Adapté aux domaines avec d’énormes quantités de données étiquetées. Facilitent construction de modèles d’apprentissage profond. Chaque neurone *j* du réseau calcule une fonction non linéaire φ(vj) où v est donné par : où : tij est l’input au neurone j avenant du neurone i et wij est le poids donné à cet input. La neurone *j* est activé en fonction de la valeur φ(vj) et de la fonction d’activation du neurone. Backpropagation : la sortie depend de la couche d’avant qui, dont la couche d’avant depend de la couche d’avant elle-mm. Réseaux de neurones entraînés par un algo de descente gradient. Changements pr chaque exemple d’entraînement sont ramenés à des niveaux inférieurs. Fonctions d’activation non linéaires aboutissent à une fonction d’erreur non convexe, mais son optimisation produit de bons résultats. Classification 2 : Machines à vecteurs de support **SVM** : SVMs travaillent en recherchant séparateurs linéaires de marge maximum entre 2 classes(on veut séparateur qui maximise écart). La raison de la variation des performances des 2 classifieurs est que l’instance de test est placée dans une région limite entre les 2 classes. Classifieur avec le + grande marge est préféré ⇒ + robuste. vecteurs de support 🡪 Pts de données d’entraînement sur les hyperplans parallèles à l’hyperplan séparateur classifieur. Hyperplan séparateur est au milieu de ces 2 hyperplans afin d’atteindre la classification la + précise. Une image contenant texte

Description générée automatiquement On assume que : 1) données d’entraînement sont linéairement séparables 2) classes sont étiquetées +1 et -1. Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

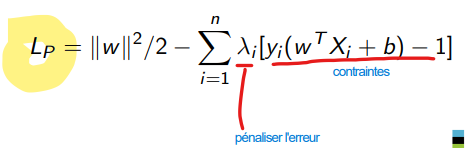
Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Relaxation Lagrangienne : associer un ensemble n de multiplicateurs non-négatifs λ = (λ1 . . . λn) ≥ 0 pour les différentes inégalités de la marge. La fonction objectif est augmentée en incorporant une pénalité lagrangienne (pénaliser l’erreur)



\* Si LP est minimisé par rapport à w et b pour un λ particulier, puis maximisée(trouver + gros coefficients) par rapport aux multiplicateurs λ, la solution duale résultante LD est une borne inférieure de la fonction objectif optimale O de la SVM linéaire.

Une image contenant texte

Description générée automatiquement

Dans le cas de la SVM linéaire, ce résultat est encore + fort car la f.o. est convexe, c.a.d O = LD.

Une image contenant texte

Description générée automatiquement

Pour une instance de test Z, sa classe F(Z) est défini par : Une image contenant texte

Description générée automatiquement

Classification et la valeur de la f.o. de la SVM sont obtenus sans connaître la matrice de données X. Tout est exprimé par produits scalaires. Cela est crucial pour l’application de la SVM pour des données pas linéairement séparables (**Kernel trick**).

Une image contenant texte

Description générée automatiquement

Projections: Non-linéarité du classifieur dépend de cmt l’espace est projeté à des dimensions + élevées. Avec les SVMs est qu’il n’est pas nécessaire de connaître les features des données dans le nouveau espace.SVM non linéaire : définir le produit scalaire (la fonction de similarité) dans la représentation transformée Φ(X) avec l’utilisation d’une fonction noyau K(Xi , Xj). Calculs effectués dans l’espace d’origine avec la fonction noyau choisie. Transformation réelle Φ n’a pas besoin d’être connue tant que la fonction noyau K soit connue. En utilisant similarité basée sur des noyaux choisis avec soin, on obtient des SVMs non-linéaires. Différentes manières de modéliser la similarité entre les données avec des noyaux. Chacune a des avantages sur certains ensembles de données, donc nécessaire de jouer avec options disponibles dans les librairies pour obtenir les meilleures performances. Une image contenant table

Description générée automatiquement

*k* plus proches voisins : 1)Trouver les k enregistrements de X qui sont les + proches de la nouvelle donnée Z que l’on veut classifier.2) Faire voter chacune de ces enregistrements selon leurs classes associées y. 3) Retourner la classe majoritaire. Succès de l’algo dépend de 2 facteurs : quantité de données d’entraînement et qualité de la mesure de distance (2 enregistrements similaires doivent faire partie de la mm classe). Grand pouvoir de représentation🡪 peut classifier des données non linéairement séparables. performance liée à notre capacité de stockage. Arbres de décision : Méthode très populaire pour l’apprentissage supervisé. Facile à utiliser et rapide. Peut aussi souffrir d’un surapprentissage. Mieux vaut s’arrêter lorsque gain d’information est faible, au lieu de 0.

