Apprentissage semi-supervisé asymétrique et estimation d'affinités locales dans les protéines

Christophe Nicolas Magnan

Laboratoire d'Informatique Fondamentale de Marseille (LIF), UMR CNRS 6166[†]
magnan@cmi.univ-mrs.fr

Résumé : Cet article présente une étude en apprentissage automatique semi-supervisé asymétrique, c'est-à-dire à partir de données positives et non-étiquetées, ainsi qu'une application à un problème bio-informatique. Nous montrons que sous des hypothèses très naturelles, le classifieur naïf de Bayes peut être identifié à partir de données positives et non étiquetées. Nous en déduisons des algorithmes que nous étudions sur des données artificielles. Enfin, nous présentons une application de ces travaux sur le problème de l'extraction d'affinités locales dans les protéines pour la prédiction des ponts disulfures. Les résultats permettent d'étayer une hypothèse sur la manière de formaliser les données biologiques pour des cas d'interactions physiques locales.

Mots-clés : Apprentissage semi-supervisé, Naïve Bayes, E.M., Ponts disulfures.

1 Introduction

De nombreux thèmes contemporains de recherche en biologie et en bio-informatique, liés à l'étude des protéines, sont étroitement en rapport avec des phénomènes d'interactions physiques locales. Les brins beta ou encore les ponts disulfures dans les protéines sont des exemples de phénomènes liés à des interactions physiques. Pouvoir prédire avec un maximum de précision et de pertinence ces interactions améliorerait de manière significative la prédiction de la structure tridimensionnelle des protéines, elle-même étant liée à sa fonction biologique. Déterminer cette structure expérimentalement, par résonance magnétique nucléaire, est une tâche longue, difficile, coûteuse et de plus, non applicable à certaines familles de protéines. Or on sait par ailleurs que la structure d'une protéine est déterminée par sa structure primaire, séquence d'acides aminés, et par son milieu d'occupation. Il est donc naturel de mettre au point des algorithmes de prédiction de la structure 3D à partir de la séquence primaire. Actuellement, près de 2 millions de séquences protéiques sont disponibles pour moins de 20000 structures 3D.

Nous nous sommes intéressés à un élément de la structure 3D : les ponts disulfures. Ces ponts sont des liaisons covalentes qui se forment entre deux cystéines suite à leur oxydation. La cystéine est un des 20 acides aminés qui constituent la séquence primaire

[†]Ces travaux sont en partie financés par l'A.C.I. Masses de Données GENOTO3D

des protéines. Une cystéine peut former un pont avec une autre cystéine proche ou distante sur la séquence, et est contrainte à une unicité de liaison.

La prédiction des ponts disulfures peut se décomposer en deux étapes : la prédiction des cystéines oxydées et la prédiction des ponts eux-mêmes. De nombreux travaux ont permis d'élaborer des méthodes adaptées à la première tâche, mais il y a peu de résultats pour la deuxième. C'est donc à la prédiction des ponts disulfures, sachant l'état d'oxydation des cystéines, que nous nous sommes intéressés.

Des acides aminés proches dans l'espace interagissent entre eux. On peut donc imaginer que les interactions entre les acides aminés situés autour de deux cystéines contribuent à ce que nous appellerons une *affinité* entre ces cystéines. Il est clair que cette information seule ne suffit pas à déterminer les ponts, mais nous cherchons à l'extraire au mieux dans le but de l'intégrer dans des processus de prédiction des ponts disulfures.

Déterminer si deux segments d'une protéine ont de l'affinité l'un pour l'autre peut être considéré comme un problème de classification supervisée, où les segments appariés sont vus comme ayant de l'affinité, et les segments non appariés comme n'en ayant pas. On peut également voir le problème de manière différente, en considérant comme précédemment que les couples de segments appariés ont de l'affinité l'un pour l'autre, mais que des couples de segments non appariés peuvent ou non en avoir. En effet, une cystéine ne pouvant appartenir qu'à un seul pont disulfure, le nombre de ponts dans une protéine est donc contraint. Nous modélisons cette situation en supposant que les paires de segments appariés appartiennent à la classe des segments qui ont de l'affinité l'un pour l'autre et que les paires de segments non appariés n'apportent pas d'informations sur la notion d'affinité.

C'est alors un cas d'apprentissage semi-supervisé, généralement rencontré sous le nom *d'apprentissage à partir de données positives et non étiquetées*, que nous appelons *asymétrique*. Ce contexte particulier d'apprentissage nécessite de montrer que le problème est bien posé (les données permettent-elles à la limite d'identifier la cible) et l'élaboration de nouveaux algorithmes.

Nous donnons dans la section 2 une brève description des méthodes utilisées lors de nos travaux ainsi que des résultats de la littérature sur ce thème de travail. En section 3, nous exposons notre étude théorique de l'apprentissage asymétrique et en section 4 et 5 nous présentons des résultats expérimentaux sur des données artificielles et réelles.

2 Préliminaires

Cette section présente les méthodes et algorithmes utilisés dans nos travaux, ainsi que divers résultats sur l'apprentissage à partir de données positives et non étiquetées.

2.1 La règle de Bayes et le classifieur naïf de Bayes

Soit $X = \prod_{i=1}^{m} X^i$ un domaine défini par m attributs symboliques. Pour tout $x \in X$, on notera x^i la projection de x sur X^i. Soit P une loi de distribution sur X et soit Y un ensemble de classes muni des lois de distributions conditionnelles $P(.|x)$ pour tout $x \in X$. La règle de décision optimale pour attribuer une classe à tout objet $x \in X$ est

la *règle de Bayes* qui sélectionne la classe $y \in Y$ possédant la plus grande probabilité sachant x. On peut formuler cette règle de la façon suivante :

$$\boxed{C_{Bayes}(x) \;=\; \underset{y}{argmax}\;\; P(y|x) \;=\; \underset{y}{argmax}\;\; P(x|y) \cdot P(y) \quad (x \in X, y \in Y)}$$

En règle générale, les quantités $P(x|y)$ ne peuvent pas être estimées à partir d'un échantillon d'apprentissage. En revanche, si les attributs sont indépendants deux à deux conditionnellement à chaque classe, on a alors $P(x|y) = \prod_{i=1}^{m} P(x^i|y)$ et dans ce cas, le nombre de paramètres à estimer devient raisonnable. Pour des classes et attributs binaires, le nombre de paramètres à estimer passe de $O(2^m)$ à $O(m)$. Que l'hypothèse d'indépendance soit vérifiée ou non, on appelle *classifieur naïf de Bayes* la règle définie par :

$$\boxed{C_{NB}(x) \;=\; \underset{y}{argmax}\;\; P(y) \prod_{i=1}^{m} P(x^i|y) \;\; (x \in X, y \in Y)}$$

L'hypothèse d'indépendance n'est pas vérifiée dans la plupart des problèmes réels. Néanmoins, le classifieur naïf de Bayes est connu pour donner de bons résultats pour des tâches de classification (Domingos & Pazzani, 1996).

