SPLEX
Statistiques pour la classification et fouille de données en génomique

Classification bayésienne
Réseaux bayésiens

Pierre-Henri WUILLEMIN

DEcision, Système Intelligent et Recherche opérationnelle
LIP6
pierre-henri.wuillemin@lip6.fr
http://webia.lip6.fr/~phw/splex

Approche probabiliste de la classification

Soient, à nouveau, deux v.a. X (de dimension d) discrète et Y (de dimension 1) discrète (*pas forcément binaire*).
Sur la base Π_a, on peut estimer les probabilités par des fréquences pour $P(X, Y)$.
Soit x une instantiation de X, on cherche sa classe y (valeur de Y).

1 Maximum de vraisemblance

$$y = \arg\max_{y_i} P(x \mid y_i)$$

2 Maximum a posteriori

$$y = \arg\max_{y_i} P(y_i \mid x)$$

D'après la règle de Bayes, $P(Y \mid X) \propto P(X \mid Y) \cdot P(Y)$, on comprend que l'intérêt du MAP est de prendre en compte un *a priori* sur la fréquence de chaque classe.

Il peut être difficile d'obtenir ces distributions.
Particulièrement : $P(X \mid Y)$ peut demander beaucoup d'observation ! !

Classifieur Bayésien Naïf

Hypothèse du classifieur bayésien naïf

On supposera que, $\forall k \neq l, X^k \perp\!\!\!\perp X^l \mid Y$

Cette hypothèse est très forte. Elle a peu de chance de s'avouer exacte dans un cas réel. Néanmoins cette approximation donne des résultats souvent satisfaisants.

Alors, le calcul du MAP s'écrit :

3 Maximum a posteriori

$$y = \arg\max_{y_i} \left(P(y_i) \cdot \prod_{k=1}^{d} P(x^k \mid y_i) \right)$$

Cette hypothèse permet donc de simplifier fortement les calculs nécessaires pour estimer le MAP.

Rapides rappels : indépendances (conditionnelles)

Soit X, Y, Z trois variables aléatoires (ou groupes de variables)

➥ si $P(Y) > 0$, $P(X|Y) = \dfrac{P(X,Y)}{P(Y)}$

$$\Longrightarrow P(X,Y) = P(X|Y)P(Y)$$

$$\begin{aligned}\Longrightarrow P(X,Y,Z) &= P(X|Y,Z)P(Y,Z)\\ &= P(X|Y,Z)P(Y|Z)P(Z)\end{aligned}$$

Indépendance marginale

$$\begin{aligned} X \perp\!\!\!\perp Y \text{ si et seulement si } & P(X,Y) = P(X)P(Y) \\ \text{si et seulement si } & P(X \mid Y) = P(X)\end{aligned}$$

Indépendance conditionnelle

$$\begin{aligned} X \perp\!\!\!\perp Y | Z \text{ si et seulement si } & P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z) \\ \text{si et seulement si } & P(X \mid Y,Z) = P(X \mid Z)\end{aligned}$$

Modèle probabiliste complexe

La représentation probabiliste d'un système est caractérisé par un univers Ω où chaque $\omega \in \Omega$ est un état du système.
Exemple : Un dé est caractérisé par $\Omega_{\text{dé}} = \{1,2,3,4,5,6\}$.

Un système complexe est caractérisé par un univers Ω de grande taille.
Exemple : $\Omega_{\texttt{voiture}}$?

Définition (Modèle décomposable)

Un modèle probabiliste (sur Ω)est décomposable lorsqu'il existe une famille $\mathfrak{X} = (X_i)_{i<n}$ de variables aléatoires sur Ω telle que chaque $\omega \in \Omega$ est caractérisé de manière unique par les valeurs $(X_i(\omega))_{i<n}$.

Exemple : $\omega_{\texttt{voiture}} =$ `(Vitesse=55, Phare=Eteint, Pneu_gauche=dégonflé,...)`.

