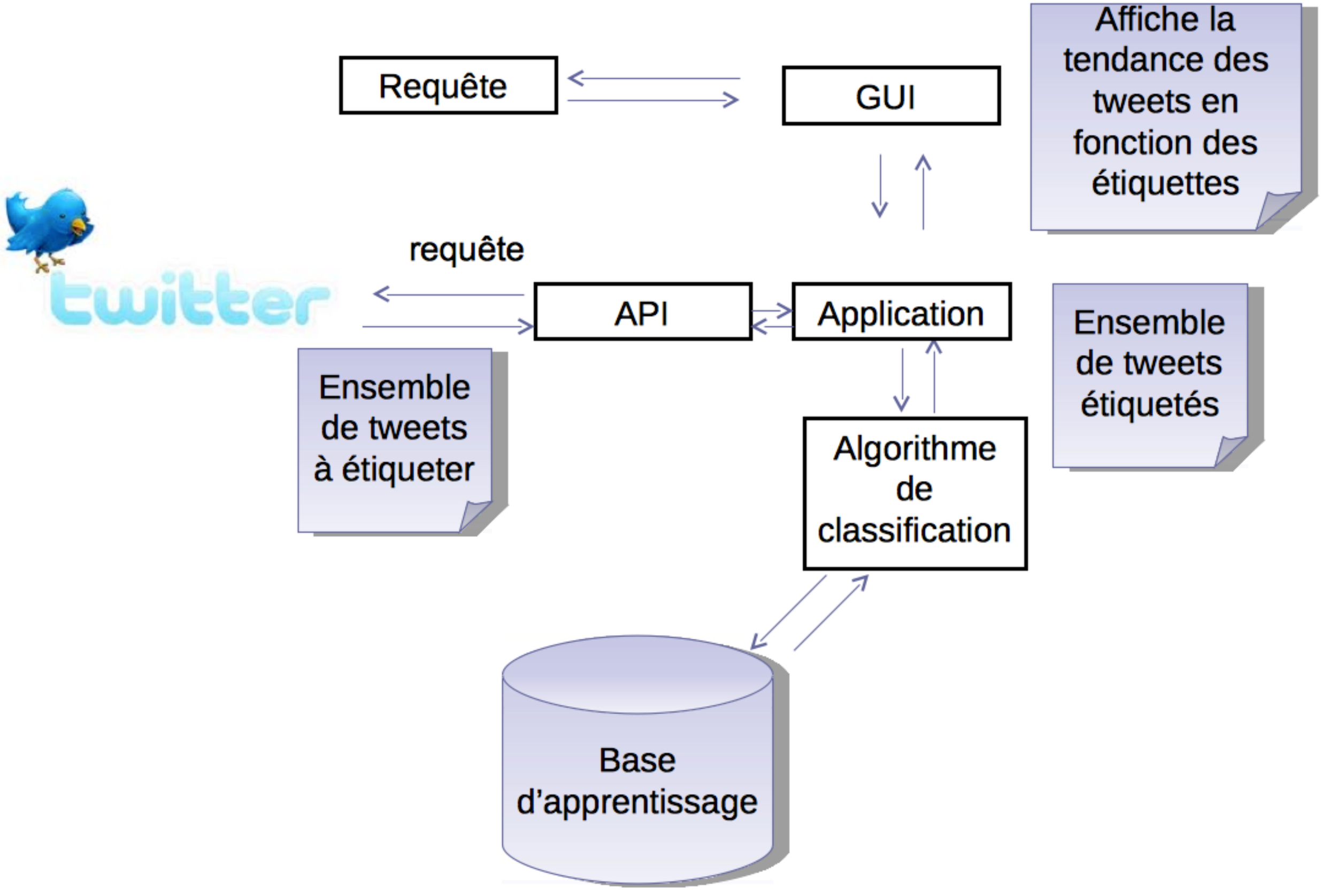
PJE : Analyse de comportements avec Twitter
Classification supervisée

Arnaud Liefooghe
arnaud.liefooghe@univ-lille1.fr

Master 1 Informatique – PJE2 – 2015-16
B. Derbel – L. Jourdan – A. Liefooghe


Requête
GUI
Affiche la tendance des tweets en fonction des étiquettes
requête
API
Application
Ensemble de tweets étiquetés
Ensemble de tweets à étiqueter
Algorithme de classification
Base d'apprentissage

Agenda

- Partie 1: SVN & API Twitter (B. Derbel)
- Partie 2: Classif mots clés & kNN (L.Jourdan)
- Partie 3 (A. Liefooghe)
  - Classification bayésienne & analyse exp.
- Partie 4: Interface utilisateur (B. Derbel)
- 16/12: Rapport et soutenance