Une image contenant texte

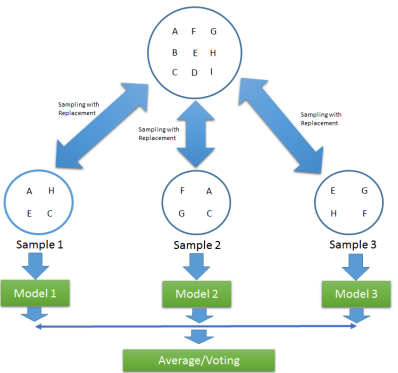
Description générée automatiquement

Entropie:

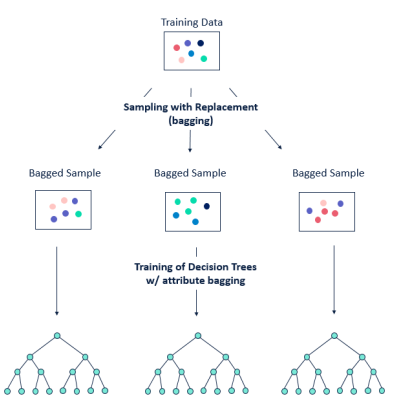
Une image contenant texte, montre

Description générée automatiquement

Meilleur attribut : Le gain (G) est la différence entre l’entropie initiale et l’espérance de l’entropie après que le feature A est choisi pour la séparation. C**ritère glouton** dans algo de construction d’une arbre de décision va choisir le feature qui maximise le gain. Ensemble d’arbres de décision : Combine pls arbres de décision pour produire une meilleure performance prédictive. Un groupe de modèles faibles se réunit pour former un modèle fort. Vote majoritaire entre plusieurs classificateurs augmente la robustesse et permet de quantifier notre niveau de confiance. Bagging : choisit des sous-ensembles de données choisies au hasard pour entraîner chaque arbre. Diminue le surapprentissage.



Forêt d’arbres de décision : Extension de la technique de bagging. Effectue aussi la sélection aléatoire des features plutôt que d’utiliser toutes les features pour construire les arbres.



Diminue la corrélation entre les arbres. Un feature très dominant oblige chaque arbre de décision à le choisir pour les premiers splits, ce qui fait que tous les arbres se comportent de façon similaire. La forêt d’arbres est composée par des arbres de décision non corrélés. Évaluation d’un modèle de classification : Étapes d’une évaluation : Paramètres🡪 ce que la fonction va apprendre. Hyperparametres🡪valeurs choisies au départ et non apprises. 1)Donnés séparés en 3 sous-nesembles : entrainement(70%), validation(15%) et test(15%). 2)On fait une liste de valeurs des hyper-paramètres à essayer.3) Pr chaque élément de cette liste, l’algo est exécuté sur l’ensemble d’entraînement et sa performance est mesurée sur l’ensemble de validation. Avec les meilleurs valeurs des hyper-paramètres identifiés avec l’ensemble de validation, on calcule la performance de l’algo sur l’ensemble de test. Procédure d’évaluation d’un algo de classification : quand pas assez de données pr séparer en 3 sous-ensembles. La validation croisée partitionne les données en *k* blocs de taille égale. On entraîne *k* modèles distincts. Le modèle *i* est entraîné sur l’union de touts les blocs sauf 1 (le bloc d’index i), et validé sur le bloc *i*. Bagging combine tout les *k* modèles entraînés dans un seul modèle de classification. Classification binaire :

