Classification supervisée

Rémy Degenne

9 février 2022

Ces notes de cours contiennent uniquement un bref résumé de ce qui a été dit en cours. Reportez vous à vos notes pour plus de détails.

1 Cours 3 : Validation d'un classeur

Une fois un classeur construit (entrainé), il faut l'évaluer.

Un classeur peut aussi dépendre de "hyper-paramètres", qu'on choisit avant la construction et qui ne sont pas optimisés pendant l'entrainement. Exemples: k dans un k-NN, le nombres de couches et de neurones dans un réseau de neurones. On a envie de construire plusieurs classeurs et de choisir les hyper-paramètres qui donnent la meilleure performance.

2 Evaluer un classeur

Definition 1. L'erreur de classification $E \in [0,1]$ d'un classeur est la probabilité que ce classeur ne prédise pas correctement la classe d'une donnée de $\mathcal{D}$.

Note: cette définition suppose qu'on dispose d'une distribution de probabilité sur $\mathcal{D}$. Non calculable en pratique.

Le *taux de succès* est $1 - E$.

Definition 2. Si le classeur a été entrainé sur les données $\mathcal{X}_{\text{train}}$, l'erreur apparente (ou erreur d'entrainement, ou d'apprentissage) E_{app} est la proportion d'exemples de $\mathcal{X}_{\text{train}}$ mal classés.

E_{app} n'est pas une bonne mesure d'erreur, parce qu'elle ne mesure pas la généralisation. Le classeur qui apprend par coeur a $E_{app} = 0$.

Apprentissage et test Pour essayer d'obtenir une estimation de E, on teste le classeur sur des données qui n'ont pas servi à l'entrainement.

On entraine le classeur sur $\mathcal{X}_{\text{train}}$, mais on mesure son erreur E_{test} sur $\mathcal{X}_{\text{test}}$.

Si on dispose d'un seul ensemble d'exemples $\mathcal{X}$, on le sépare en deux (de taille inégale) avant l'apprentissage. Plusieurs manières de le faire:

- Aléatoirement. En pratique: attention à garder la correspondance entre les données et leurs étiquettes.

- En respectant les classes: le problème de l'aléatoire est qu'une classe peut se retrouver presque entièrement dans le *train* (et être presque absente du test). Solution: dans chaque classe, on coupe en deux.

Attention: le jeu de test ne doit jamais servir à prendre la moindre décision concernant le classeur. Par exemples: on ne peut pas choisir la taille d'un MLP en se servant de E_{test}, sinon $\mathcal{X}_{test}$ a indirectement servi à apprendre, et l'estimation de E par E_{test} n'est plus valable.

2.1 Mesures de performance

On a entrainé un classeur sur un jeu d'exemples $\mathcal{X}_{train}$, et on veut mesurer sa performance (sur $\mathcal{X}_{train}$ ou sur un autre ensemble comme $\mathcal{X}_{test}$).

On donne ici quelques notions pour la classification binaires (l'extension à des classes multiples est aisée).

Erreur L'erreur d'un classeur, comme définit précédemment, est la proportion d'exemples mal classés.

Précision et Rappel On suppose ici que les deux classes sont "positif" ou "négatif".

- VP: nombre de vrai positifs. Exemples classés positifs alors que leur étiquette est positif.
- FP: nombre de faux positifs. Exemples classés positifs alors que leur étiquette est négatif.
- VN: nombre de vrai négatifs. Exemples classés négatifs alors que leur étiquette est négatif.
- FN: nombre de faux négatifs. Exemples classés négatifs alors que leur étiquette est positif.

N est le nombre total d'exemples: $N = VP + FP + VN + FN$.
L'erreur sur $\mathcal{X}$ est $E_{\mathcal{X}} = \frac{FP+FN}{N}$.
Précision:
- pour les positifs: $\frac{VP}{VP+FP}$. la précision mesure la proportion d'exemples dont l'étiquette est positif parmi ceux qui sont classés comme positifs.
- pour les negatifs: $\frac{VN}{VN+FN}$

Rappel:
- pour les positifs: $\frac{VP}{VP+FN}$. Le rappel mesure la proportion d'exemples classés positifs parmi tous les exemples dont l'étiquette est positif.
- pour les negatifs: $\frac{VN}{VN+FP}$.

Mesure F: un seul nombre pour agréger précision et rappel.

$$F = \frac{2}{\frac{1}{rappel} + \frac{1}{precision}} = \frac{2VP}{2VP + FN + FP}$$

$F = 1$ si le classeur est parfait (sur ces données). F est proche de zéro si le classeur est mauvais. Pour que F soit proche de 1, il faut que le rappel et la précision soient tous les deux proches de 1.

Problème de la représentativité des classes Problème: les données peuvent contenir beaucoup d'exemples d'une classe et peu d'une autre. L'erreur apparente est une mesure qui ne cherche pas à "corriger" ce déséquilibre.

Un modèle peut sacrifier de la performance sur la classe minoritaire pour être meilleur sur la majorité, et obtenir un meilleur score global. Ce n'est pas souhaitables dans de nombreux cas. Par exemple, supposons qu'on essaie de détecter une maladie et que notre ensemble d'exemples a 99% de personnes saines et 1% de personnes malades. Alors le classeur qui classe toutes les données comme "saines" a un taux de succès de 99% : l'apprentissage aura du mal à trouver mieux.

Méthode 1: rééquilibrer les classes en enlevant des exemples (si on a beaucoup de données). Problème: on apprend potentiellement moins bien si on utilise moins d'exemples.