Agenda

- 03/11: C-TD & TP
  - Classification bayésienne
- 10/11: C-TD & TP
  - Classification bayésienne (variantes)
- 17/11: C-TD & TP
  - Analyse expérimentale

Contenu

- Classification supervisée
- Classification bayésienne
- Exemple : Classification de texte

Sources et références

- Apprentissage artificiel : Concepts et algorithmes, *A. Cornuéjols et L. Miclet*
- Fouille de données, *F. Decomité*
- Fouille de données, *P. Preux*
- Apprentissage à partir d'exemples, *F. Denis et R. Gilleron*

Classification supervisée

Classification supervisée *vs.* non-supervisée

- Clustering
  - Le but est de regrouper des objets similaires (pas d'attribut particulier)
- Fouille supervisée
  - Il existe un attribut particulier : la classe
  - Le but est de «deviner» la classe d'un exemple en ne connaissant que la valeur de ses autres attributs

Classification supervisée *vs.* non-supervisée

- Clustering
  - On conserve toutes les données
- Fouille supervisée
  - On modélise : les données servent à construire le modèle et sont (généralement) oubliées ensuite

2 types de classifieurs

- Classifieurs qui utilisent directement les exemples pour prédire la classe d'une donnée
- Classifieurs pour lesquels on a d'abord construit un modèle et qui, ensuite, utilisent ce modèle pour effectuer leur classification/prédiction

Problèmes

- Méthode d'induction du classifieur ?
- Comment utiliser le classifieur obtenu ?
- Comment évaluer la qualité du classifieur obtenu : taux d'erreur (ou de succès) ?
- Comment traiter les attributs manquants dans le jeu d'apprentissage ? dans une donnée à classer ?
- Comment estimer la tolérance au bruit ?
  Le bruit concerne ici la valeur des attributs de l'exemple avec lequel on construit le classifieur

Vocabulaire

- Classification : prévoir une classe discrète
- Prédiction : prévoir une valeur continue (degré de confiance)

Principe

- Une instance = une suite de valeurs d'attributs et une classe (a_1, a_2, ..., a_n, c)
- A l'aide d'un ensemble d'exemples, on veut construire un modèle des données (classifieur, prédicteur, ...)
- On demande à ce classifieur de trouver la classe de nouveaux exemples

