Un peu d'ingénierie des données

Laurent Miclet

IRISA (Lannion)
France

EMINES 2024

1. Un peu d'ingénierie des données
   - Acquisition et compréhension
2. Normaliser les données
   - Prétraitements
3. Réduire le nombre des attributs
   - Éliminer les attributs inutiles
   - Remplacer deux attributs corrélés par un seul
   - Utiliser l'Analyse en Composantes Principales et d'autres méthodes
   - Exploration

Plan

Plan

1. Un peu d'ingénierie des données
   - Acquisition et compréhension
2. Normaliser les données
   - Prétraitements
3. Réduire le nombre des attributs
   - Éliminer les attributs inutiles
   - Remplacer deux attributs corrélés par un seul
   - Utiliser l'Analyse en Composantes Principales et d'autres méthodes
   - Exploration

Plan

- Normalisation min-max : adapter linéairement les données à une plage comprise, par exemple, entre 0 et 1. La valeur minimale est 0 et la valeur maximale est 1.
- Normalisation par le test Z (hypothèse gaussienne) : mettre les données à l'échelle en fonction de la moyenne et de l'écart type : diviser la différence entre les données et la moyenne par l'écart type.
- Mise à l'échelle logarithmique : mettre les données à l'échelle en déplaçant le séparateur décimal de la valeur de l'attribut.

Plan

Quelques problèmes

- Doublons
- Points aberrants,
- Discrétisation,
- Données manquantes
- Augmentation (rotations, bruitage, etc)

Données manquantes

Destruction Destruction de la donnée ou de l'attribut

Imputation simple Mettre (imputer) à la donnée manquante la moyenne de l'attribut.

Imputation par régression En deux passes : d'abord une régression multivariée sur tous les exemples complets. Ensuite une interpolation sur les donnés manquantes par le modèle de régression obtenu.

Imputation par EM *Impute, estimate and iterate until convergence*.
L'étape E estime les valeurs manquantes sachant les donnés observés.
L'étape M : les valeurs estimées courantes sont utilisées paur maximiser la probabilité de toutes les données.
Imputation par donneur (*hot-deck*) On utilise des exemples complets et proches comme "donneurs" des donnés manquantes.

Imputation par k-ppv .
On cherche les k-ppv de la donnée (sans l'attribut

Plan

1. Un peu d'ingénierie des données
   - Acquisition et compréhension
2. Normaliser les données
   - Prétraitements
3. Réduire le nombre des attributs
   - Éliminer les attributs inutiles
   - Remplacer deux attributs corrélés par un seul
   - Utiliser l'Analyse en Composantes Principales et d'autres méthodes
   - Exploration

Plan

Plan

Plan

Plan

Visualisation, essai d'une méthode générique et bouclage avec l'étape précédente

Algorithmes d'apprentissage supervisé : vocabulaire

- Espace des hypothèses $\mathcal{H}$. Tous les AD
- Hypothèse h produite par un algorithme $\mathcal{A}$. Un AD
- Généralisation (induction). Régression et Classification.
- Exemple étiqueté par un professeur (un expert, un oracle,. . .).
- Ensemble d'apprentissage S : tous les exemples.
- Ensemble d'entraînement E : données pour l'algorithme $\mathcal{A}$.
- Ensemble de validation V : réglage de l'algorithme $\mathcal{A}$.
- Ensemble de test : estimation finale de la qualité de $\mathcal{A}$ après réglage.
- Erreur empirique (ou apparente) : mesurée sur l'ensemble d'entraînement S.
- Erreur réelle : estimée sur l'ensemble de test T.

Plan

1. Un peu d'ingénierie des données
   - Acquisition et compréhension
2. Normaliser les données
   - Prétraitements
3. Réduire le nombre des attributs
   - Éliminer les attributs inutiles
   - Remplacer deux attributs corrélés par un seul
   - Utiliser l'Analyse en Composantes Principales et d'autres méthodes
   - Exploration

Mesure de l'erreur

Hypothèses statistiques sur les exemples

- Les données sont stationnaires dans le temps.
- Les exemples sont statistiquement indépendants.
- Les exemples sont alors dits i.i.d. : indépendants et identiquement distribués.

Mesure de l'erreur sur un ensemble de test

Régression Moyenne des distances entre la courbe prédite et les exemples de test.