Les paramètres nécessaires à l'évaluation de C_{NB} lorsque $Y = \{0, 1\}$ sont les probabilités $\alpha = P(y = 1)$, la probabilité d'observer un exemple de la classe notée 1 que nous appelons classe *positive* et l'ensemble des $\lambda_{ikj} = P(x^i = k|y = j)$, les probabilités d'observer l'attribut i de x égal à k sachant que x est de la classe j ($j \in \{0, 1\}$). Une instance de ces paramètres sera appelée *modèle* et sera notée θ.

2.2 Le principe du maximum de vraisemblance et application au classifieur naïf de Bayes

L'objectif de l'apprentissage automatique est de construire un modèle qui rend compte des données. Le principe du maximum de vraisemblance définit un critère permettant de choisir un tel modèle.

2.2.1 Principe du maximum de vraisemblance

Pour un échantillon $S = \{(x_s, y_s), s \in 1, ..., l\}$ de données indépendamment et identiquement distribuées selon la loi jointe $P(x, y) = P(x)P(y|x)$ et un modèle θ, on appelle *vraisemblance* (resp. *log-vraisemblance*) de S pour le modèle θ et on note $L(\theta, S)$ (resp. $l(\theta, S)$) les valeurs :

$$\boxed{L(\theta, S) = \prod_{s=1}^{l} P(x_s, y_s|\theta) \;\text{ et }\; l(\theta, S) = log\; L(\theta, S)}$$

Le principe du maximum de vraisemblance recommande de trouver un modèle θ tel que $L(\theta, S)$ - et donc aussi $l(\theta, S)$ - soit maximale.

2.2.2 Application au classifieur naïf de Bayes

Si on note n_0 le nombre de données classées '0' dans l'échantillon S d'apprentissage, n_1 le nombre de données classées '1' ($n_0 + n_1 = l$), n_{ij}^k le nombre d'exemples tels que $x^i = k$ et $y = j$ et $Dom(x^i)$ l'ensemble des valeurs que peut prendre l'attribut x^i, on peut écrire la vraisemblance et la log-vraisemblance de S dans le modèle θ en fonction de α et des λ_{ikj} :

$$L(\theta, S) = \prod_{s=1}^{l} P(y_s) \left[\prod_{i=1}^{m} P(x_s^i | y_s) \right] = \alpha^{n_1} \cdot (1-\alpha)^{n_0} \cdot \prod_{\substack{1 \le i \le m, 0 \le j \le 1 \\ k \in Dom(x^i)}} \lambda_{ikj}^{n_{ij}^k}$$

$$\begin{aligned} l(\theta, S) &= log\ L(\theta, S) \\ &= n_1 \cdot log\alpha + n_0 \cdot log(1-\alpha) + \sum_{\substack{1 \le i \le m, 0 \le j \le 1 \\ k \in Dom(x^i)}} n_{ij}^k \cdot log\ \lambda_{ikj} \end{aligned}$$

Cette fonction trouve son maximum pour le modèle θ_{mv}^S suivant :

- $\alpha = \frac{n_1}{n_0 + n_1}$, la proportion d'exemples positifs dans les données d'apprentissage,
- $\lambda_{ikj} = \frac{n_{ij}^k}{\sum_{r \in Dom(x^i)} n_{ij}^r}$, le rapport du nombre de données d'apprentissage de classe j tel que $x^i = k$ par le nombre de données étiquetées j.

2.2.3 Cas semi-supervisé

En contexte semi-supervisé, on dispose de deux échantillons de données : $S_{lab} = \{(x_1, y_1), ..., (x_l, y_l)\}$, un échantillon de données étiquetées, et $S_{unl} = \{x'_1, ..., x'_{l'}\}$, un échantillon de données non étiquetées. On modélise la présence de ces deux échantillons par l'existence d'un oracle qui, avec une certaine probabilité β fournit un exemple étiqueté et avec une probabilité $1 - \beta$ procure un exemple non étiqueté. Le paramètre β complète le modèle θ vu précédemment. Dans ce nouveau modèle, que nous notons θ', les probabilités d'avoir $z = (x, y) \in S_{lab}$ et d'avoir $z = x \in S_{unl}$ se calculent de la manière suivante :

$$P(z = (x, y) | \theta', z \in S_{lab}) = \beta . P(x, y | \theta)$$

$$P(z = x | \theta', x \in S_{unl}) = (1 - \beta) . P(x | \theta)$$

avec $P(x|\theta) = P(y = 1|\theta) . P(x, y|y = 1, \theta) + P(y = 0|\theta) . P(x, y|y = 0, \theta)$

La vraisemblance de S_{lab} *et* S_{unl} pour le modèle θ' s'écrit alors :

$$\begin{aligned} &L(\theta', S_{lab}, S_{unl}) = \prod_{s=1}^{l} \beta P(x_s, y_s | \theta) \prod_{r=1}^{l'} (1 - \beta) P(x'_r | \theta) \\ &= \beta^l L(\theta, S_{lab}) (1 - \beta)^{l'} \prod_{r=1}^{l'} \left(\alpha \prod_{\substack{1 \le i \le m \\ k / x_r^i = k}} \lambda_{ik1} + (1 - \alpha) \prod_{\substack{1 \le i \le m \\ k / x_r^i = k}} \lambda_{ik0} \right) \end{aligned}$$

Cette formule permet de trouver analytiquement la valeur optimale de β pour maximiser la vraisemblance : $\beta = \frac{l}{l+l'}$, soit la proportion d'exemples étiquetés dans l'ensemble d'apprentissage. En revanche, elle ne permet pas de trouver les paramètres α et λ_{ikj} du modèle θ' qui maximisent la vraisemblance, comme dans le cas supervisé du modèle naïf de Bayes. La méthode E.M. permet de pallier ce problème (*cf.* section 2.3).