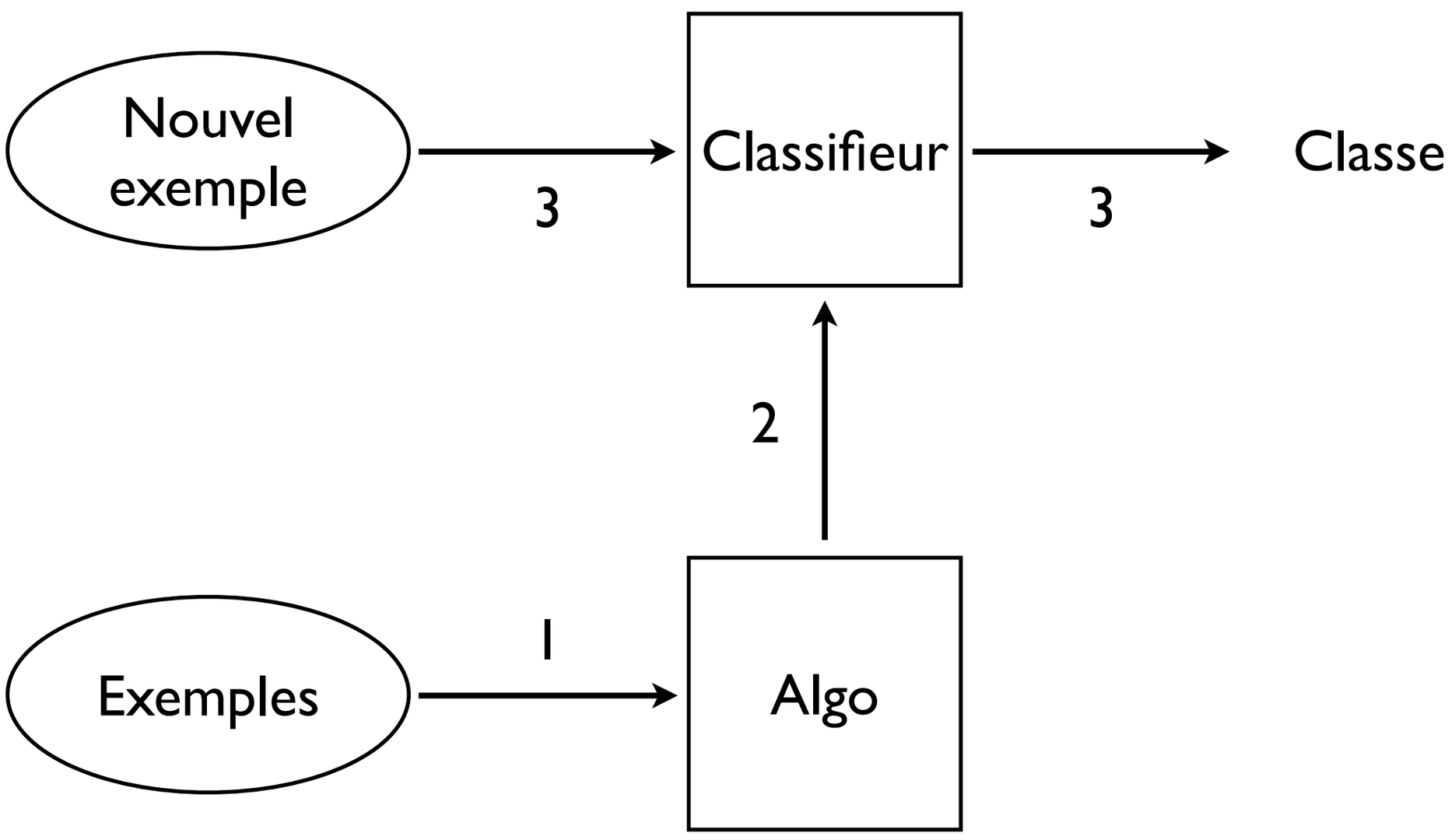
Principe

- Modèle : découvrir la structure des données
- Quels sont les attributs importants pour deviner une classe ?

Mode opératoire

- Etape 1 : Construction du modèle à partir de l'ensemble d'apprentissage (training set)
- Etape 2 : Evaluation de la qualité/précision du classifieur
- Etape 3 : Utilisation du modèle pour la classification de nouvelles instances

Schéma



1 - Construction du modèle

- Chaque instance est supposée appartenir à une classe prédéfinie
- La classe d'une instance est déterminée par l'attribut «classe»
- L'ensemble des instances d'apprentissage est utilisé dans la construction du modèle
- Le modèle est représenté par des règles de classification, arbres de décision, formules mathématiques, ...

2 - Évaluation du modèle

- Estimer le taux d'erreur
  - La classe connue d'une instance test est comparée avec le résultat du modèle
  - Taux d'erreur = pourcentage de tests incorrectement classés par le modèle

3 - Utilisation du modèle

- Classification de nouvelles instances (inconnues)

Domaines d'application

- Délivrance de crédit
- Diagnostic médical
- Prédiction du cours d'une action
- Optimisation d'un envoi de courrier
- ...

La classification dans le processus du data-mining

- Collecte, préparation des données
- Données d'apprentissage
- Évaluation, validation

Apprentissage

- On manipule :
  - Des données
  - Des hypothèses
  - On veut trouver la meilleure hypothèse en fonction des données disponibles

Quelques méthodes de classification

- Arbres de décision : minimiser l'erreur de classification
- Classification bayésienne (classifieur naïf, réseaux bayésiens) : hypothèse la plus probable
- Réseaux de neurones : minimiser l'erreur quadratique
- ...

Qu'est-ce qu'un bon classifieur ?

- Intuition : classifieurs binaires discrets
- 4 cas
- Vrai positif : exp. positif classé positif
- Faux négatif : exp. positif classé négatif
- Vrai négatif : exp. négatif classé négatif
- Faux positif : exp. positif classé négatif