Classification : Matrice de confusion $\left(M_{ij}\right)$: Nombre d'exemples de test classés j de vraie classe j.
Taux d'erreur (apparente) : normaliser la somme de la diagonale

Mesure de l'erreur pour deux classes

En ligne : la classe estimée. En colonne, la classe réelle.

	+ (P)	− (N)
+	**Vrais positifs** (VP)	Faux positifs (FP)
−	Faux négatifs (FN)	**Vrais négatifs** (VN)

- *Taux de bonne prédiction (accuracy)* ou *Précision* ou Estimation du taux d'erreur réelle $\widehat{R}_{\text{Réel}}(h)$ ou *Taux d'erreur apparente* ou *Taux d'erreur empirique* :

$$TEA = \frac{VP + VN}{P + N}$$

- Parfois une pondération, si les erreurs n'ont pas les mêmes conséquences.

Quelques autres façons de mesurer l'erreur

- *Rappel* : $\frac{VP}{VP+FN}$
- *Précision* : $\frac{VP}{VP+FP}$
- *sensibilité* ou Taux de VP : taux_VP = $\frac{VP}{VP+FN}$
- *spécificité* : ou taux de VN : taux_VN = $\frac{VN}{VN+FP}$
- F_β_*mesure*, moyenne harmonique du rappel et de la précision :

$$\frac{(1+\beta^2)\cdot rappel \cdot précision}{\beta^2 \cdot rappel + précision} \qquad \beta > 0$$

- F_*mesure* pour $\beta = 1$:

$$\frac{2 \cdot rappel \cdot précision}{rappel + précision}$$

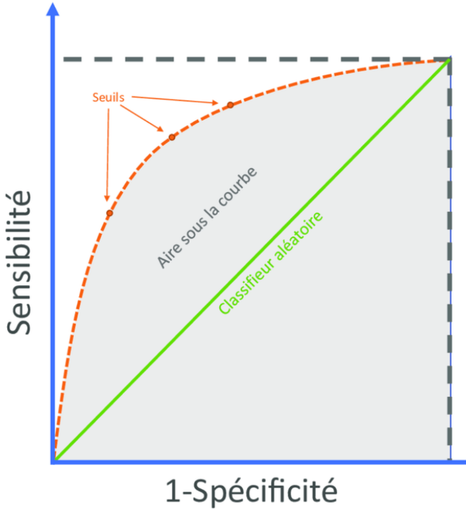
Courbe ROC et critère AUC

sensibilité : taux_VP = $\frac{VP}{VP+FN}$
spécificité : taux_VN = $\frac{VN}{VN+FP}$

Quand le classificateur fournit une valeur continue (par exemple, la régression logistique) on doit définir un seuil de classification ou de décision.

Valeur supérieure à ce seuil : classe 1.
Valeur inférieure à ce seuil : classe 2.

Le mieux n'est pas toujours 0.5.



Intervalle de confiance sur le taux d'erreur.

On mesure $E = FP + FN$ erreurs sur les $T = VP + FP + FN + VN$ exemples de test. On note le taux d'erreur apparent TEA ou $\widehat{R}_{\text{Réel}}(h) = T/E$.

Intervalle de confiance de TEA à x % : $\left[TEA \;\pm\; \zeta(x) \sqrt{\dfrac{TEA(1-TEA)}{T}} \;\right]$.

x	50 %	68 %	80 %	90 %	95 %	98 %	99 %
$\zeta(x)$	0.67	1.00	1.28	1.64	1.96	2.33	2.58

Par exemple, pour $t = 300$ et $VP + VN = 15$, on a $\widehat{R}_{\text{Réel}}(h) = 0.2$ et l'intervalle de confiance à 95 % de $TEA =$ vaut :

$$\left[0.2 \pm 1.96\sqrt{\tfrac{0.2(1-0.2)}{300}}\right] \approx [0.25, 0.15]$$

La probabilité que $R_{\text{Réel}}(h)$ soit dans cet intervalle est supérieure à 95 %.
Avec la même valeur de TEA, mais avec $T = 1000$: $[0.225, 0.175]$

