Approximation sur les données : un problème de distances

Richard Lassaigne

IMJ/Logique mathématique

CNRS-Université Paris Diderot

Applications :

- Correction orthographique
- Reconnaissance de la parole
- Alignement de séquences génomiques
- Traduction automatique
- Traitement du langage naturel
- Recherche sur le WEB
- Traitement de données massives

Exemples de distance :

- Distance de **Hamming**
- Distance d'**édition** (ou de Levenshtein)
- Distance d'édition avec déplacement

Problème fondamental (en statistique et analyse de données) :

- Comment **tester** les propriétés des distributions de probabilités sous-jacentes aux données provenant d'expériences, de populations, bases de données...
- La quantité **énorme** de données à traiter est le principal obstacle à l'utilisation des méthodes classiques en statistique ou en apprentissage

Exemples de distance (entre ensembles ou entre distributions) :

- Divergence de **Kullback-Leibler** (pseudo-distance)
- Indice et distance de Jaccard
- Distance de **variation totale**
- Distances l_1, l_2
- **Earth Mover's Distance** (Kantorovich, Wasserstein)

- La **distance d'édition** est calculable en temps polynomial mais les algorithmes classiques sont impraticables sur de **grandes masses** de données
- Il est fort peu probable qu'il existe un algorithme exact en temps **sous-quadratique** à moins qu'une hypothèse *sérieuse* de complexité ne soit fausse
- Mais cela n'empêche pas une recherche importante sur les algorithmes d'**approximation**
- Les distances entre **distributions de probabilités** Distance l_1 et Earth Mover's Distance (EMD)
- Le **test** (probabiliste) de **propriétés** : Satisfaire une propriété ou être loin de la satisfaire
- Tests d'**identité**, de **proximité** ou d'**indépendance** pour les distributions de probabilités

Degré de **(dis)similarité** entre 2 chaînes de caractères

Exemple : **Alignement** de séquences de bases nucléiques (ADN)

x	A	G	G	C	T	A	T	C	A	C	C	T	G	A	C	C	T
T	A	G	x	C	T	A	T	C	A	C	x	x	G	A	C	C	G
i			d								d	d					s

La distance d'édition entre 2 chaînes de caractères est
le nombre minimum d'opérations d'édition (5)

- **insertion** (i)
- **suppression** (d)
- **substitution** (s)

pour transformer l'une des chaînes dans l'autre

2 chaînes de caractères et leur alignement avec 2 *gaps* (I,C)

I	N	T	E	x	N	T	I	O	N
x	E	X	E	C	U	T	I	O	N
d	s	s		i	s				

Si le **coût** de chaque opération est 1, la distance d'édition est 5
Si le coût de la **substitution** est 2 (Levensthein), elle est égale à 8

L'espace des suites d'opérations possibles peut être très grand mais la distance d'édition correspond à un plus court chemin

Pour 2 chaînes de caractères X et Y ($|X| = n$ et $|Y| = m$)
$D(i, j)$ est la distance d'édition entre le **préfixe** $X[1, ..., i]$ de **longueur** i et le **préfixe** $Y[1, ..., j]$ **de longueur** j
La distance d'édition entre X et Y est $D(n, m)$

Programmation **dynamique** :

La distance d'édition est obtenue par construction du **tableau** des distances d'éditions entre **préfixes**

Algorithme (Wagner et Fisher, 1974) :

- Pour $i = 1, \ldots, n$, $D[i, 0] := i$
- Pour $j = 1, \ldots, m$, $D[0, j] := j$
- Pour $i = 1, \ldots, n$

  Pour $j = 1, \ldots, m$

  Si $X[i-1] = Y[j-1]$, $\delta := 0$, sinon $\delta := 1$

$$D[i,j] := min \begin{cases} D[i-1,j]+1 \\ D[i,j-1]+1 \\ D[i-1,j-1]+\delta \end{cases}$$