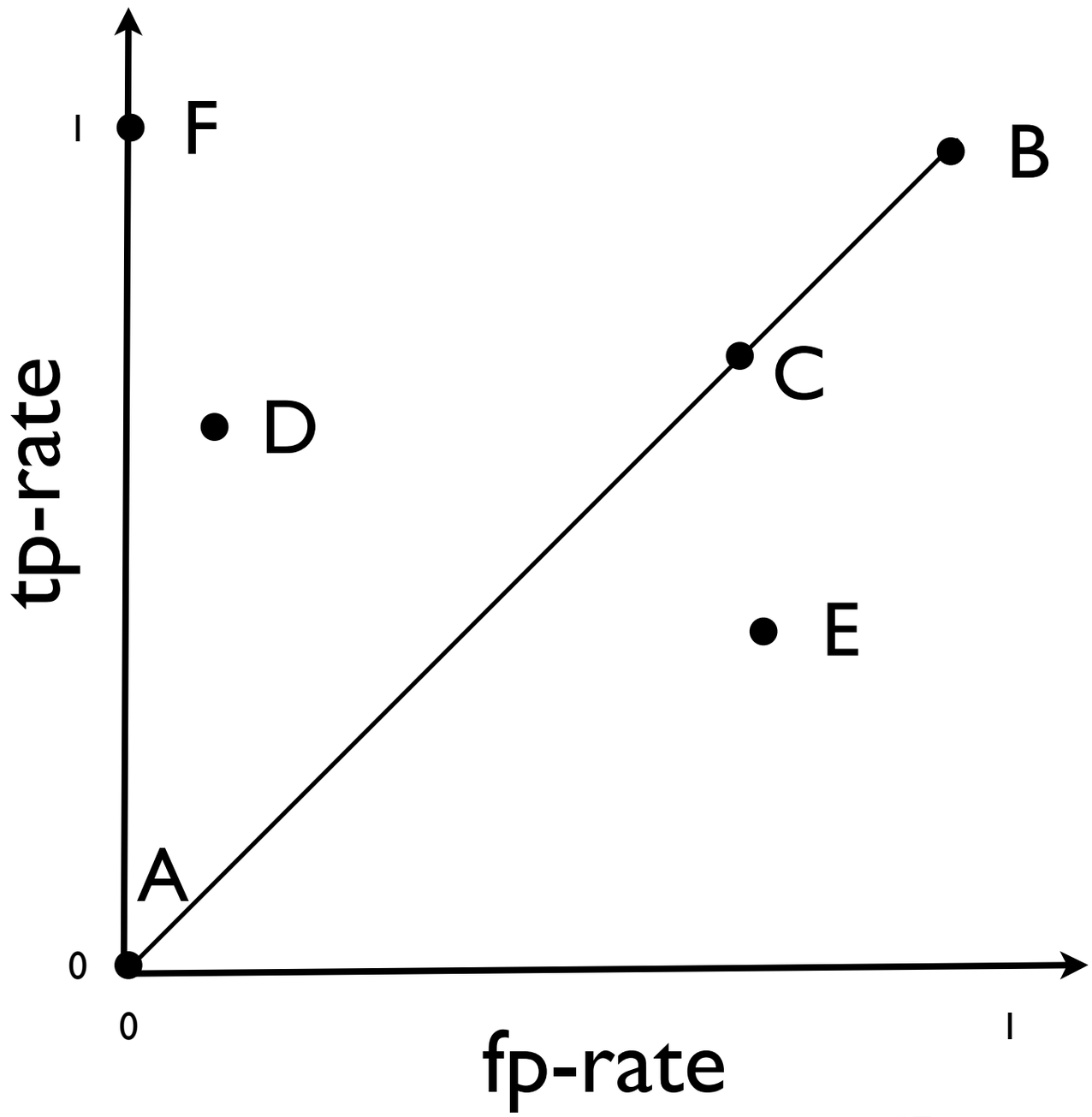
Matrice de confusion

vrai classe → ↓ classé	positif	négatif
positif	VP	FP
négatif	FN	VN
total	P	N

Taux

- Proportion de positifs bien classés
  - tp-rate = VP / P
- Proportion de négatifs mal classés
  - fp-rate = FP / N

Figure



- A = tous négatifs
- B = tous positifs
- C = k% positifs
- D = conservateur
- E < aléatoire
- F = class. idéal

Classification bayésienne

Principe

- On doit inférer (deviner) des quantités gouvernées (décrites) par des probabilités : on veut se servir de ces probabilités pour guider l'inférence
- Cadre plus général que la classification

Classification bayésienne

- À chaque hypothèse :
  - On associe une probabilité (probabilité d'être la solution)
  - L'observation d'une (ou de plusieurs) instances peut modifier cette probabilité
  - On peut parler de l'hypothèse la plus probable, au vu des instances

Buts (possibles)

- Formaliser les méthodes et les intuitions
- Préciser la notion de ‘plus probable’
- Nouveaux algorithmes d’apprentissage
- Analyse d’autres algorithmes ne manipulant pas explicitement des probabilités

Classification bayésienne

- Approche probabiliste
- Basée sur les probabilités conditionnelles (et la règle de Bayes)
- Connaissances *a priori*
- Prévision du futur à partir du passé
- Suppose l'indépendance des attributs

Classification bayésienne

- Différente de l'approche basée sur les fréquences !
- Fréquences : on estime la probabilité d'occurrence d'un événement
- Bayésienne : on estime la probabilité d'occurrence d'un événement sachant qu'une hypothèse préliminaire est vérifiée (connaissance)

