Apprentissage automatique

Concepts fondamentaux - prétraitement de données

PRÉTRAITEMENT DE DONNÉES

Sujets: prétraitement de données

- Jusqu'à maintenant, on a supposé que les entrées $\mathbf{x}$ prenaient la forme d'un vecteur dans $\mathbb{R}^D$
  - chaque élément x_i de $\mathbf{x}$ est un scalaire réel

- Quoi faire lorsque ce n'est pas le cas ?

- Une phase de prétraitement doit être suivie afin de convertir les entrées en vecteur

PRÉTRAITEMENT DE DONNÉES

Sujets: donnée catégorique

- Une **donnée catégorique** prend une valeur parmi un ensemble de symboles fini Ω

- Exemples :
  - le sexe d'une personne : $\Omega=\{\text{'femme'}, \text{'homme'}\}$
  - la réponse à un sondage : $\Omega=\{\text{'oui'}, \text{'non'}, \text{'peut-être'}\}$
  - un mot : $\Omega=\{\text{'le'}, \text{'la'}, \ldots\}$
  - etc.

PRÉTRAITEMENT DE DONNÉES

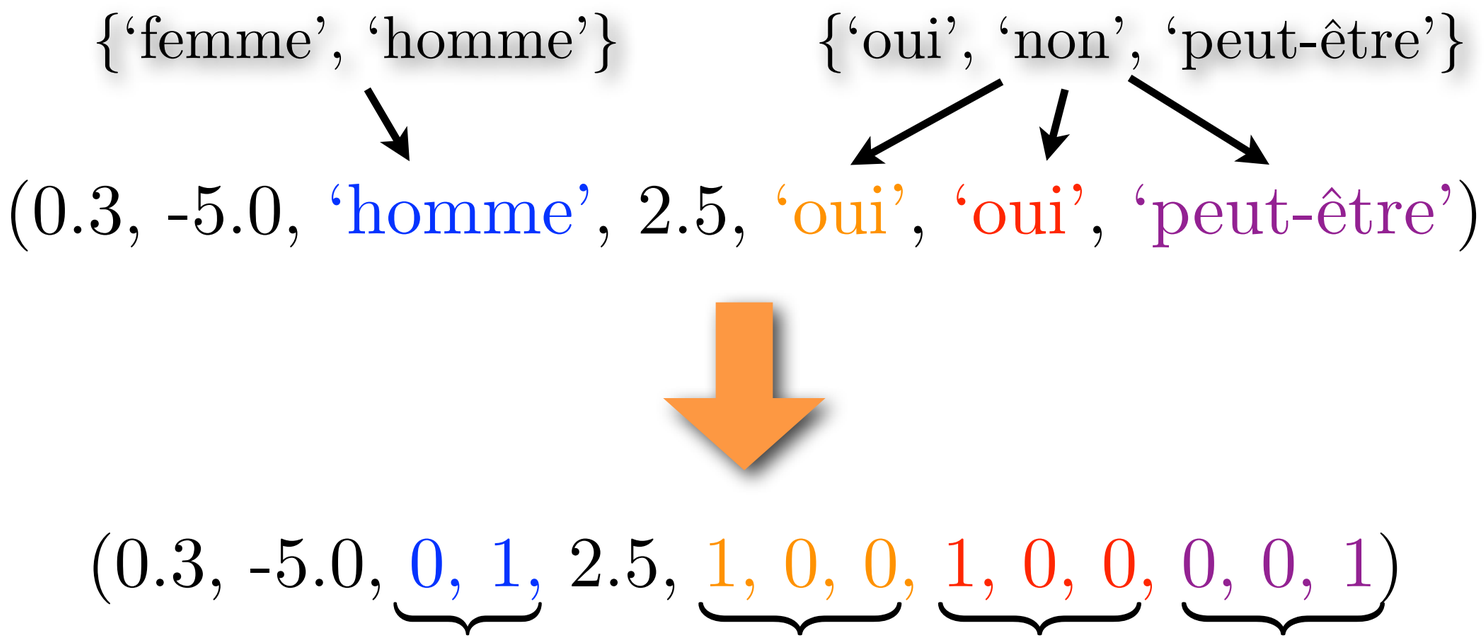
Sujets: représentation *one-hot*

- On convertit sous une forme vectorielle appelée ***one-hot***
  - crée un vecteur de taille $|\Omega|$ et rempli de 0s
  - on associe chaque position dans le vecteur à un symbole différent
  - on assigne à 1 l'élément à la position du symbole observé
- Exemple :
  - réponse au sondage observée : 'non'
  - positions assignées dans cet ordre : 'oui','non','peut-être'
  - vecteur *one-hot* généré sera : $(0,1,0)$