Méthode 2: donner un poids aux classes dans le calcul de l'erreur, qui compense la différence de taille.

2.2 (Hyper-paramètres et) sur-apprentissage

Sur-apprentissage: en dessinant les courbes d'erreurs sur l'apprentissage et le test, on voit qu'au cours de l'apprentissage l'erreur d'apprentissage diminue, mais l'erreur de test arrête de diminuer au bout d'un moment, voire augmente après avoir atteint un minimum. C'est le sur-apprentissage.

On peut sur-apprendre par rapport à l'apprentissage de paramètres, ou par rapport à l'ajout d'hyper-paramètres.

On aimerait pouvoir s'arrêter au moment où l'erreur de test est minimale, mais on n'a pas le droit de changer l'apprentissage en fonction de ce qu'on voit sur le test, sinon l'estimation de l'erreur de test n'est plus valide.

2.3 Mesurer la performance pendant l'apprentissage: la validation

Comme on n'a pas le droit d'utiliser l'ensemble de test pour faire des choix pendant l'apprentissage et que faire les choix à partir de l'ensemble d'apprentissage conduit à des choix biaisés (ce qui n'est pas souhaitable), on introduit un troisième ensemble.

Validation simple couper les exemples en trois ensembles: $\mathcal{X}_{train}$, $\mathcal{X}_{test}$ et $\mathcal{X}_{val}$. Le nouvel ensemble $\mathcal{X}_{val}$ sert à estimer l'erreur pendant l'apprentissage, et à prendre des décisions: quand s'arrêter, quelle taille de MLP prendre, etc.

L'estimation finale de l'erreur est faite sur l'ensemble de test.

Problème: on a encore enlevé des exemples à $\mathcal{X}_{train}$. On aimerait bien entrainer sur le plus grand nombre possible d'exemples.

Validation croisée On coupe $\mathcal{X}$ en $\mathcal{X}_{train}$ et $\mathcal{X}_{test}$ et on met $\mathcal{X}_{test}$ de côté. On coupe ensuite $\mathcal{X}_{train}$ en n ensembles (attention aux proportions des classes).

On procède à n apprentissages, et dans chaque apprentissage on prend comme ensemble d'entrainement $n-1$ morceaux de $\mathcal{X}_{train}$. On calcule une estimation de l'erreur sur le morceau restant.

A la fin, on garde le modèle qui a la meilleure performance sur son ensemble de validation. On peut ensuite évaluer ce modèle sur $\mathcal{X}_{test}$, après l'avoir éventuellement ré-entrainé sur $\mathcal{X}_{train}$ tout entier.

2.4 Niveaux de confiance

On sait maintenant évaluer un classeur après son apprentissage (test) et pendant (validation). Mais on n'obtient que des estimations. La question est donc: quelle confiance accorder à ces estimations ? Intuitivement, plus l'ensemble de test est gros, plus l'estimation est bonne. Mais un gros ensemble de test réduit la taille de l'ensemble d'apprentissage. Quelle taille minimale permet d'avoir une estimation fiable ?

Supposons que le classeur fasse une erreur avec une probabilité E. On mesure $E_{test} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathbb{I}\{\hat{y}_i \neq y_i\}$.

Chaque $Z_i = \mathbb{I}\{\hat{y}_i \neq y_i\}$ est une variable aléatoire Bernoulli, qui vaut 0 avec probabilité $1-E$ et 1 avec probabilité E. On cherche à savoir si $E_{test} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} Z_i$ est proche de E.

Par la loi des grand nombres, on sait que E_{test} se rapproche de E quand N_{test} devient grand. On sait aussi que sa variance est $\frac{E(1-E)}{N_{test}}$ et que quand N_{test} devient grand, $\frac{E_{test}-E}{\sqrt{E(1-E)/N_{test}}}$ tend vers une Gaussienne $\mathcal{N}(0,1)$ (théorème central limite).

On veut des intervalles de confiance, c'est à dire des bornes $-a(N_{test}, \delta)$ et $b(N_{test}, \delta)$ telles que avec probabilité $1-\delta$,

$$E_{test} - E \in [-a(N_{test}, \delta), b(N_{test}, \delta)]$$

Si on a obtenu un tel z, on peut dire que avec cette probabilité $E \in [E_{test} - z, E_{test} + z]$.

Pour trouver ces bornes, on utilise l'approximation que $\frac{E_{test}-E}{\sqrt{E(1-E)/N_{test}}}$ est Gaussienne. A partir de la loi d'une gaussienne centrée de variance 1, on obtient un z_δ tel que avec probabilité $1-\delta$

$$\frac{E_{test} - E}{\sqrt{E(1-E)/N_{test}}} \in [-z_\delta, z_\delta]$$

On peut maintenant résoudre en E: avec probabilité $1-\delta$, si on mesure E_{test},

alors $E \in [E_{inf}, E_{sup}]$, où

$$E_{inf} = \frac{1}{1 + z_\delta / N_{test}} \left(E_{test} + \frac{z_\delta^2}{2N_{test}} - z_\delta \sqrt{\frac{E_{test}}{N_{test}} - \frac{E_{test}^2}{N_{test}} + \frac{z_\delta^2}{4N_{test}^2}} \right) ,$$

$$E_{sup} = \frac{1}{1 + z_\delta / N_{test}} \left(E_{test} + \frac{z_\delta^2}{2N_{test}} + z_\delta \sqrt{\frac{E_{test}}{N_{test}} - \frac{E_{test}^2}{N_{test}} + \frac{z_\delta^2}{4N_{test}^2}} \right) .$$

On peut donc en conclure quelle valeur on veut pour N_{test}.