- Retourner $D[n, m]$

Exemple : $X = INTENTION$ $Y = EXECUTION$

	ε	E	X	E	C	U	T	I	O	N
ε	**0**	1	2	3	4	5	6	7	8	9
I	**1**	1	2	3	4	5	6	6	7	8
N	2	**2**	2	3	4	5	6	7	7	7
T	3	3	**3**	3	4	5	5	6	7	8
E	4	3	4	**3**	**4**	5	6	x	7	8
N	5	4	4	4	4	**5**	6	7	7	7
T	6	5	5	5	5	5	**5**	6	7	8
I	7	6	6	6	6	6	6	**5**	6	7
O	8	7	7	7	7	7	7	6	**5**	6
N	9	8	8	8	8	8	8	7	6	**5**

Complexité

- Algorithme originel (Wagner et Fisher, 1974)
  Temps et espace **quadratiques** dans la longueur n
- Algorithme amélioré (Hirschberg, 1975)
  Temps quadratique et espace **linéaire**
- Algorithme exact le plus rapide (Masek et Paterson, 1980)
  Amélioration sur le temps (facteur **logarithmique**)
  Temps en $O(n^2/log\ n)$
- Algorithme différence (Myers, 1986)
  Temps en $O(n \times d)$ (d est la **distance d'édition**)
  Espace linéaire. Temps en $O(n + d^2)$ en moyenne

Dans le contexte des ensembles de données de (très) grande taille

- Problème 1 : Existence d'un algorithme exact fonctionnant en temps **sous-quadratique** ?

- Problème 2 : Conception d'algorithmes d'**approximation** efficaces pour la distance d'édition

Un algorithme A est un algorithme d'ε-**approximation** pour la distance D si pour toute entrée (X, Y)

$$\frac{|A(X,Y) - D(X,Y)|}{D(X,Y)} \leq \varepsilon$$

Remarque : ε peut être fonction de la taille n de l'entrée

- Algorithme de $\sqrt{n}$-**approximation** en temps **linéaire**

Conséquence facile de l'algorithme différence de Myers

- Algorithme de $n^{3/7}$-**approximation** en temps **quasi-linéaire** (Bar-Yossef, Jayram, Krauthgamer et Kumar, 2004)

- Algorithme de $n^{1/3+o(1)}$-**approximation** en temps $\tilde{O}(n)$ (T. Batu, F. Ergun et C. Sahinalp, 2006)

Remarque : la notation $\tilde{O}(f(n))$ signifie $f(n)\ .\ log^{O(1)}\ f(n)$

- Algorithme de $2^{\tilde{O}(\sqrt{log\ n})}$-**approximation** en temps **presque linéaire** (A. Andoni et K. Onak, 2009)

- Pour tout $\varepsilon > 0$, $(log\ n)^{O(1/\varepsilon)}$-**approximation** en temps $n^{1+\varepsilon}$ (modèle de requête asymétrique) (A. Andoni, R. Krauthgamer et K. Onak, 2010)

- Comment classifier les problèmes (vraiment) difficiles ?

- Un problème canonique **difficile** : le problème SAT
  Entrée : n variables propositionnelles $x_1, \ldots, x_n$
  une formule F conjonction de m **clauses** C_i où
  chaque clause est une k-disjonction de x_j ou $\neg x_j$
  Sortie : une valuation **satisfaisant** la formule F
  ou NON si la formule n'est pas satisfaisable

- Théorème (S. Cook, R. Karp, 1972)
  Le problème SAT est NP-**complet** pour $k \geq 3$
  S'il existe un algorithme résolvant le problème SAT en temps
  polynomial alors **tout problème** de NP l'est et ainsi $P = NP$

- Le meilleur algorithme connu pour le problème SAT
  fonctionne en temps $O(2^{n-(cn/k)} \ . \ n^d)$ (c, d constantes)
  Ce problème est conjecturé comme vraiment difficille

- R. Impagliazzo, R. Paturi et F. Zane (2001) ont proposé deux conjectures pour la **difficulté** du problème SAT

- **Strong Exponential Time Hypothesis (SETH)** : pour tout $\varepsilon > 0$, il existe un k tel que le problème SAT pour n variables et m clauses ne peut pas être résolu en temps $2^{(1-\varepsilon)n}\ .\ poly(m)$