Modèle probabiliste (2)

Modèle probabiliste complexe

Dans un modèle décomposable, une probabilité sur Ω sera donc représentée par une loi jointe des variables de $\mathfrak{X}$.

$$\forall \omega \in \Omega, p(\omega) = p\left(X_1 = X_1(\omega), X_2 = X_2(\omega), \cdots, X_n = X_n(\omega)\right)$$

Explosion combinatoire : Si toutes les variables sont binaires, un système factorisé en n variables nécessitent $\approx 2^n$ valeurs !

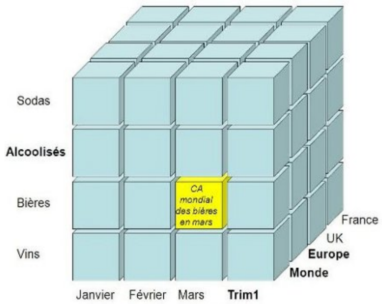
La factorisation peut-elle permettre d'améliorer la compacité ? Grâce à l'indépendance conditionnelle ! !

2^3 $\quad p(X,Y,Z) = p(X)\cdot p(Y \mid X)\cdot p(Z \mid X,Y)$ $\quad 2 + 2^2 + 2^3$

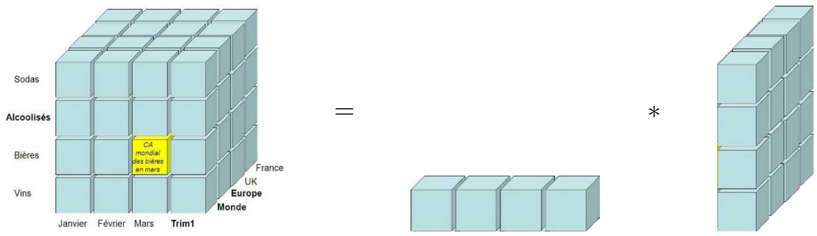
Avec $X \perp\!\!\!\perp Y$ et $Z \perp\!\!\!\perp X, Y$:
2^3 $\quad p(X,Y,Z) = p(X)\cdot p(Y)\cdot p(Z)$ $\quad 2 + 2 + 2$

Modèles complexes factorisable

		Janvier	Février	Mars	Trim1
France	Bières	70	70	80	220
	Vins	100	110	90	300
	Total	**170**	**180**	**170**	**520**
UK	Bières	250	220	240	710
	Vins	50	40	60	150
	Total	**300**	**260**	**300**	**860**
Total Europe		***470***	***440***	***470***	***1380***



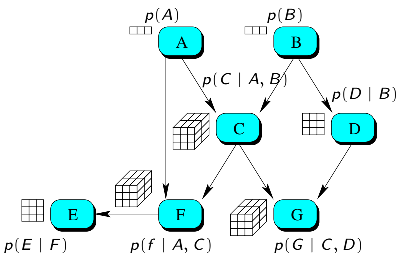
Comment voir dans ce modèle que $Mois \perp\!\!\!\perp \{Pays, Boisson\} | ?$



Réseaux bayésiens

$p(A, B, C, D, E, F, G,) = p(A) \cdot p(B) \cdot P(C \mid A, B) \cdot P(D \mid B) \cdot P(E \mid F) \cdot P(F \mid A, C) \cdot P(G \mid C, D)$

On utilise une représentation graphique : les parents d'un nœud sont les variables conditionnantes de la décomposition.
Tout se passe comme si l'information était localisée dans les nœuds !



Factorisation dans un BN

$P(X_1, \cdots, X_n) = \prod_i P(X_i \mid \text{parents}_{X_i})$

Inférences dans un BN

- $P(A|G = 1)$
- $P(G|A = 0)$
- $P(E \mid A = 0, G = 1)$
- ...

Classifieur Bayésien Naïf

Hypothèse du classifieur bayésien naïf

On supposera que, $\forall k \neq l, X^k \perp\!\!\!\perp X^l | Y$

Cette hypothèse est très forte. Elle a peu de chance de s'avouer exacte dans un cas réel. Néanmoins cette approximation donne des résultats souvent satisfaisants.