2.2.4 Cas semi-supervisé asymétrique

Le cas semi-supervisé asymétrique est proche du cas semi-supervisé classique. L'ensemble S_{lab} est réduit à $S_{pos} = \{(x_1, 1), ..., (x_l, 1)\}$, cela entraîne que le paramètre β représente maintenant la probabilité d'observer un exemple positif. Avec les mêmes notations que précédemment, la vraisemblance de S_{pos} et S_{unl} pour le modèle θ' s'écrit :

$$\boxed{\begin{array}{l} L(\theta', S_{pos}, S_{unl}) \\ = \beta^l L(\theta, S_{pos})(1-\beta)^{l'} L(\theta, S_{unl}) \\ = \beta^l \alpha^l \prod_{r=1}^{l} \left(\prod_{\substack{1 \leq i \leq m \\ k/x_r^i = k}} \lambda_{ik1} \right) (1-\beta)^{l'} \prod_{r=1}^{l'} \left(\alpha \prod_{\substack{1 \leq i \leq m \\ k/x_r^i = k}} \lambda_{ik1} + (1-\alpha) \prod_{\substack{1 \leq i \leq m \\ k/x_r^i = k}} \lambda_{ik0} \right) \end{array}}$$

2.3 La méthode E.M. (Expectation, Maximisation)

La méthode E.M. a été élaborée par (Dempster *et al.*, 1977) pour l'inférence de modèles de mélange de densités.

2.3.1 Méthode

Cette section décrit la méthode E.M. en suivant (Hastie *et al.*, 2001). Soient θ' un modèle, Z l'ensemble des données observées, Z_m les données manquantes, et T l'ensemble de données complètes d'un problème, $T = (Z, Z_m)$. Si on note :

- $l_0(\theta', T)$ la log-vraisemblance de T dans le modèle θ',
- $l_1(\theta', Z_m|Z)$ la log-vraisemblance de Z_m dans le modèle θ' sachant Z,
- $l(\theta', Z)$ la log-vraisemblance de Z dans le modèle θ',

alors $l(\theta', Z) + l_1(\theta', Z_m|Z) = l_0(\theta', T)$, soit :

$$\boxed{l(\theta', Z) = l_0(\theta', T) - l_1(\theta', Z_m|Z)}$$

En supposant que les données sont générées selon θ et que Z a été observé, les termes de l'égalité précédente sont des variables aléatoires dépendantes de Z_m, on peut donc calculer leur espérance :

$$\boxed{E(\ l(\theta', Z)|Z, \theta\ ) = E(\ l_0(\theta', T)|Z, \theta\ ) - E(\ l_1(\theta', Z_m)|Z, \theta\ )}$$

Soit, en posant $Q(\theta', \theta) = E(\ l_0(\theta', T)|Z, \theta\ )$ et $R(\theta', \theta) = E(\ l_1(\theta', Z_m)|Z, \theta\ )$ et en remarquant que $E(\ l(\theta', Z)|Z, \theta\ ) = l(\theta', Z)$:

$$\boxed{l(\theta', Z) = Q(\theta', \theta) - R(\theta', \theta)}$$

La méthode du maximum de vraisemblance demande de chercher un modèle θ' qui maximise $l(\theta', Z)$. La méthode E.M. est une heuristique, basée sur le résultat suivant qui énonce que maximiser Q ne peut pas faire décroître la vraisemblance.

Théorème 1
Si $Q(\theta', \theta) > Q(\theta, \theta)$ *alors* $l(\theta', Z) > l(\theta, Z)$ *(Dempster* et al.*, 1977)*

La méthode E.M. peut être décrite par l'algorithme suivant :

Algorithme 1
Entrée : Z
1) Choisir un modèle $\hat{\theta}^0$
2) Calculer $Q(\hat{\theta}^i, \hat{\theta}^i)$ *pour le* i *courant (phase d'estimation)*
3) Trouver $\hat{\theta}^{i+1}$ *tel que* $Q(\hat{\theta}^{i+1}, \hat{\theta}^i) > Q(\hat{\theta}^i, \hat{\theta}^i)$ *(phase de maximisation)*
4) Itérer à l'étape deux jusqu'à convergence
Sortie : *un modèle* θ^c

L'algorithme converge vers un maximum local de la vraisemblance. (Dempster *et al.*, 1977) proposent de répéter l'expérience et de choisir le modèle θ^c de plus grande vraisemblance.

2.3.2 Application au classifieur naïf de Bayes

Pour le classifieur naïf de Bayes en contexte semi-supervisé, $Z = \{S_{lab}, S_{unl}\}$ avec $S_{lab} = \{(x_1, y_1), ..., (x_l, y_l)\}$ et $S_{unl} = \{x'_1, .., x'_{l'}\}$, les données manquantes Z_m sont les étiquettes des données de S_{unl} et $T = (Z, Z_m)$. Avec les mêmes notations que précédemment, les paramètres $\alpha = P(y = 1)$ et $\lambda_{ikj} = P(x^j = k|y = i)$ se calculent de la manière suivante à chaque itération de l'algorithme (McCallum *et al.*, 1999) :

$$\alpha = \frac{n_1 + \sum_{s=1}^{l'} \hat{P}(y'_s = 1|x'_s, \hat{\theta})}{l + l'}, \quad \lambda_{ikj} = \frac{n_{ij}^k + \sum_{s=1}^{l'} \hat{P}(y'_s = j|x_s'^i = k, \hat{\theta})}{\sum_{r \in Dom(x^i)} [n_{ij}^r + \sum_{s=1}^{l'} \hat{P}(y'_s = j|x_s'^i = r, \hat{\theta})]}$$

où les $\hat{P}$ sont estimées en fonction du modèle courant $\hat{\theta}$. (McCallum *et al.*, 1999) proposent l'algorithme suivant :

Algorithme 2 (EM+NB semi-supervisé)
Entrée : S_{lab}, S_{unl}
1) $\hat{\theta}^0 =$ *N.B.*(S_{lab})*, le modèle appris sur les données étiquetées*
2) $\forall x' \in S_{unl}$ *calculer* $P(y' = j|x', \hat{\theta}^k)$*, avec le modèle* $\hat{\theta}^k$ *courant,* $j \in \{0, 1\}$
3) Maximiser $Q(\hat{\theta}^{k+1}, \hat{\theta}^k)$
4) Itérer à l'étape 2 jusqu'à convergence
Sortie : *un modèle* θ^c

Les résultats sur un problème de classification de textes montrent une amélioration sensible des résultats lors de l'ajout de données non étiquetées aux données étiquetées.

2.4 Apprentissage à partir de données positives et non étiquetées

Le thème de l'apprentissage à partir de données positives et non étiquetées a déjà été abordé par divers chercheurs (Denis, 1998; Denis *et al.*, 1999; Liu & Li, 2003). Dans ce contexte, on trouve également l'utilisation du classifieur naïf de Bayes pour une application à la classification de textes (Denis *et al.*, 2003). Dans ces travaux, les auteurs partent de l'hypothèse que le paramètre $\alpha = P(y = 1)$ est connu, ce paramètre étant indispensable pour calculer le classifieur (*cf.* section 3.1). Or, ce paramètre étant généralement inconnu, l'estimation de celui-ci est donc un problème latent.