- A. Backurs et P. Indyck (2015) ont montré un résultat de **borne inférieure** relativisé à l'hypothèse SETH
  Théorème : Si la distance d'édition peut être calculée en temps $O(n^{(2-\delta)})$ pour une constante $\delta > 0$, alors le problème SAT avec n variables et m clauses peut être résolu en temps $m^{O(1)}\ .\ 2^{(1-\varepsilon)n}$ (ε constante > 0)

- Exemple d'application : Systèmes d'extraction d'images
  Représentation par **histogrammes** à plusieurs dimensions
  Réponse à une requête dans une Base d'images :
  les images ayant les histogrammes les plus **proches**
  Mesure nécessaire de la **dissimilarité** entre histogrammes

- **Histogramme** $\{h_{\mathbf{i}}\}$: application $\mathbf{i} \longrightarrow h_{\mathbf{i}}$
  $\mathbf{i}$ vecteur de **dimension** d (représentant des couleurs, par ex.)
  $h_{\mathbf{i}}$ mesure de la **masse** de la distribution correspondante

- Mesures de dissimilarité entre 2 histogrammes $\{h_{\mathbf{i}}\}$ et $\{k_{\mathbf{i}}\}$:
  Les mesures **bin-by-bin** comparent $h_{\mathbf{i}}$ avec $k_{\mathbf{i}}$
  Les mesures **cross-bin** comparent $h_{\mathbf{i}}$ avec $k_{\mathbf{j}}$

Quelle est votre **distance** préférée ?

- Distance de Minkowski :

$$d_{L_p}(H,K) = \Big(\sum_i |h_{\mathbf{i}} - k_{\mathbf{i}}|^p\Big)^{1/p}$$

La **distance** L_1 est souvent utilisée pour la couleur

Dautres applications utilisent les distances L_2 ou L_∞

- Divergence de **Kullback-Leibler** (théorie de l'information) :

$$d_{KL}(H,K) = \sum_i h_{\mathbf{i}} \ . \ \log \frac{h_{\mathbf{i}}}{k_{\mathbf{i}}}$$

La divergence KL n'est pas symétrique et

est sensible au découpage de l'histogramme

Divergence de **Jeffrey** (stable, symétrique et robuste) :

$$d_J(H,K) = \sum_i \Big(h_{\mathbf{i}} \ . \ \log \frac{h_{\mathbf{i}}}{m_i} + k_{\mathbf{i}} \ . \ \log \frac{k_{\mathbf{i}}}{m_i}\Big) \text{ où } m_i = \frac{h_{\mathbf{i}} + k_{\mathbf{i}}}{2}$$

- Les distances *cross-bin* utilisent une distance **de base**
  entre les vecteurs caractéristiques utilisés dans l'histogramme
  EMD peut être défini comme le **coût minimal** payé
  pour **transformer** un histogramme dans un autre

- Un histogramme : une **masse de terre** dans un espace
  Un autre : une collection de **trous** dans le même espace
  EMD mesure le **travail minimum** pour remplir les trous

- Cas discret du problème de transport de Monge (1781)
  Etudié par Kantorovich (Prix Nobel d'économie 1975)

EMD est la solution d'un problème de **transport** représenté comme un problème de **flot minimal** :

Signatures $P = \{(x_1, p_1), \ldots, (x_n, p_n)\}$ et $Q = \{(y_1, q_1), \ldots, (y_m, q_m)\}$

Matrice des distances **de base** : $D = (d_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$

Déterminer le flot $F = (f_{ij})$ qui **minimise** le coût global

$$\sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} d_{ij} \text{ sous les } \textbf{contraintes}$$

$$f_{ij} \geq 0 \ (1 \leq i \leq n,\ 1 \leq j \leq m)$$

$$\sum_{j=1}^{n} f_{ij} \leq p_i \ (1 \leq i \leq m)$$

$$\sum_{i=1}^{m} f_{ij} \leq q_j \ (1 \leq j \leq n)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} = min(\sum_{i=1}^{n} p_i, \sum_{j=1}^{m} q_j)$$