PRÉTRAITEMENT DE DONNÉES

Sujets: représentation *one-hot*

- Exemple :



PRÉTRAITEMENT DE DONNÉES

Sujets: donnée manquante

- Si une donnée catégorique est manquante, une approche simple est de laisser tous les éléments du vecteur à 0
- Pour une donnée réelle, on ne peut simplement laisser à 0
  - si $x_i = 0$, est-ce que la donnée originale était 0 ou était manquante ?
- On ajoute plutôt une entrée binaire indiquant les valeurs manquantes (seulement pour les x_i qui peuvent manquer)
  - $(0.5, \text{-}0.8, \textbf{?}, 2.1)$ devient $(0.5, \text{-}0.8, \mathbf{0}, \mathbf{1}, 2.1)$
  - $(1.1, 0.2, \mathbf{3.8}, \text{-}0.7)$ devient $(0.5, \text{-}0.8, \mathbf{3.8}, \mathbf{0}, 2.1)$

PRÉTRAITEMENT DE DONNÉES

Sujets: normalisation

- Il est souvent préférable de normaliser les données réelles, afin qu'elle prennent des valeurs «proches» de 0
  - des valeurs trop élevées pourraient créer des instabilités numériques

- Suffit de normaliser les données, en soustrayant la moyenne et divisant par l'écart-type
  - cette opération est appliquée individuellement pour chaque x_i de $\mathbf{x}$, c'est-à-dire chaque colonne de la matrice $\mathbf{X}$

Apprentissage automatique

Concepts fondamentaux - comparaison d'algorithmes

COMPARAISON D'ALGORITHMES

Sujets: comparaison d'algorithme

- Comment déterminer si un algorithme A est vraiment meilleur qu'un algorithme B ?
  - on regarde l'erreur de test, i.e. notre mesure de leur performance de généralisation

- Est-ce que l'algorithme ayant l'erreur de test la plus basse est nécessairement (strictement) meilleur ?
  - peut-être que l'algorithme a «été chanceux» et que la conclusion aurait été différente sur un autre ensemble de test

COMPARAISON D'ALGORITHMES

Sujets: intervalle de confiance

- En plus de rapporter l'erreur moyenne sur l'ensemble de test, on rapporte également un **intervalle de confiance**
  - e.g. l'intervalle tel que la probabilité qu'il contienne la vraie performance de généralisation est de 95%

- Si les intervalles de deux algorithmes ne se chevauchent pas, ils ont probablement des performances différentes

COMPARAISON D'ALGORITHMES

Sujets: intervalle de confiance, *standard error*

- Comment calculer cet intervalle ?
  - on utilise le fait que, si l'ensemble de test est assez grand, l'erreur moyenne suit une distribution qui est bien approximée par une loi Normale ayant la moyenne μ et variance σ^2_{ste} suivante :

$$\mu = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(\mathbf{x}_n, t_n) \in \mathcal{D}_{\text{test}}} L(t_n, y(\mathbf{x}_n))$$

$$\sigma^2_{\text{ste}} = \frac{\sigma^2}{|\mathcal{D}_{\text{test}}|} \qquad \sigma^2 = \frac{1}{|\mathcal{D}_{\text{test}}| - 1} \sum_{(\mathbf{x}_n, t_n) \in \mathcal{D}_{\text{test}}} (L(t_n, y(\mathbf{x}_n)) - \mu)^2$$

COMPARAISON D'ALGORITHMES

Sujets: intervalle de confiance, *standard error*

- Comment calculer cet intervalle ?
  - on utilise le fait que, si l'ensemble de test est assez grand, l'erreur moyenne suit une distribution qui est bien approximée par une loi Normale ayant la moyenne μ et variance σ^2_{ste} suivante :

$$\mu = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(\mathbf{x}_n, t_n) \in \mathcal{D}_{\text{test}}} L(t_n, y(\mathbf{x}_n))$$

standard error

$$\sigma^2_{\text{ste}} = \frac{\sigma^2}{|\mathcal{D}_{\text{test}}|} \qquad \sigma^2 = \frac{1}{|\mathcal{D}_{\text{test}}| - 1} \sum_{(\mathbf{x}_n, t_n) \in \mathcal{D}_{\text{test}}} (L(t_n, y(\mathbf{x}_n)) - \mu)^2$$

COMPARAISON D'ALGORITHMES

Sujets: intervalle de confiance

- Comment calculer cet intervalle ?
  - ainsi, la probabilité que la vraie erreur de généralisation se trouve dans l'intervalle suivant est de 95% :

$$\mu \pm 1.96\sigma_{\text{ste}}$$