Il a été montré dans (Whiley & Titterington, 2002; Geiger *et al.*, 2001) que, sous l'hypothèse d'indépendance des attributs conditionnellement à chaque classe, les paramètres du modèle sont identifiables à partir de la distribution $P(.)$ sur X lorsque le nombre d'attributs est supérieur à deux et à la détermination des classes près, c'est-à-dire $P(.)$ détermine l'ensemble $\{\alpha, 1 - \alpha\}$. Des formules analytiques permettant d'estimer les paramètres du modèle à partir d'échantillons d'exemples non étiquetés sont également fournies. Mais dans le cas qui nous intéresse, nous disposons aussi d'exemples positifs qui doivent permettre d'identifier les classes et d'obtenir de meilleures estimations.

3 Estimation des paramètres d'un classifieur naïf de Bayes

Après avoir exposé quelques propriétés sur le cas général de l'apprentissage asymétrique (section 3.1), nous établissons une formule qui montre que les paramètres du modèle sont identifiables lorsque le nombre d'attributs est supérieur ou égal à deux (section 3.2). Cette formule permet de définir un estimateur consistant du paramètre $P(y = 1)$. En section 3.3, nous proposons une adaptation de l'algorithme 2 au cas semi-supervisé asymétrique en vue de comparer les deux algorithmes sur des données artificielles.

3.1 Cas général

Le cadre général de l'apprentissage statistique suppose l'existence de distributions $P(.)$ sur X et $P(.|x)$ sur Y pour tout $x \in X$. Dans le cas $Y = \{0, 1\}$, ces distributions sont déterminées par la donnée de $P(.)$ et $P(.|y = 1)$ sur X, et $P(y = 1)$. En effet :

$$\boxed{P(y = 1|x) = \frac{P(x|y = 1) \cdot P(y = 1)}{P(x)} \text{ et } P(y = 0|x) = 1 - P(y = 1|x)}$$

Des échantillons de données positives et non étiquetées, S_{pos} et S_{unl}, permettent de déterminer des estimations des distributions $P(.)$ sur X et $P(.|y = 1)$ sur X. Dans le cas général le paramètre $P(y = 1)$ doit être connu.

Propriété 1
En règle générale, P(y=1) n'est pas déterminé par la donnée de $P(x)$ et $P(x|y = 1)$.

Soit $r = Inf\{\frac{P(x)}{P(x|y=1)} \mid x \in X \text{ et } P(x|y=1) \neq 0\}$. Alors pour tout $\lambda \in]0, r]$, il existe P' tel que pour tout $x \in X$, $P'(x) = P(x)$, $P'(x|y=1) = P(x|y=1)$ et $P'(y=1) = \lambda$. En effet, soit λ tel que $0 < \lambda \leq r$. Alors, en posant :

$$\boxed{P'(x|y=0) = \frac{P'(x) - P'(x|y=1).\lambda}{1-\lambda} \ \forall x \in X}$$

on obtient $P'(y=1) = \lambda$. L'ensemble des λ acceptables étant l'intervalle $]0, r]$, le paramètre $P(y=1)$ n'est donc pas déterminé.

Remarque
Dans certains cas, $P(y=1)$ est déterminé par les paramètres $P(x)$ et $P(x|y=1)$. Un cas trivial est celui des modèles déterministes : pour tout x, $P(y=1|x) = 1$ ou $P(y=1|x) = 0$. Dans ce cas $P(y=1) = \sum_{x \in X} P(x)P(y=1|x) = \sum_{P(x|y=1) \neq 0} P(x)$.

3.2 Déterminisme et identification du paramètre P(y=1)

Nous montrons dans cette section que les distributions $P(.)$ sur X et $P(.|y=1)$ sur X déterminent le paramètre $P(y=1)$ pour les distributions suivant l'hypothèse naïve de Bayes et nous donnons un estimateur consistant pour ce paramètre. L'apprentissage à partir de données positives et non étiquetées est donc envisageable sans hypothèse supplémentaire.

Théorème 2
Pour les distributions de probabilité satisfaisant l'hypothèse naïve de Bayes, la donnée de $P(.)$ sur X et de $P(.|y=1)$ sur X détermine le paramètre $P(y=1)$ sous réserve qu'il existe au moins deux attributs distincts x^i et x^j tels que $P(x^i = .|y=1) \neq P(x^i = .|y=0)$ et $P(x^j = .|y=1) \neq P(x^j = .|y=0)$.

Démonstration
Avant le cas général, nous traitons les deux cas limites du paramètre $P(y=1)$:

- Remarquons tout d'abord que le cas $P(y=1) = 0$ est impossible puisque l'on suppose l'existence de l'ensemble S_{pos}.
- De plus, sous les conditions du théorème, nous montrons que $P(y=1) = 1 <=> P(.) = P(.|y=1)$ pour tout $x \in X$, en effet,
  - l'implication $=>$ est triviale,
  - supposons que $P(.) = P(.|y=1)$ et $P(y=1) < 1$, alors :
    $P(.|y=1) \cdot (1 - P(y=1)) = P(.|y=0) \cdot (1 - P(y=1))$ et donc :
    $P(.|y=1) = P(.|y=0)$, ce qui contredit l'hypothèse.

Nous considérons maintenant le cas général $0 < P(y=1) < 1$.
Soient $p_{ik} = P(x^i = k|y=1)$ et $q_{ik} = P(x^i = k|y=0)$ $\forall i \in \{1, ..., m\}$. Pour tout couple (i, j) d'attributs distincts, et pour tout couple k, l de valeurs respectives des attributs x^i et x^j, on peut déduire le système d'équations suivant :

$$\begin{cases} \alpha_{ik} = P(x^i = k) = p_{ik}.P(y=1) \ + \ q_{ik}.(1 - P(y=1)) \\ \alpha_{jl} = P(x^j = l) = p_{jl}.P(y=1) \ + \ q_{jl}.(1 - P(y=1)) \\ \alpha_{ik,jl} = P(x^i = k \cap x^j = l) = p_{ik}.p_{jl}.P(y=1) \ + \ q_{ik}.q_{jl}.(1 - P(y=1)) \end{cases}$$

Considérons un couple (i, j) d'attributs distincts et un couple (k, l) de valeurs d'attributs respectives pour x^i et x^j tels que $p_{ik} \neq q_{ik}$ et $p_{jl} \neq q_{jl}$, avec les égalités provenant des deux premières équations du système, on peut écrire :

$$\boxed{q_{ik} = \frac{\alpha_{ik} - p_{ik}.P(y=1)}{1 - P(y=1)} \quad q_{jl} = \frac{\alpha_{jl} - p_{jl}.P(y=1)}{1 - P(y=1)}}$$