Color-based Image Retrieval

L1 distance

Jeffrey divergence

Earth Mover Distance

- EMD est due à Y. Rubner, C. Tomasi et L.J. Guibas (2000)
  Algorithme **exponentiel** pour le problème de transport
  dans le pire des cas, **super-cubique** en **moyenne**
  L'algorithme de J. Orlin est en temps $O(N^3 logN)$

- EMD avec distance **de base** L_1 : H. Ling et K. Okada (2006)
  Algorithme **expérimentalement** en temps **quadratique**,

- P. Indyck et N. Thaper (2003) : **Approximation** de EMD-L_1
  par **plongement** dans l'espace $\mathbb{R}^d$ muni de la norme l_1
  Pour des ensembles de vecteurs caractéristiques $\subseteq [\Delta]^d$
  Le calcul du plongement est en temps $O(Nd log\Delta)$

- A. Andoni, P. Indyck et R. Krauthgamer (2007) : construction
  d'un **plongement** pour des sous-ensembles de taille s
  de $[\Delta]^d$ avec une **distorsion** en $O(log(s).log(d\Delta))$

- Problème : **estimation** de EMD entre 2 distributions avec accès seulement à des **échantillons** des distributions

- **Testeur** de proximité pour EMD :
  2 distributions P, Q sur un espace métrique M et $\varepsilon > 0$
  Algorithme A t.q. avec **forte probabilité** ($\geq 2/3$) :
  si $P = Q$, alors l'algorithme A **accepte**,
  si $EMD(P, Q) > \varepsilon$, alors l'algorithme A **rejette**

- Un **estimateur** avec **erreur additive** pour EMD :
  Algorithme qui, étant donné les mêmes entrées,
  retourne une valeur dans $[EMD(P, Q) - \varepsilon, EMD(P, Q) + \varepsilon]$

- La mesure de **complexité** est la **taille** de l'**échantillonnage**

[K.D. Ba, H.L. Nguyen, H.N. Nguyen et R. Rubinfeld, 2009]

Testeur de proximité pour EMD :

- Contexte : 2 distributions P, Q sur $M \subseteq [0,1]^d$
  Considérer une grille sur $[0,1]^d$ de pas $\frac{1}{2^i}$
  et les **approximations** grossières de P, Q sur cette grille
- Le testeur pour EMD utilise un nombre $\log(2d/\varepsilon)$ fois
  un testeur des approximations pour la **distance** l_1
  La complexité en **échantillons** est $\tilde{O}((2d/\varepsilon)^{2d/3})$

Estimateur pour EMD : Algorithme $A(P, Q, \varepsilon)$

- Soit G la grille sur $[0,1]^d$ de pas $\frac{\varepsilon}{4d}$
  et P', Q' les distributions **induites** par P, Q
- Prendre $O((\frac{4d}{\varepsilon})^{d+2})$ échantillons pour P' et Q'
  et soit $\hat{P}', \hat{Q}'$ les distributions **empiriques** résultantes
- Retourner $EMD(\hat{P}', \hat{Q}')$

La complexité en **échantillons** est $O((\frac{4d}{\varepsilon})^{d+2})$

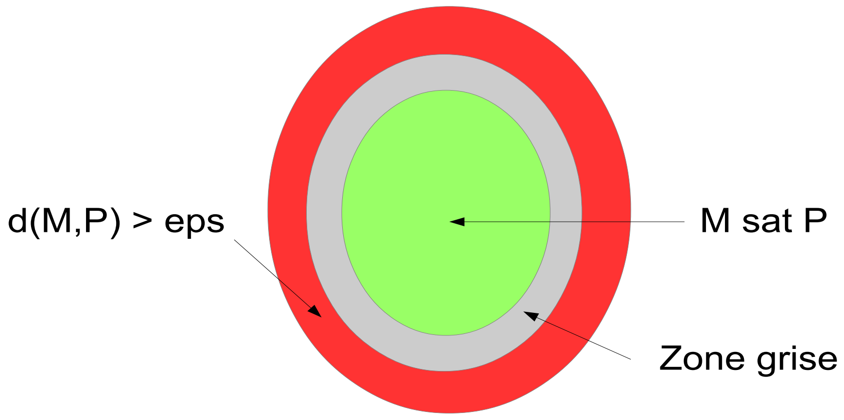
Test (probabiliste) de Propriétés

Satisfaction **classique** : $\mathcal{M} \models \mathbf{P}$

$\mathcal{M}$ satisfait la propriété $\mathbf{P}$

Satisfaction **approchée** :