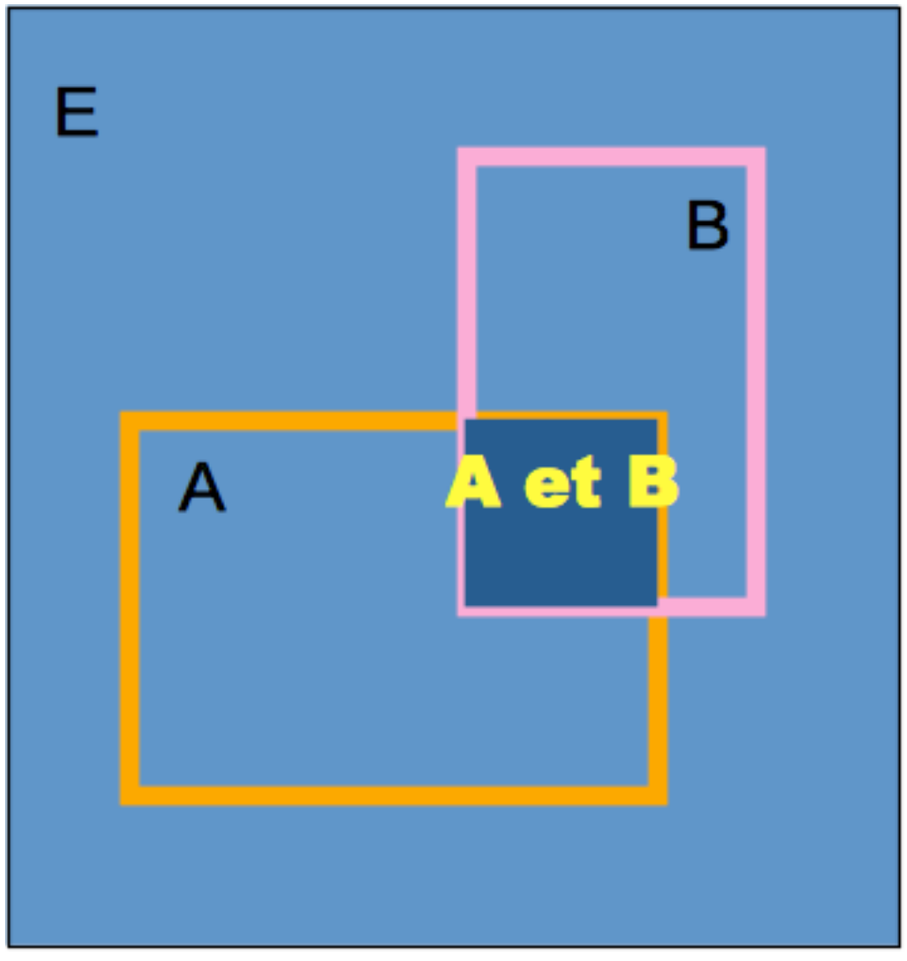
Probabilités

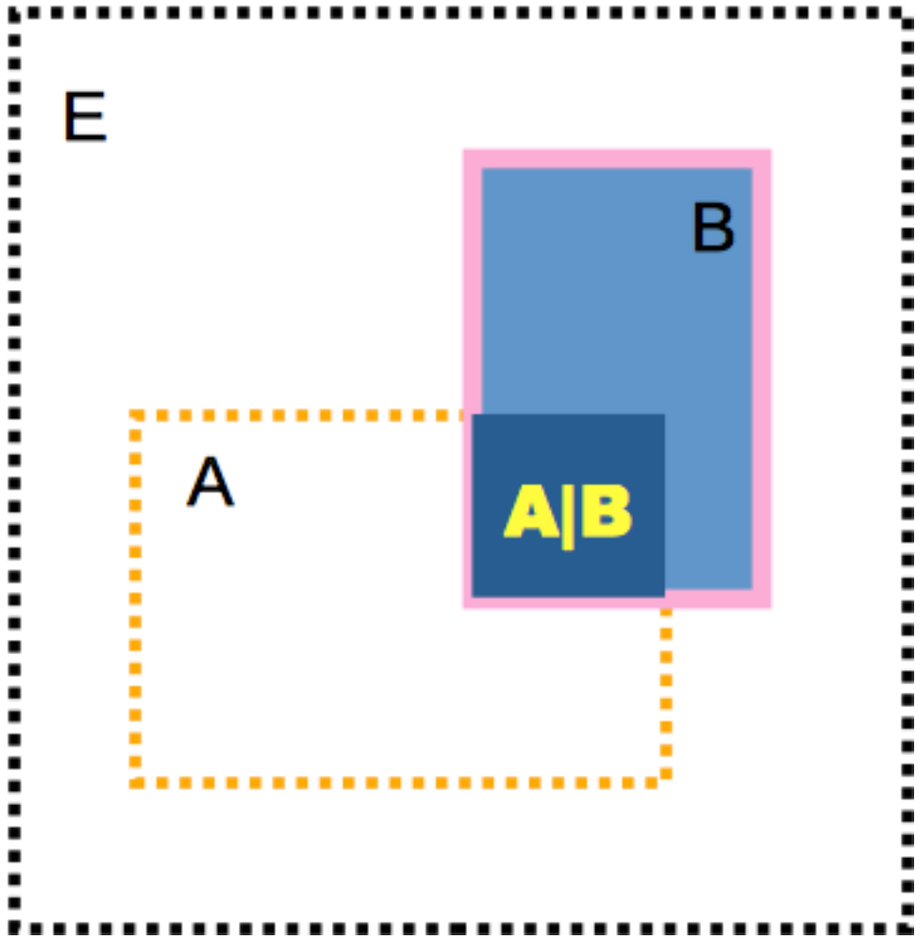
- La probabilité d'un événement A est notée P(A)
- Elle est comprise entre 0 et 1
- La probabilité d'un événement certain vaut 1
- La probabilité d'un événement impossible vaut 0
- Si A et B sont indépendants
  - $P(A \cup B) = P(A) + P(B)$
- P(non A) = 1 - P(A)