En remplaçant q_{ik} et q_{jl} dans la troisième équation du système, on obtient, après simplification, l'équation du premier degré en $P(y = 1)$ suivante :

$$\boxed{P(y=1)(p_{ik}p_{jl} - \alpha_{ik}p_{jl} - \alpha_{jl}p_{ik} + \alpha_{ik,jl}) = \alpha_{ik,jl} - \alpha_{ik}\alpha_{jl}}$$

Pour obtenir une expression analytique de $P(y = 1)$ à partir de cette équation, il est nécessaire de montrer que $p_{ik}p_{jl} - \alpha_{ik}p_{jl} - \alpha_{jl}p_{ik} + \alpha_{ik,jl} \neq 0$ et que $\alpha_{ik,jl} \neq \alpha_{ik} \cdot \alpha_{jl}$ (sans quoi $P(y = 1) = 0$).

- On peut écrire ($p_{ik}p_{jl} - \alpha_{ik}p_{jl} - \alpha_{jl}p_{ik} + \alpha_{ik,jl}$) en fonction de p_{ik}, p_{jl}, q_{ik}, q_{jl}, et $P(y = 1)$ en remplaçant α_{ik}, α_{jl}, $\alpha_{ik,jl}$ par leur définition. On obtient :

$$\boxed{p_{ik}p_{jl} - \alpha_{ik}p_{jl} - \alpha_{jl}p_{ik} + \alpha_{ik,jl} = (1 - P(y=1)).(p_{ik} - q_{ik}).(p_{jl} - q_{jl})}$$

Donc sous les conditions du théorème : $p_{ik}p_{jl} - \alpha_{ik}p_{jl} - \alpha_{jl}p_{ik} + \alpha_{ik,jl} \ \neq 0$.

- $\alpha_{ik,jl} = \alpha_{ik} \cdot \alpha_{jl}$ signifie que les attributs descriptifs sont indépendants. Or, nous travaillons sous l'hypothèse d'indépendance des attributs conditionnellement à chaque classe. Les deux conditions réunies ont pour conséquence directe (développement non précisé faute de place) : $P(y = 1) \cdot (1 - P(y = 1)) \cdot (p_{ik} - q_{ik}) \cdot (p_{jl} - q_{jl}) = 0$. Sous les conditions du théorème, cette égalité n'est jamais vérifiée.
On peut donc écrire :

$$\boxed{(1) \quad P(y=1) = \frac{\alpha_{ik,jl} - \alpha_{ik}\alpha_{jl}}{p_{ik}p_{jl} - \alpha_{ik}p_{jl} - \alpha_{jl}p_{ik} + \alpha_{ik,jl}}}$$

Les paramètres $\alpha_{ik,jl}, \alpha_{ik}, \alpha_{jl}, p_{ik}, p_{jl}$ pouvant être calculés à partir des distributions $P(.)$ et $P(.|y = 1)$, $P(y = 1)$ est bien déterminé par $P(.)$ et $P(.|y = 1)$.■

Cette formule conduit à une estimation naturelle du paramètre $P(y = 1)$. Soient $\hat{\alpha}_{ik,jl}, \hat{\alpha}_{ik}, \hat{\alpha}_{jl}, \hat{p}_{ik}, \hat{p}_{jl}$ des estimateurs des paramètres $\alpha_{ik,jl}, \alpha_{ik}, \alpha_{jl}, p_{ik}, p_{jl}$ respectivement, on considère :

$$(2) \quad \hat{P}(y=1) = \frac{\sum_{i,j,k,l} |\hat{\alpha}_{ik,jl} - \hat{\alpha}_{ik}\hat{\alpha}_{jl}|}{\sum_{i,j,k,l} |\hat{p}_{ik}\hat{p}_{jl} - \hat{\alpha}_{ik}\hat{p}_{jl} - \hat{\alpha}_{jl}\hat{p}_{ik} + \hat{\alpha}_{ik,jl}|}$$

avec $i \neq j$ et k et l un couple de valeurs respectives des attributs x^i et x^j. On en déduit l'algorithme d'apprentissage suivant :

Algorithme 3 (NB semi-sup. asymétrique)
Entrée : S_{pos}, S_{unl}
1) Calculer les estimateurs $\hat{\alpha}_{ik,jl}, \hat{\alpha}_{ik}, \hat{\alpha}_{jl}, \hat{p}_{ik}, \hat{p}_{jl}$ *des paramètres* $\alpha_{ik,jl}, \alpha_{ik}, \alpha_{jl}, p_{ik}, p_{jl}$ *sur* S_{pos} *et* S_{unl}
2) Calculer $\hat{P}(y=1)$ *par la formule (2)*
3) Calculer les estimateurs des paramètres manquants du modèle : $\hat{q}_{ik}$, $\hat{q}_{jl}$
Sortie : *un modèle* $\hat{\theta}$

En l'absence de résultats théoriques sur la vitesse de convergence de l'estimateur (2), nous avons décidé de comparer les résultats obtenus par l'algorithme 3 à ceux que l'on obtient en maximisant la vraisemblance au moyen de la méthode E.M.. Pour cela, nous proposons un algorithme adapté de l'algorithme 2 à la section suivante.

3.3 Algorithme naïf de Bayes en contexte semi-supervisé asymétrique

Nous présentons ici un algorithme itératif, adapté de l'algorithme 2 (McCallum *et al.*, 1999) qui permet d'estimer le paramètre $P(y=1)$ sur le critère du maximum de vraisemblance. Nous proposons la solution suivante :

Algorithme 4 (EM+NB asym. + aléa)
Entrée : S_{pos}, S_{unl}
M=$\emptyset$
Estimer les quantités $P(x^i = k|y=1)$ *et* $P(x^i=k)$ *avec* S_{pos} *et* S_{unl}.
Répéter
1) Tirer aléatoirement un paramètre $P(y=1)$
2) Calculer un modèle θ^0 *à partir des estimations des* $P(x^i=k|y=1)$ *et* $P(x^i=k)$ *et du paramètre* $P(y=1)$ *tiré alátoirement.*
3) $\forall x' \in S_{unl}$ *calculer* $P(y=j|x', \theta^k)$, *avec le* k *courant et où* $j \in \{0,1\}$
4) Maximiser $Q(\theta^{k+1}, \theta^k)$ *(cf section 2.3.2 pour le détail des calculs)*
5) Itérer à l'étape 3 jusqu'à convergence
6) Insérer le modèle θ *final dans M*
Choisir un modèle θ *dans M de vraisemblance maximale.*
Sortie : *un modèle* θ

On peut également dériver de cet algorithme un cinquième algorithme en remplaçant le tirage aléatoire de $P(y=1)$ à la phase 1 de l'algorithme par l'estimation de $P(y=1)$ donnée par la formule (2). Dans ce cas, la boucle répéter est inutile. Nous noterons **Algorithme 5 (EM+NB asym. + (2))** cette variante.