$\mathcal{M} \models_{\varepsilon} \mathbf{P}$ si $\mathcal{M}$ est ε-proche de $\mathcal{M}'$ tel que $\mathcal{M}' \models \mathbf{P}$



Testeur : Algorithme **probabiliste** A

- si $\mathcal{M} \models \mathbf{P}$, alors A **accepte**
- si $\mathcal{M}$ est **ε-loin** de P, alors A **rejette** avec forte probabilité

  Le temps de calcul peut être **indépendant** de $|\mathcal{M}|$
  mais dépend de $1/\varepsilon$

Exemple : Pour la distance de **Hamming** entre les mots l'appartenance d'un mot à un langage **régulier** est testable avec un nombre de **requêtes** en $O(\log^3(1/\varepsilon)/\varepsilon)$
(N. Alon, M. Krivelevich, I. Newman et M. Szegedy, 2000)

Etant donné des **échantillons** obtenus à partir d'une (ou plus) distribution **inconnue**, décider si elle satisfait une propriété.

- Problème classique en **statistique**
  (Neymann, Pearson, 1933 ; Lehmann, Romano, 2005)

- **Test de propriétés** (informatique théorique depuis 2000)
  (Goldreich, Ron, 2000 ; Batu et al, FOCS 2000)

- Test d'**identité** (distribution inconnue / distribution connue)
  Test de **proximité** pour 2 distributions inconnues
  Lecture Notes for Testing Properties of Distributions
  (O. Goldreich, 2016)

Lemme (Chan, Diakonikolas, Valiant et Valiant, 2014) :

Soient P, Q des distributions **inconnues** sur un domaine de taille n

Il existe un algorithme qui :

- sur l'entrée $n, \varepsilon > 0$ et $b \geq max(||P||_2, ||Q||_2)$,
- utilise $O(bn/\varepsilon^2)$ **échantillons** des distributions P et Q
- et distingue, avec probabilité $\geq 2/3$, entre les cas $P = Q$ et $||P - Q||_2 \geq \varepsilon/\sqrt{n}$

Remarque :

- Si $||P||_2$ et $||Q||_2)$ sont **petites**, alors le test est **efficace**
  Si $||P||_2 = ||Q||_2 = O(1/\sqrt{n})$, la complexité est en $O(\sqrt{n}/\varepsilon^2)$
- En fait, il suffit que l'une des deux soit petite car il est facile de détecter une **grande différence** entre les deux

Idée principale : **Split distribution** et **Poissonisation**

- P distribution et S multi-ensemble de $[n]$
  Split distribution P_S associée à la distribution P :
  Soit $a_i = 1+$ le nombre d'éléments de S égaux à i
  Correspondance entre $[n + |S|]$ et $B = \{(i, j) : i \in [n], 1 \leq j \leq a_i\}$
  Distribution P_S à support B : $i \in_r P$ et $j \in_r [a_i]$

- Lemme : Soit P une distribution sur $[n]$
  Pour tous multi-ensembles $S \subseteq S'$ de $[n]$, $||P_{S'}||_2 \leq ||P_S||_2$
  Si S est obtenu en prenant $Poisson(m)$ **échantillons** de P, alors

$$\mathbb{E}[||P_S||_2^2] \leq 1/m$$

P une distribution **inconnue** et Q une distribution **donnée** sur $[n]$

P_S, Q_S split distributions associées à P, Q relativement à S

Propriétés :

- On peut simuler un échantillon de P_S ou Q_S à partir d'un échantillon de P ou Q
- Les différences en norme l_1 sont conservées : $||P_S - Q_S||_1 = ||P - Q||_1$

Testeur d'identité pour la norme l_1 :