Probabilités conditionnelles

- P(A|B) = Probabilité que l'événement A survienne si l'événement B survient
- $P(A|B) = P(A \cap B) / P(B)$
- $P(A \cap B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$

Probabilités conditionnelles





Exemple

- 99% des sujets atteint d'une maladie M sont positifs à un test de dépistage
- La maladie M touche 10% de la population
- Quelle est la fraction de la population des sujets malades positifs au test de dépistage ?
- $P(M)=0.1$, $P(T|M)=0.99$
- $P(T \cap M) = P(T|M) \cdot P(M) = 0.99 \cdot 0.1 = 9.9\%$

Indépendance

- Deux événements sont indépendants si la connaissance de l'un ne modifie pas la probabilité de l'autre
- Si A et B sont indépendants, alors :
  - $P(A|B) = P(A)$
- Deux événements A et B sont indépendants si
- $P(A \wedge B) = P(A) \cdot P(B)$, $P(A), P(B) > 0$

Théorème de Bayes

- $P(A|B) = P(A \cap B) / P(B)$
- $P(A \cap B) = P(B|A) \cdot P(A)$
- Donc, **$P(A|B) = P(B|A) \cdot P(A) / P(B)$**

Problématique

- On veut calculer, pour chaque classe, la probabilité pour que ce soit la solution, sachant qu'on a observé $(a_1,...,a_n)$
- Garder la meilleure
- Rejeter les hypothèses de probabilité nulle
- Tout garder (traitements ultérieurs)

Problématique

- Quelle est l'hypothèse la plus probable, au vu de l'ensemble d'apprentissage ?
- Pour une instance donnée, au vu de l'ensemble d'apprentissage, quelle sera la classification la plus probable de cet exemple ?

Classifieur bayésien optimal

- Classification optimale si les probabilités de chaque hypothèse sont connues
- Pas souvent le cas :
  - Trop d'hypothèses, trop de calculs, trop d'estimations
- Simplification ?

Application à la classification

- $P(c_k|a_1,...,a_n) = P(a_1,...,a_n|c_k) \cdot P(c_k) / P(a_1,...,a_n)$

- $P(a_1,...,a_n|c_k)$, $P(a_1,...,a_n)$ et $P(c_k)$ peuvent être estimées sur les instances de l'ensemble d'apprentissage

Application à la classification

- $P(c_k|a_1,...,a_n) = P(a_1,...,a_n|c_k) \cdot P(c_k) / P(a_1,...,a_n)$
  - $P(c_k) \approx$ proportion d'instances de la classe c_k
  - $P(a_1,...,a_n) \approx$ proportion d'instances d'attributs $(a_1,...,a_n)$
  - $P(a_1,...,a_n|c_k) \approx$ nb fois où on rencontre $(a_1,...,a_n)$ dans les instances de la classe c_k (vraissemblance)

Quelques observations

- $P(c_k|a_1,...,a_n) = P(a_1,...,a_n|c_k) \cdot P(c_k) / P(a_1,...,a_n)$
  - $P(c_k|a_1,...,a_n)$ croît quand $P(c_k)$ croît : plus c_k est probable, plus il y a de chances qu'elle soit la classe
  - $P(c_k|a_1,...,a_n)$ croît quand $P(a_1,...,a_n|c_k)$ croît : si $(a_1,...,a_n)$ arrive souvent quand c_k est la classe, alors il y a des chances que c_k soit la classe
  - $P(c_k|a_1,...,a_n)$ décroît quand $P(a_1,...,a_n)$ croît : si $(a_1,...,a_n)$ est courant, il nous apprend peu sur c_k

Classification bayésienne

- C = (c_1,...,c_k) ensemble de classes (chacune munie d'une probabilité)
- (a_1,...,a_n) un ensemble d'attributs
- Retourner la classe ayant la probabilité la plus forte après l'observation de (a_1,...,a_n)
- Hypothèse Maximale A Posteriori : h_{MAP}
  - $h_{MAP} = \text{argmax}_{ck \in C} P(a_1,...,a_n|c_k) \cdot P(c_k)/P(a_1,...,a_n)$