4 Résultats expérimentaux sur des données artificielles

4.1 Exemple

Cette section présente un exemple de déroulement de l'algorithme 4 et expose le protocole expérimental utilisé dans le cadre des expériences sur des données artificielles.

Le modèle cible θ_c est tiré aléatoirement, il satisfait l'hypothèse naïve de Bayes. Les données possèdent 50 attributs binaires, l'ensemble S_{pos} contient 20 données $(x_i, 1),\ i \in \{1, ..., 20\}$, tirées aléatoirement selon les distributions $P(x|y = 1)$ du modèle cible, l'ensemble S_{unl} contient 1000 données $x'_i,\ i \in \{1, ..., 1000\}$, tirées aléatoirement selon les distributions $P(x)$ et enfin, pour tester la performance des modèles inférés, un ensemble S_{test} contenant 1000 données $(x''_i, y''_i), i \in \{1, ..., 1000\}$, $y''_i \in \{0, 1\}$, est généré selon les distributions $P(y = 1)$, $P(x|y = 1)$ et $P(x|y = 0)$.

Les trois critères observés lors du déroulement de l'algorithme sont : le paramètre $P(y = 1)$, le taux d'erreur sur les données test des différents modèles et la log-vraisemblance des modèles inférés sur l'ensemble S_{test}. Le modèle θ_c généré aléatoirement prend les valeurs suivantes pour ces paramètres : $P(y = 1) = 0,5798$, $Erreur(\theta_c, S_{test}) = 0,045$ et $l(\theta_c, S_{test}) = -5651,84$.

La phase d'initialisation est numérotée 0. Le paramètre $P(y = 1)$ tiré aléatoirement à la phase 1 de l'algorithme est indiqué en gras. Chaque étape correspond à une phase complète d'itération (indices 3, 4 et 5 de l'algorithme 4). La dernière ligne, marquée en gras, indique la dernière étape avant convergence.

étape k	P(y=1)	log-vraisemblance(θ^k,S_{test})	Erreur(θ^k,S_{test})
0	**0,7868**	-6813,95	0,248
1	0,8286	-5909,26	0,188
2	0,7762	-5804,40	0,124
3	0,7221	-5735,62	0,111
4	0,6744	-5694,87	0,046
5	0,6368	-5673,13	0,046
6	0,6104	-5663,93	0,045
7	0,5934	-5661,03	0,045
8	0,5830	-5660,45	0,044
9	**0,5767**	**-5660,38**	**0,045**

Tableau 1

Une seule itération de la boucle "répéter" est indiquée, nous avons constaté que différents $P(y = 1)$ tirés aléatoirement menaient la plupart du temps au même modèle.

4.2 Comparaison de l'algorithme naïf de Bayes dans les cas semi-supervisés classique et asymétrique

Cette section présente les résultats expérimentaux obtenus sur des données générées artificiellement selon le protocole exposé section 4.1. Différentes tailles des ensembles S_{lab} et S_{unl} et différents nombres d'attributs binaires par donnée sont testés. Ces résultats permettent de comparer les performances des algorithmes 2, 3, 4 et 5.

L'algorithme 2 utilise un ensemble S_{lab} de données étiquetées et un ensemble S_{unl} de données dont la classe n'est pas connue, les tailles de ces ensembles sont signalées dans le tableau 2. Les algorithmes 3, 4 et 5 utilisent les données positives de l'ensemble S_{lab} et le même ensemble S_{unl}. La taille de S_{pos} varie en fonction du paramètre $P(y = 1)$ du modèle cible et de la taille de S_{lab} et vaut approximativement $P(y = 1) * |S_{lab}|$.

Enfin, chaque résultat présenté dans le tableau 2 est une moyenne calculée sur 200 expériences. La lecture en ligne du tableau permet d'observer l'évolution des performances des algorithmes en fonction du nombre d'attributs. En colonne, elle permet d'observer les changements induits par des modifications de la taille des ensembles S_{lab} et S_{unl}, et permet également de comparer les algorithmes entre eux. Entre parenthèses sont indiqués les écarts-types des taux d'erreurs et en gras, les moyennes des vitesses de convergence apparentes pour les algorithmes utilisant la méthode E.M. (2,4 et 5), soit la moyenne du nombre d'itérations avant stabilisation.

	Nb attributs x^i	20	50
Performance Modèle cible		0.0410 (0.0265)	0.0035 (0.0038)
$\|S_{lab}\| = 50$ $\|S_{unl}\| = 100$ $\|S_{pos}\| =$ $\alpha \cdot \|S_{lab}\|$	Algo 2 EM+NB semi-supervisé	0.0787 (0.1481) **24.46**	0.0621 (0.2166) **8.22**
	Algo 3 NB semi-sup. asym.	0.1236 (0.0753)	0.1037 (0.0662)
	Algo 4 EM+NB asym.+aléa	0.0592 (0.0601) **30.70**	0.0346 (0.1040) **9.99**
	Algo 5 EM+NB asym.+(2)	0.0653 (0.0531) **22.64**	0.0465 (0.1364) **7.63**
$\|S_{lab}\| = 100$ $\|S_{unl}\| = 1000$ $\|S_{pos}\| =$ $\alpha \cdot \|S_{lab}\|$	Algo 2 EM+NB semi-supervisé	0.0536 (0.0978) **23.19**	0.0140 (0.0936) **6.53**
	Algo 3 NB semi-sup. asym.	0.0911 (0.0527)	0.0514 (0.0478)
	Algo 4 EM+NB asym.+aléa	0.0460 (0.0425) **30.83**	0.0221 (0.0962) **9.03**
	Algo 5 EM+NB asym.+(2)	0.0460 (0.0288) **24.09**	0.0170 (0.0656) **6.15**
$\|S_{lab}\| = 1000$ $\|S_{unl}\| = 5000$ $\|S_{pos}\| =$ $\alpha \cdot \|S_{lab}\|$	Algo 2 EM+NB semi-supervisé	0.0434 (0.0294) **21.61**	0.0037 (0.0040) **4.11**
	Algo 3 NB semi-sup. asym.	0.0490 (0.0285)	0.0140 (0.0182)
	Algo 4 EM+NB asym.+aléa	0.0434 (0.0294) **28.56**	0.0103 (0.0573) **7.74**
	Algo 5 EM+NB asym.+(2)	0.0440 (0.0289) **21.13**	0.0088 (0.0690) **5.14**

Tableau 2

L'algorithme 2 est celui qui utilise le plus de données, c'est l'algorithme le plus efficace lorsque le nombre de données est grand. L'estimateur (2), utilisé pour l'étape initiale de l'algorithme 5, permet d'augmenter la rapidité de convergence par rapport à un choix aléatoire de $P(y = 1)$ (algorithme 4). On observe également une amélioration rapide des performances de l'algorithme 3 lorsque le nombre de données augmente.