3 ML & MAP pour un *Naive Bayes*

$$\text{ML} : \hat{y} = \arg\max_{y_i} \left(\prod_{k=1}^{d} P(x^k \mid y_i) \right)$$

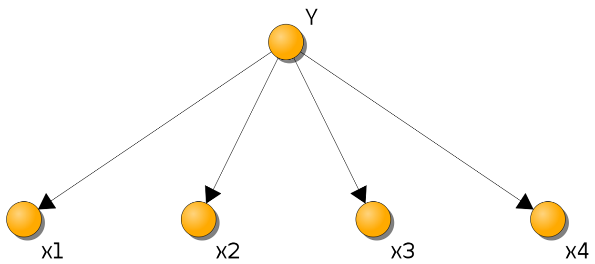
$$\text{MAP} : \hat{y} = \arg\max_{y_i} \left(P(y_i) \cdot \prod_{k=1}^{d} P(x^k \mid y_i) \right)$$

Cette hypothèse permet donc de simplifier fortement les calculs nécessaires pour estimer ML et MAP.

Naive bayes comme un BN

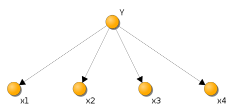
Classifieur bayésien naïf

$\forall k \neq l, X^k \perp\!\!\!\perp X^l \mid Y$ et $P(x, y) = P(y) \cdot \prod_{k=1}^{d} P(x^k \mid y)$



- Estimation des paramètres : trivial (si Π_a sans valeurs manquantes)
- ML : $\prod_{k=1}^{d} P(x^k \mid y)$ ...
- MAP : $P(y \mid x_1, \cdots, x_d)$: inférence dans le BN !

Estimation des paramètres d'un CBN



Soit Π_a une BDD sans valeur manquante de N cas. On note η_C le nombre de cas de la base vérifiant la condition C.

estimation des paramètres

- $P(y) \approx \frac{\eta_{Y=y}}{N} = \frac{\eta_y}{N}$
- $P(x \mid y) \approx \frac{\eta_{(X=x, Y=y)}}{\eta_{Y=y}} = \frac{\eta_{xy}}{\eta_y}$

Ajustement des paramètres (éviter les 0)

- **ajustement de Laplace** $P(x \mid y) \approx \frac{\eta_{xy}+1}{\eta_y+|X|}$
- **a priori de Dirichlet** $P(x \mid y) \approx \frac{\eta_{xy}+\alpha_{xy}}{\eta_y+\alpha_y}$ avec $\alpha_y = \sum_x \alpha_{xy}$
- **actualisation de Ney-Essen**

On retire à tout x une valeur fixe δ et on répartit uniformément la somme collectées.

$D_y = \sum_x \min(\eta_{xy}, \delta)$ et $\forall x, P(x \mid y) = \frac{\eta_{xy} - \min(\eta_{xy}, \delta) + \frac{D_y}{|X|}}{\eta_y}$

Codage de textes et approche *Naive Bayes* (1/3)

Soit X l'ensemble des documents $\{d_i\}_{i=1,\ldots,N}$, chacun des documents étant composé d'une suite $|d|$ mots w_j : $d_i = (w_1, \ldots, w_{|d_i|})$
Nous allons construire un modèle Θ_m pour chaque classe de document. Nous nous appuierons ensuite sur ces modèles pour construire un classifieur de phrase.