Hypothèses MAP, ML

- $h_{MAP} = \text{argmax}_{c_k \in C}$

$$P(a_1,...,a_n|c_k) \cdot P(c_k) \,/\, P(a_1,...,a_n)$$

- $P(a_1,...,a_n)$ est constant
  - $h_{MAP} = \text{argmax}_{c_k \in C}\ \ P(a_1,...,a_n|c_k) \cdot P(c_k)$
- Maximum de vraisemblance
  - $h_{ML} = \text{argmax}_{c_k \in C}\ \ P(a_1,...,a_n|c_k)$

Classifieur bayésien naïf

- Appliquer le théorème de Bayes pour définir un algorithme de classification simple et efficace (en pratique)
- Caractéristiques :
  - Classification supervisée
  - Classes discrètes

Classifieur bayésien naïf

- On suppose que tous les attributs sont indépendants (eg. absence d'information)
- $P(a_1,...,a_n|c_k) = \prod P(a_i|c_k)$
  - $h_{MAP} = \text{argmax}_{ck \in C} \prod P(a_i|c_k) \cdot P(c_k)$
- $\prod P(a_i|c_k)$ estimé à partir de l'ensemble d'apprentissage

Classifieur bayésien naïf

- Attribut discret
  - $P(a_i|c) = n_{ic} / n_c$
  - n_{ic} = nombre d'instances de la classe c qui ont comme valeur a_i pour l'attribut considéré
  - n_c = nombre d'instances de la classe c

Classifieur bayésien naïf

- Attributs binaires : 2n valeurs à estimer
- Attributs indépendants
- Naïf : hypothèse d'indépendance (jamais vérifiée, simplification)
- Procédure sub-optimale, mais intérêt pratique
- Validité ? Robustesse ?

Classifieur bayésien naïf

- Soit X = (a_1,...,a_n) l'exemple à classer
- Estimer $P(X|c_k) \cdot P(c_k)$ pour chaque classe c_k
- Affecter à X la classe c_k telle que la probabilité $P(X|c_k) \cdot P(c_k)$ est la plus grande

Exemple: contrôle fiscal

- Faut-il effectuer un contrôle fiscal ?
- Echantillon

salaire	impôts	étudiant	contrôle
< 30	< 20 %	oui	négatif
30 - 50	< 20 %	non	positif
30 - 50	< 20 %	oui	positif
30 - 50	> 20 %	non	négatif
> 50	< 20 %	non	positif

- Contrôler ?

35	6 %	oui	?

Exemple: contrôle fiscal

- À classer : X = (sal=35, imp=6%,etu=oui)
  - P(pos|X) ? P(neg|X) ?
- Positif : P(sal=30-50|pos) · P(imp<20%|pos) · P(etu=oui|pos) · P(pos) = (2/3 · 1 · 1/3) · 3/5 = **0.13**
- Négatif : P(sal=30-50|neg) · P(imp<20%|neg) · P(etu=oui|neg) · P(neg) = (1/2 · 1/2 · 1/2) · 2/5 = **0.05**
- On effectuera donc un contrôle !

Exemple: tennis

- **Ciel, Température,Humidité,vent,Jouer?**
- Soleil,Chaud,Forte,faible,Non
- Soleil,Chaud,Forte,Fort,Non
- Couvert,Chaud,Forte,faible,Oui
- Pluie,Doux,Forte,faible,Oui
- Pluie,Frais,Normale,faible,Oui
- Pluie,Frais,Normale,Fort,Non
- Couvert,Frais,Normale,Fort,Oui
- Soleil,Doux,Forte,faible,Non
- Soleil,Frais,Normale,faible,Oui
- Pluie,Doux,Normale,faible,Oui
- Soleil,Doux,Normale,Fort,Oui
- Couvert,Doux,Forte,Fort,Oui
- Couvert,Chaud,Normale,faible,Oui
- Pluie,Doux,Forte,Fort,Non

Exemple: tennis

- Quelle classe attribuer à : (Soleil,Frais,Forte,Fort) ?

Exemple: tennis

- X = (Soleil,Frais,Forte,Fort)
- P(X|oui) · P(oui)

  = 2/9 · 3/9 · 3/9 · 3/9 · 9/14

  = 0.00529
- P(X|non) · P(non)

  = 3/5 · 1/5 · 4/5 · 3/5 · 5/14

  = **0.02057**

	oui	non
P(h)	9/14	5/14
Soleil	2/9	3/5
Frais	3/9	1/5
Forte	3/9	4/5
Fort	3/9	3/5

Exemple: tennis

- Confiance accordée à la classification
- P(non|X) = 0.02057 / (0.02057+0.00529)

  = **0.795**

Remarques

- Pas de construction de modèle
- Pas d'explication
- Efficacité comparable à celle d'autres algorithmes

Qualités du classifieur bayésien naïf

- **Prédiction** : comparable aux autres algorithmes
- **Vitesse** : linéaire en apprentissage, constant en classification
- **Robustesse** (bruit) : très bonne
- **Changement d'échelle** : les exemples ne sont pas en mémoire centrale (occupation mémoire = nb attributs x nb classes)
- **Lisibilité** : nulle

Aller plus loin... ?