5 Prédiction de ponts disulfures dans les protéines

Les ponts disulfures sont des liaisons covalentes, entre deux acides aminés (cystéines) de la chaîne protéique, qui contraignent la structure 3D de cette protéine. Des travaux préliminaires ont montré que l'information recherchée, tel couple de cystéines forme-t-il un pont, est vraisemblablement dispersée et portée par des éléments très divers : environnement, propriétés chimiques, etc... Mais il est vraisemblable que cette information soit aussi en partie portée par le voisinage des cystéines. C'est cette part de l'information que nous cherchons à extraire, sans prétendre pouvoir résoudre le problème de la prédiction des ponts disulfures avec cette seule donnée.

Nous pensons qu'il est préférable de considérer un couple de cystéines non observées appariées comme un exemple de classe indéterminée plutôt que de le considérer comme un exemple *négatif*, c'est-à-dire n'étant pas compatible. En effet, chaque cystéine est contrainte à une unicité de liaison et il est pourtant probable qu'elle soit compatible avec plusieurs autres cystéines de la même protéine. C'est pourquoi nous cherchons à montrer qu'il ne faut pas considérer les couples non observés appariés comme des représentants de couples qui ne peuvent pas s'apparier.

5.1 Les données

Les données sont extraites de la *Protein Data Bank* (PDB) pour les besoins du groupe de travail de l'ACI GENOTO3D (*http ://www.loria.fr/˜guermeur/GdT/*). Le fichier de données dont nous disposons contient 227 séquences protéiques (mots sur un alphabet à 20 lettres) qui ont toutes leurs cystéines oxydées et appariées par un pont disulfure. La répartition des protéines en fonction de leur taille (nombre d'acides aminés de la séquence) est indiquée figure 1 et en fonction du nombre de ponts figure 2.

figure 1

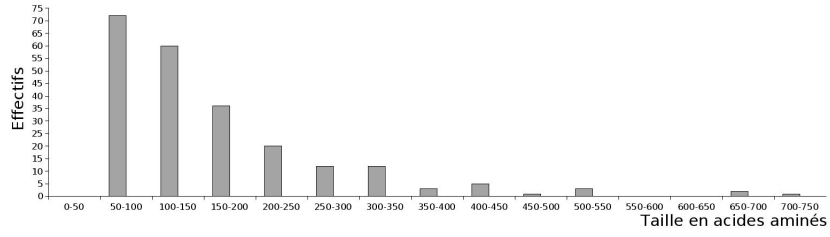
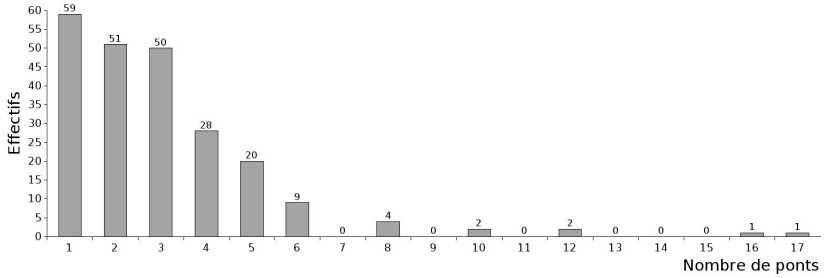


figure 2



5.2 Protocole expérimental

5.2.1 Modélisation des données

Nous cherchons à estimer des affinités locales dans les protéines. Dans le cas des ponts disulfures, ces attractions se font entre les acides aminés proches des deux cystéines appariées. C'est pourquoi nous avons extrait de chaque séquence protéique des fragments de taille fixe centrés sur les cystéines. Nous appelons ces fragments des *fenêtres* et notons $x_{-n}, .., x_{-1}, x_0, x_1, ..., x_n$ une fenêtre de rayon n (x_0 est donc une cystéine). Nous travaillons sur un alphabet de 231 lettres (nombre de couples ordonnés sur un alphabet à 21 lettres : les 20 acides aminés et un caractère pour les fins de chaîne). Pour représenter l'affinité entre deux segments f et f', trois codages sont testés :

- Codage simple : $\{(x_i, x'_i)\},\ i \in \{-n, ..., n\}, i \neq 0, x_i \in f, x'_i \in f'$
- Codage double : $\{(x_i, x'_i)\} \cup \{(x_i, x'_{-i})\},\ i \in \{-n, ..., n\}, i \neq 0, x_i \in f, x'_i \in f'$
- Codage croisé : $\{(x_i, x'_j)\},\ i, j \in \{-n, ..., n\}, x_i \in f, x'_i \in f'$

Une donnée est donc un couple de fenêtres representé par 231 attributs n-aires. Chacun de ces attributs représentant le nombre d'occurrences du couple dans la donnée.

5.2.2 Protocole d'apprentissage

Pour une protéine contenant n ponts, on compte $n(2n-1)$ couples de fenêtres potentiellement en interaction. Si un couple est observé comme formant un pont, on le considère comme un exemple positif. Quand aux autres couples, nous les considérons dans un premier temps comme des exemples ne pouvant pas former un pont, et dans un deuxième temps comme des exemples non étiquetés. Pour le premier cas, nous avons utilisé l'algorithme naïf de Bayes (section 2.1), et pour le deuxième l'algorithme 4.

L'apprentissage se fait sur des protéines ayant le même nombre n de ponts. Nous avons étudié $n = 2, 3, 4$ et 5. Le cas $n = 1$ étant trivial, il est pas étudié. Pour $n > 5$, nous ne disposons pas d'assez de données pour que les résultats soient significatifs.

5.2.3 Protocoles de test

Pour tester la pertinence des estimations d'affinité issues des deux algorithmes, nous proposons un protocole de test ne prennant en compte que cette information et permettant de comparer la qualité des estimations des deux algorithmes :

- calculer pour chaque couple de fenêtres d'une protéine test l'affinité entre ces deux fenêtres dans le modèle généré par l'algorithme d'apprentissage ;
- trouver la configuration la plus vraisemblable. Cela revient à trouver dans un graphe complet le couplage parfait de poids maximal, où les sommets sont les fenêtres d'une protéine et les arêtes les affinités. Ceci se fait en temps polynomial.