1. Donner un cas d'usage classique pour ce type de classifieur. Imaginer un modèle simple pour répondre à ce problème.
2. Calculer la probabilité $p(d)$ d'observer un document d en fonction des $p(w_j)$ (probabilité d'observation d'un mot w_j) lorsque l'on fait l'hypothèse de l'indépendance des tirages des w_j.
3. On introduit maintenant la variable x_i^j qui traduit le nombre d'apparition du mot j dans le document i. On introduit également l'ensemble $D = \{w_1, \ldots, w_{|D|}\}$ contenant tout le vocabulaire utilisé dans un corpus. Écrire $P(d)$ comme une fonction de x_i^j.
4. Pour trouver les paramètres Θ_m, on souhaite maximiser la log-vraisemblance de ces paramètres sur l'ensemble de la classe m de la base documentaire. Montrer que le problème d'optimisation permettant de trouver Θ_m en fonction des $P(w_j|\Theta_m)$ s'écrit :

$$\Theta_m = \arg\max_{\Theta_m} \sum_{i=1}^{|C_m|} \sum_{j=1}^{|D|} x_i^j \log P(w_j|\Theta_m)$$

Codage de textes et approche *Naive Bayes* (2/3)

5. On choisit de construire un modèle dans lequel $\Theta_m = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_j \\ \vdots \\ \theta_{|D|} \end{bmatrix} = \begin{bmatrix} \vdots \\ P(w_j|\Theta_m) \\ \vdots \end{bmatrix}$.

   En introduisant la contrainte $\sum_j \theta_j = 1$, on peut alors montrer que les paramètres optimaux (maximisant la vraisemblance) sont ceux qui satisfont :

$$\theta_j = \frac{\sum_{d_i \in C_m} x_i^j}{\sum_{d_i \in C_m} \sum_{j \in D} x_i^j}$$

   À quoi correspondent chacun des paramètres θ_j ?

   Soit la base de donnée suivante, établir le dictionnaire D puis calculer Θ_1 et Θ_2.

La voiture est au garage.	Le lion est dans la savane.
La voiture est sur la route.	Le lion guette sa proie.
La jeep roule sur la route.	L'antilope est une proie pour le lion.

Codage de textes et approche *Naive Bayes* (3/3)

6. À votre avis, pourquoi cet algorithme s'appelle-t-il *Naive Bayes* ? Intuitivement, pensez-vous qu'il donne de bons résultats ?
7. Calculer et comparer les probabilités $p(\Theta_1|d_i)$ et $p(\Theta_2|d_i)$ pour les documents de la base d'apprentissage.
8. Êtes-vous satisfaits des résultats obtenus ?

   Dans la pratique, on redéfinit les paramètres comme :

$$\theta_j = \frac{\sum_{d_i \in C_m} x_i^j + 1}{\sum_{d_i \in C_m} \sum_{j \in D} x_i^j + |D|}$$

   Expliquer les raisons de ce choix en analysant les résultats précédents et les résultats sur les phrases suivantes :
   - Le monospace roule sur la route
   - Le lion apprécie également les gazelles
9. Quels sont les mots qui *participent* le plus à la classification des phrases ? Quels traitements effectuer pour avoir des *mots-clés* plus pertinents ?

Naive bayes : synthèse

Avantages

- Facile à construire (structure du BN donnée par hypothèse)
- Classification rapide et efficace. Permet un $d > 10000$! ! !
- Malgré une hypothèse forte d'indépendance, Naïve Bayes fait aussi bien et même parfois mieux que des classifieurs plus sophistiqués (Langley et al. 1992).

Défauts

- Hypothèse trop forte qui peut dégrader fortement les résultats du classifieur.

Améliorations du modèle

- **feature joining** et **feature selection** pour diminuer d et rendre compte de *features* très corrélés.
- **relaxation des hypothèses d'indépendances entre features** : rajouter des arcs entre les X_i... mais lesquels ?

Divergence de Kullback-Leibler, Informations mutuelles

Soient P et Q deux distribution sur le même espace X. Comment les comparer ?

Divergence de Kullback-Leibler (entropie relative)

$$D_{KL}(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

Cette mesure s'interprète comme la différence moyenne du nombre de bits nécessaires au codage d'échantillons de P selon que le codage est choisi optimal pour la distribution P ou Q.

Notre problème est ici de tester si une indépendance entre 2 variables est "valide".