- Deux cas extrêmes
  - Hypothèse d'indépendance des attributs : classifieur naïf, programmable mais approximatif
  - Dépendance des attributs deux-à-deux : trop complexe

Aller plus loin... ?

- Certains attributs sont liés par l'intermédiaire de la classe (ou d'un autre attribut)
- Fixer un petit nombre de dépendances entre les attributs

Classification de texte

Classification de textes

- Algo basé sur le classifieur bayésien naïf
- Simplifications supplémentaires
- Modèle de données modifié
- Classification supervisée
- Méthode efficace, mise en œuvre ‘facile’

Classification de textes

- Les exemples sont des textes, munis d'une classe :
  - Lettres : {V. Hugo, Napoléon} → quel auteur ?
  - AFP{sujets} → quel sujet ?
  - News {Newsgroups} → quel groupe ?
  - Mails : {Spam, Non Spam} → filtrer

    (certains filtres d'emails performants)

Classification de textes

- Comment transformer un texte ‘brut’ en schéma (attributs,valeurs) ?

- Quel sens donner à $P(c_k|A)$, A étant un texte ?

Représentation

- Texte : suite de caractères, lettres, chiffres, ponctuation, espaces, ...
- Représentation pour classification bayésienne ? Attributs ?

Première possibilité

- Schéma (attributs,valeurs)
- Un attribut est une position dans le texte
- Valeur de l'attribut : le mot présent en cette position
- Exemple : il était une fois ...

  a_1 a_2 a_3 a_4

  → $P(c_k) \cdot P(a_1=\text{‘il’}|c_k) \cdot P(a_2=\text{‘était’}|c_k) \cdot ...$

Première possibilité

- Remarques
  - $P(a_1 = \text{“il”} | c_k)$ grand quand c_k = ‘conte’
  - Hypothèse d’indépendance très simplificatrice (voir fausse !) :“il était une fois” caractéristique des contes
- Simplification nécessaire
  - Probas sur les mots, pas sur les textes

Première possibilité

- Quel sens donner à $P(c_k|A)$? Impossible !
- $P(a_i=w_j|c_k)$
  - a_i : position dans le texte → ≃100 positions
  - w_j : mot du texte → ≃20000 mots ≠
  - c_k : classe → 2 classes
  - 4 millions de probabilités à estimer !
  - Il faudrait plus de 4 millions de textes !

Deuxième possibilité

- La probabilité de rencontrer un mot dans un texte est indépendante de sa position
- Attribut = mot du vocabulaire
- Booléen : présence/absence du mot
- On ne se préoccupe ni de l'ordre de mots, ni de leur organisation dans le texte, ni de leur nombre d'occurrences

Deuxième possibilité

- $P(a_i=m_j|c_k) = P(m_j|c_k) = n_{jk} / n_k$
  - n_{jk} : nombre de fois où le mot m_j apparaît dans un texte de la classe c_k
  - n_k : nombre total de mots des textes de la classe c_k

Représentation

- Abstraction de l'**ordre** des mots, de la **syntaxe**, et de la **sémantique** des textes

Classification de textes

- Texte → suite de mots → sac de mots

  Algorithme (apprentissage) :

- Pour chaque classe c_k, estimer $P(c_k) = n_k / n$
  - n_k : nombre de textes de la classe c_k
  - n : nombre total de texte
- Pour tout mot m_j, estimer $P(m_j|c_k)$

Classification de textes

- Algorithme (classification) :
- D : texte à classifier
- Pour chaque mot m_j de D
  - $P(m_j|c_k)$ pour chaque classe c_k
- Pour chaque classe c_k
- $P(c_k|D) = \prod_{j \in D} P(m_j|c_k) \cdot P(c_k)$

  → Retourner la classe c_k maximisant $P(c_k|D)$

Classification de textes

- Complexité : O(n log(n)), avec n nombre de mots total (recherche d'un mot)
- Simplifications abusives ?
- Efficace ?

Améliorations

Classification de texte

Troisième possibilité

- Supprimer du ‘vocabulaire’ les articles, pronoms, ...
- Réduit la taille de l’ensemble des mots considérés
- Ces mots n’ajoutent rien au sens du texte

Classification de textes

- Regarder les n-grams au lieu des mots ? (avec jokers ?)
- Autres algorithmes (non bayésiens) : Boostexter