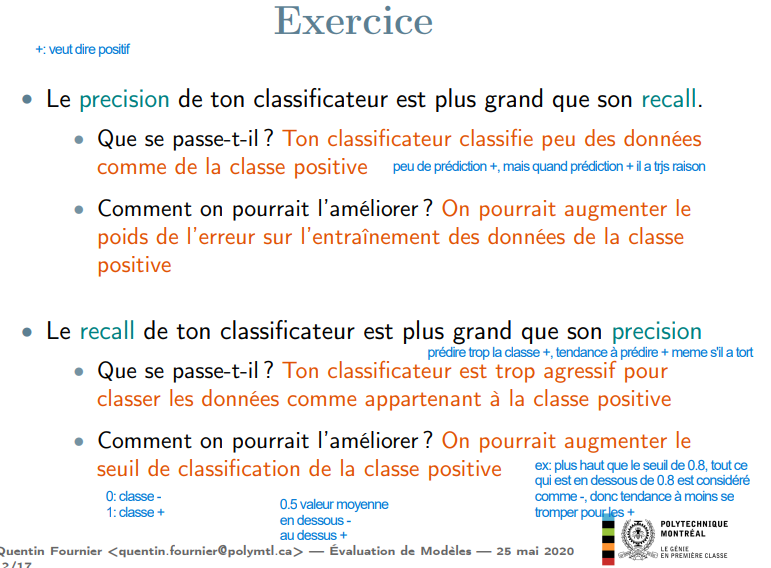
Une image contenant texte

Description générée automatiquement

Accuracy : le rapport entre les prédictions correctes et les prédictions totales. Un classificateur randomisé aurait une espérance d’accuracy de 50%. Si on prend juste la classe + nombreuse comme prédition (classificateur gourmand) accuracy ≥ 50%. Précision : Quand |P| << |N|, mesurer l’accuracy est inutile(si on a des classes qui sont sur-représenté). L’accuracy d’un classificateur gourmand serait de 95% lorsque Donc, ex :dataset de 10 TP et de 10 FP donne 10 / (10+0) = 1, mais pas trjs très précis! Atteindre valeur de precision élévée est impossible pr le classificateur randomisé ou le gourmand. Recall : mesure la capacité d’avoir raison uniquement sur les instances positives. Être + tolérants aux faux positifs (ex : erreurs où nous effrayons une personne en bonne santé avec un mauvais diagnostic) que les faux négatifs. Attention : dire que tout le monde a un cancer donne un recall parfait! Ceci donne un bon recall, mais une mauvaise précision! F-score : Mesure équilibré de score unique Moyenne harmonique est trjs < ou = à la moyenne arithmétique, ce qui rend difficile l’obtention d’un F-score élevé. Exercice EXAMEN :

Une image contenant texte

Description générée automatiquement



Receiver-Operator Curves (ROC) : Faire varier le seuil d’un modèle modifie le *recall* et la *precision*. La surface sous une ROC est une bonne mesure d’évaluation d’un modèle. Si seuil = 1, aucune instance +. Si seuil = 0, ca veut dire je prédis que tout est + et aucune instance est -. Évaluation de systèmes multiclasses : classification devient + difficile avec + de classes(ex : reconnaissance de chiffres). On note C[i, j] le nombre d’objets de la classe i classés comme de la classe j. Precision*i* est la fraction de tous les objets déclarés de la classe i qui étaient en fait de la classe i 🡪 🡪 instance de la classe i correctement prédite de la classe i / instances de la classe j. Recall*i* est la fraction de tous les membres de la classe i qui ont été correctement identifiés comme tel 🡪 🡪correctement prédit classe i / instances de la classe i. C[i,j] 🡪 i est l’objet de la classe i, j est le prédit commun appartenant à la classe j. Un taux de classification trop bas est décourageant et trompeur avec pls classes. Le taux de réussite du top-K donne du crédit si la bonne étiquette aurait été l’une des premières suppositions. Il est important de choisir K afin que de réelles améliorations puissent être reconnues. Si K = au nb de classes, le taux de réussite est 100%. Si K = 1 on a simplement l’accuracy(la précision et le rappel). On choisit K t.q la performance du classificateur est supérieure à celle d’un classificateur aléatoire. Choisir K pas trop élevé et pas trop petit! Valeurs numériques : l’erreur est une fonction de la différence entre la prévision f (X) et l’observation y : **Absolute error** :|f (X) − y|. **Absolute percentage error** : |f (X) − y|/y. **Square error** :(f (X) − y)^2 (toujours non-n/gatif). Ex : **Min square error** : (sensible aux outliers). **Root min square error** : (+ facile à interpréter).

Chap 9 :