Validation croisée

Entrée: Un ensemble $\mathcal{S}$ d'apprentissage
 Diviser $\mathcal{S}$ en N sous-ensembles disjoints $\mathcal{S}_1, \cdots, \mathcal{S}_N$
 $i \leftarrow 1$
 tant que $i < N$ **faire**
  $E \leftarrow \mathcal{S}_i$; $T \leftarrow \mathcal{S} - \mathcal{S}_i$; Entraîner $\mathcal{A}$ sur E
  Mesurer l'erreur apparente e_i sur T ; $i \leftarrow i + 1$
 fin tant que
 Faire la moyenne e des e_i de 1 à N
Sortie: e est une estimation robuste de l'erreur réelle

Pour plus de deux classes

Certaines méthodes fonctionnent directement avec un nombre quelconque de classes.
Par exemple, les Arbres de décision, les k-ppv, les réseaux connexionnistes.
D'autres non, comme les Classificateurs linéaires ou les SVM.

Trois solutions simples

- Une classe contre toutes les autres. Attention aux probabilités a priori !
  Nombre d'entraînements : C
- Une classe contre une autre. Nombre d'entraînements : $C\frac{1}{2}C(C-1)$
- Classification hiérarchique

Error-Correcting Output-Coding (ECOC)

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
Classe 1	0	0	0	0	0	0	0
Classe 2	0	1	1	0	0	1	1
Classe 3	0	1	1	1	1	0	0
Classe 4	1	0	1	1	0	1	0
Classe 5	1	1	0	1	0	0	1

- Ici, 5 classes et 7 classificateurs.
- Chaque classificateur est entraîné sur les classes notés 1 dans sa colonne.
- Chaque classe est associée à un *code correcteur d'erreurs* de 7 bits

ECOC

Test

Test d'un exemple à classer : une réponse de chacun des 7 classificateurs, par exemple (0 1 0 1 0 0 1).

Décodage

Distance de Hamming aux codes des classes :

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	Distance
	0	1	0	1	0	0	1	
Classe 1	0	0	0	0	0	0	0	3
Classe 2	0	1	1	0	0	1	1	3
Classe 3	0	1	1	1	1	0	0	3
Classe 4	1	0	1	1	0	1	0	5
Classe 5	1	1	0	1	0	0	1	1

Conclusion : la Classe 5 est la plus vraisemblable.

Plan

Analyse de la variance

- **Hypothèse** (trop) forte : On suppose que les scores de résultat sont produits par une distribution gaussienne.
- Dans ce cas, les statistiques classiques peuvent être employées.
- ANOVA : analyse de la variance. Vraisemblance d'un résultat.
- Si une méthode a résultat très bon et toutes les autres un résultat moyen, ANOVA aura tendance à rejeter la "bonne" méthode.

Tests de rang

- K algorithmes à comparer sur N jeux de test.
- Sur chaque test, le meilleur algorithme obtient le rang 1, le deuxième le rang 2, etc.
- Soit r_i^j le rang du j-ème algorithme sur le jeu de données i parmi N. Le test de Friedman compare les rangs moyens $R_j = \frac{1}{N} \sum_i r_i^j$ des algorithmes.
- Si tous les algorithmes étaient équivalents (hypothèse *nulle*), leurs rangs devraient suivre la *statistique de Friedman* :

$$\chi_F^2 = \frac{12\,N}{K(K+1)} \left[\sum_j R_j^2 - \frac{K(K+1)^2}{4} \right]$$

- χ^2 avec $K-1$ degrés de liberté, pour N et K suffisamment grands ($N > 10$ et $K > 5$).
- On mesure l'écart à cette statistique.

Test de Cochran

- On compare K algorithmes.
- On note
  - G_i le nombre d'exemples de l'ensemble de test $\mathcal{T}$ qui sont correctement classés par l'algorithme i
  - $T = \sum_{i=1}^{M} G_i$.
  - K_i le nombre de modèles ayant correctement classé le i^e exemple de $\mathcal{T}$
- Sous l'hypothèse que les performances des K classifieurs ne diffèrent pas significativement, la quantité

$$Q = (K-1) \frac{K \sum_{i=1}^{K} G_i^2 - T^2}{KT - \sum_{i} K_i^2}$$

suit approximativement une loi d χ^2 à $K-1$ degrés de liberté.

- On teste l'écart à cette statistique.