Information mutuelle (et conditionellement à Z)

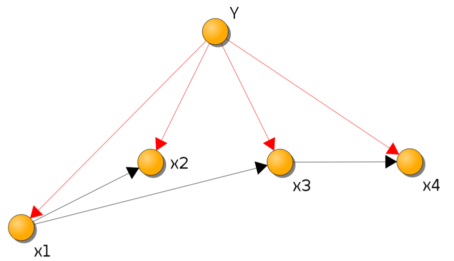
Soit X_1, X_2 et Z trois v.a. L'information mutuelle entre X_1 et X_2 selon P s'écrit :

$$I_P(X_1; X_2) = \sum_{(x_1,x_2)} P(x_1, x_2) \log \frac{P(x_1, x_2)}{P(x_1) \cdot P(x_2)}$$

$$I_P(X_1; X_2 \mid Z) = \sum_{(x_1,x_2,z)} P(x_1, x_2, z) \log \frac{P(x_1, x_2, z)}{P(x_1, z) \cdot P(x_2, z)}$$

TAN = Tree-Augmented Naive Models

Idée : Toute variable X_i peut avoir un parent autre que Y (mais un seul !).

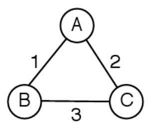


Idées de l'algorithme

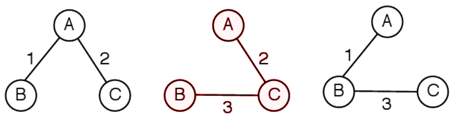
- Quels (X_i,X_j) méritent d'être reliées ? Maximisation de $I_P(X_i; X_j \mid Y)$
- Quels (X_i,X_j) ne méritent pas d'être reliées ?
  $I_P(X_i; X_j \mid Y) < seuil$ ($seuil = \mathbb{E}(I_P(.;. \mid Z)$ par exemple)
- Comment choisir l'arbre dans le graphe résultant ?
  Maximum weighted spanning tree (Kruskal)
- Comment orienter les (X_i,X_j) sélectionnées ? Maximisation de $I_P(X_i; Y)$

Encore une aparté : Algorithme de Kruskal

Soit un graphe non orienté, à arcs valués :



Dont voici les arbres couvrants :



Comment trouver l'arbre couvrant de poids maximal ?

Kruskal's algorithm (*maximum weighted spanning tree*)

1. Sort the edges of $\mathbb{G}$ into decreasing order by weight. Let T be the set of edges comprising the maximum weight spanning tree. Set $T = \emptyset$.
2. Add the first edge to T.
3. Add the next edge to T if and only if it does not form a cycle in T. If there are no remaining edges exit and report $\mathbb{G}$ to be disconnected.
4. If T has $n-1$ edges (where n is the number of vertices in $\mathbb{G}$) stop and output T. Otherwise go to step 3.

algorithme Construit-TAN

Construit-TAN

1. $\forall i < j$, calculer $I_P(X_i; X_j \mid Y)$
2. Valuer les arcs du graphe non-orienté complet des (X_i) par les $I_P(X_i; X_j \mid Y)$
3. Calculer $\mathbb{E}(I_P(.;. \mid Z) = \frac{\sum_{i<j} I_P(X_i;X_j|Z)}{d(d-1)/2}$
4. Supprimer les arcs de valuation $< \mathbb{E}(I_P(.;. \mid Z)$
5. Appliquer Kruskal sur ce graphe pour obtenir une forêt couvrante de poids max.
6. Sur chaque partie connexe C de cette forêt, chercher $X_{root} = \arg\max_{X \in C} I_P(X; Y)$.
7. Utiliser ce X_{root} comme racine pour l'orientation de C.
8. Rajouter Y dans le graphe, père de tous les X.
9. Apprendre les paramètres du BN.

Classifieur par BN

Principe général

- Apprentissage de structure d'un BN à partir de la base $X + Y$.
- Utilisation de l'inférence pour calculer $P(Y \mid x)$
- ⚠ uniquement besoin de la couverture de Markov !!!