Nous avons effectué des validations croisées 10-folds pour chacun des codages proposés. Nous comparons ces résultats avec ceux d'un tirage aléatoire d'une configuration de ponts. Pour une protéine contenant n ponts, l'espérance mathématique du nombre de ponts correctement prédits par un choix aléatoire est $\frac{n}{2n-1}$. Ce résultat a été utilisé dans d'autres études sans jamais avoir été démontré. Nous l'avons prouvé mais notre démonstration est fastidieuse, aussi nous avons choisi de ne pas la faire figurer.

5.3 Résultats expérimentaux

Les performances des deux algorithmes sur les données biologiques sont maximales pour le codage croisé et très inférieures pour les autres codages. Nous donnons donc les résultats pour le codage croisé. Le tableau suivant présente les moyennes des résultats obtenus sur des séries de 100 expériences faites selon le protocole présenté section 5.3.

Nb de ponts/cystéines par protéine	2/4	3/6	4/8	5/10
Nb de protéines	51	50	28	20
Nb et % de ponts correctement prédits aléatoirement (espérance)	34 33,33%	30 20%	16 14,3%	11 11,1%
Nb et % de ponts correctement prédits Algorithme NB (supervisé)	41 40,2%	26,25 17,5%	14,22 12,7%	5,8 5,8%
Nb et % de ponts correctement prédits Algorithme 4 (semi-sup. asymétrique)	60 **58,8%**	50,1 **33,4%**	18,26 **16,3%**	13,2 **13,2%**

Résultats expérimentaux sur les données biologiques

Les résultats connus pour ce problème d'apprentissage sont (Fariselli & Casadio, 2001; Fariselli *et al.*, 2002; Vullo & Frasconi, 2004). Les meilleurs de ces résultats (Fariselli *et al.*, 2002) sont plus élevés que les notres. Ces résultats ont été obtenus par des méthodes plus sophistiquées (réseaux de neurones récursifs), avec plus de données, et en intégrant d'autres informations comme l'information évolutionnaire, c'est-à-dire un codage des segments selon des profils. Il est donc difficile de comparer nos résultats aux leurs, mais voici un tableau synthétique de leurs résultats à titre indicatif :

Nb de ponts par protéine	2 ponts	3 ponts	4 ponts	5 ponts
Nb de protéines	156	146	99	45
% de ponts correctement prédits	**73**	**56**	**37**	**30**

Résultats obtenus par (Fariselli *et al.*, 2002)

Néanmoins, nos résultats sont suffisants pour conclure sur deux points importants :
- l'apprentissage semi-supervisé asymétrique donne des résultats tout à fait satisfaisants (données artificielles et biologiques), ce qui nous encourage à poursuivre nos travaux dans cette voie ;
- notre hypothèse semble vérifiée : il est préférable de considérer les couples de cystéines non appariées comme des exemples non étiquetés plutôt que négatifs. Cette hypothèse devrait pouvoir être intégrée à des méthodes plus sophistiquées (réseaux de neurones, SVMs) de façon à exploiter au mieux l'information locale.

6 Conclusion

Nous montrons dans cet article que le problème de l'apprentissage semi-supervisé asymétrique lorsque les attributs descriptifs suivent l'hypothèse naïve de Bayes est bien posé. Nous fournissons un estimateur consistant qui permet d'identifier à la limite le modèle cible. Nous proposons également un algorithme itératif de construction de modèles basé sur le critère du maximum de vraisemblance.

Les résultats obtenus sur des données biologiques étayent une hypothèse biologique qui se veut originale quant à la façon de modéliser les données. Nous cherchons actuellement à appliquer ce procédé à d'autres données (brins beta en particulier) ainsi qu'à améliorer nos résultats sur les ponts disulfures en intégrant plus d'informations sur ces ponts. Nous essayons également d'intégrer les estimations d'affinités locales issues de notre méthode à d'autres méthodes d'apprentissage comme les SVM.

7 Remerciements

Je tiens à remercier François Denis (LIF, Marseille) et Cécile Capponi (LIF, Marseille - LMGM, Toulouse) pour leur aide et leurs conseils durant cette étude. Mais également Liva Ralaivola (LIF, Marseille), Christophe Geourjon (IBCP, Lyon) et Laurent Brehelin (LIRMM, Montpellier) pour leur participation et les diverses idées proposées.

Références

DEMPSTER A., N.M.LAIRD & D.B.RUBIN (1977). Maximum likelihood from incomplete data via the em algorithm. In *Journal of the Royal Statistical Society*, p. 39 :1–38.

DENIS F. (1998). Pac learning from positive statistical queries. In *The 9th International Workshop on Algorithmic Learning Theory*.

DENIS F., DECOMITE F., GILLERON R. & LETOUZEY F. (1999). Positive and unlabeled examples help learning. In *The 10th International Workshop on Algorithmic Learning Theory*.

DENIS F., GILLERON R., LAURENT A. & TOMMASI M. (2003). Text classification and co-training from positive and unlabeled examples. In *Proceedings of the ICML 2003 Workshop : The Continuum from Labeled to Unlabeled Data*, p. 80–87.

DOMINGOS P. & PAZZANI M. (1996). Simple bayesian classifiers do not assume independance. In A. P. . M. PRESS, Ed., *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*.

FARISELLI P. & CASADIO R. (2001). Prediction of disulfide connectivity in proteins. In *Bioinformatics*, number 17(10), p. 957–964.

FARISELLI P., MARTELLI P. & CASADIO R. (2002). A neural network-based method for predicting the disulfide connectivity in proteins. In *Proceedings of KES 2002, Knowledga based intelligent information engineering systems and allied technologies*, number 1, p. 464–468.

GEIGER D., HECKERMAN D., KING H. & MEEK C. (2001). Stratified exponential families : Graphical models and model selection. In *The Annals of Statistics*, number 29(2), p. 505–529.

HASTIE T., TIBSHIRANI R. & FRIEDMAN J. (2001). *The elements of statistical learning*.

LIU B. & LI X. (2003). Learning to classify text using positive and unlabeled data. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*.

MCCALLUM A., THRUN S. & MITCHELL T. (1999). Text classification from labeled and unlabeled documents using e.m.

VULLO A. & FRASCONI P. (2004). Disulfide connectivity prediction using recursive neural networks and evolutionary information. In *Bioinformatics*, number 20(5), p. 653–659.

WHILEY M. & TITTERINGTON D. (2002). Model identifiability in naive bayesian networks. In *Technical Report*.