- Etant donné Q, construire le multi-ensemble S qui contient $\lfloor nq_i \rfloor$ copies de i
- Utiliser le **testeur de base** pour distinguer entre $P_S = Q_S$ et $||P_S - Q_S||_1 \geq \varepsilon$

Analyse : $n + |S| \leq 2n$ et $||Q_S||_2 = O(1/\sqrt{n})$

Le **test d'identité** entre P_S et Q_S s'effectue en

$O(||Q_S||_2|S|/\varepsilon^2)$ c'est-à-dire $O(\sqrt{n}/\varepsilon^2)$

Difficulté : La distribution Q n'est pas connue
On va utiliser un nombre approprié d'échantillons de Q
pour définir l'ensemble de split S

Testeur de proximité pour la norme l_1 :

- Soit $k = min\{n, n^{2/3}\varepsilon^{-4/3}\}$
- Prendre $Poisson(k)$ **échantillons** de Q pour définir S
- Utiliser le **testeur de base** pour distinguer entre $P_S = Q_S$ et $||P_S - Q_S||_1 \geq \varepsilon$

Analyse : Avec forte probabilité, $|S| = O(n)$ et $||Q_S||_2 = O(1/\sqrt{k})$
Le **testeur de base** utilise $O(nk^{-1/2}/\varepsilon^2)$ échantillons
Le nombre total d'échantillons est en
$O(k + nk^{-1/2}/\varepsilon^2) = O(max\{n^{2/3}\varepsilon^{-4/3}, \sqrt{n}/\varepsilon^2\})$

- La **distance d'édition** est utilisée en recherche d'information
  L'algorithme originel (DP) est en temps **quadratique**
  Amélioration **en moyenne** (recherche de plus court chemin)
- **Approximation** linéaire résultant de cette amélioration
  Compromis entre la **qualité** de l'approximation et le **temps**
- Peu d'espoir pour un algorithme en temps **sous-quadratique**
  Recherche importante sur la comparaison **expérimentale**
  G. Navarro : *A guided tour to approximate string matching*

- Une distance adaptée à la comparaison entre distributions
  Earth Mover's Distance rel. à une distance de base
  Algorithme **super-cubique** en moyenne

- Le **Test de Propriétés** : une méthode pour obtenir des
  Algorithmes **probabilistes** en temps **sous-linéaire**

- A. Andoni, R. Krauthgamer and K. Onak. *Polylogarithmic Approximation for Edit Distance and the Asymmetric Query Complexity*. Proc. 51th Symposium on Foundations of Computer Science, 2010
- K.D. Ba, H.L. Nguyen, H.N. Nguyen and R. Rubinfeld. *Sublinear Time Algoritms for Earth Mover's Distance*. Theory of Computing Systems, 48(2),p.428-442, 2011
- A. Backurs and P. Indyck. *Edit Distance cannot be computed in Strongly Subquadratic Time unless SETH is false*. Proc. 47th Symposium on Theory of Computing, 2015
- S. Chan, I. Diakonikolas, P. Valiant and G. Valiant. *Optimal Algorithms for Testing Closeness of Discrete Distributions*. Proc. 25th ACM-SIAM Symposium on Discrete Algorithms, p. 1193-1203, 2014

- I. Diakonikolas and D. Kane. *A New Approach for Testing Properties of Discrete Distributions*. arXiv :1601.05557, 2016
- W.J. Masek and M.S. Paterson. *A faster algorithm computing string edit distances*. Journal of Computer and System Sciences 20(1), p.18-31, 1980
- E.W. Myers. *An O(ND) difference algorithm and its variants*. Algorithmica 1, p. 251-266, 1986
- G. Navarro. *A guided tour to approximate string matching* ACM Computing Surveys 33(1), p. 31-88, 2001
- Y. Rubner, C. Tomasi and L.J. Guibas. *The Earth Movers's Distance as a Metric for Image Retrieval*. Int. Journal of Computer Vision 40 (2), p. 99-121, 2000
- R.A. Wagner and M.J. Fisher. *The String-to-String Correction Problem*. Journal of ACM 21, 1, p. 168-173, 1974