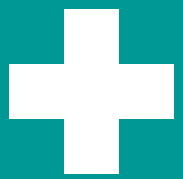
femto-st SCIENCES & TECHNOLOGIES

Sécurité Appliquée Protection de la vie privée-PVP Apprentisages machine et Confidentialité différentielle

Jean-François Couchot
Université de Franche-Comté, UFR-ST

04/12/23



cnrs UBFC UNIVERSITÉ BOURGOGNE-FRANCHE-COMTÉ UNIVERSITÉ DE FRANCHE-COMTÉ SUP MICRO TECH ENSMM utbm université de technologie Belfort-Montbéliard

Plan

Plan

Apprentissage supervisé confidentiellement privé

- Classification bayésienne
- Classification bayésienne ϵ-DP
- Régression linéaire multiple
- Régression linéaire ϵ-DP

Apprentissage non suppervisé confidentiellement privé

Plan

Rappel : apprentissage suppervisé probabiliste

Exemple [1] de classification

- Connaissant une valeur pour chaque attribut de mesure : predire Outcome
- Comparer les probabilités suivantes et conclure :

$$\Pr[\text{OutCome} =' \text{YES}' | \text{Preg} = p, \text{Gluc} = g, \ldots, \text{Age} = a]$$
$$\Pr[\text{OutCome} =' \text{NO}' | \text{Preg} = p, \text{Gluc} = g, \ldots, \text{Age} = a]$$

- A évaluer : $\Pr[Y_1 | X_1, X_2, \ldots, X_d]$ et $\Pr[Y_2 | X_1, X_2, \ldots, X_d]$
- Remarque : attributs discrets (Preg., Gluc, ...) ou réels (BMI, DPF)

1. https ://www.kaggle.com/uciml/pima-indians-diabetes-database

Rappel : théorème de Bayes

Théorème (Théorème de Bayes)

$$\Pr[Y|X_1,\ldots,X_d] = \frac{\Pr[Y]\times\Pr[X_1,\ldots,X_d|Y]}{\Pr[X_1,\ldots,X_d]}$$

Simplifications immédiates

$$\left.\begin{array}{l}\Pr[Y_1|X_1,\ldots,X_d] = \frac{\Pr[Y_1]\times\Pr[X_1,\ldots,X_d|Y_1]}{\Pr[X_1,\ldots,X_d]}\\ \Pr[Y_2|X_1,\ldots,X_d] = \frac{\Pr[Y_2]\times\Pr[X_1,\ldots,X_d|Y_2]}{\Pr[X_1,\ldots,X_d]}\end{array}\right\} \begin{array}{l}\text{m̂ dénom. } \Pr[X_1,\ldots,X_d]\\ \rightsquigarrow \text{ son calcul : inutile}\end{array}$$

A évaluer : $\Pr[Y_j]$, et $\Pr[X_1,\ldots,X_d|Y_j]$

- $\Pr[Y_j]$: fréquence d'apparition de la valeur Y_j pour l'attribut Y
- $\Pr[X_1,\ldots,X_d|Y_j] = \Pr[X_1|Y_j]\times\cdots\times\ \Pr[X_d|Y_j] = \prod_{i=1}^{d}\Pr[X_i|Y_j]$:
  - Hypothèse naïve : indépendance de chaque X_i p.r. aux autres $X_{i'}$, $i'\neq i$, conditionnellement à Y_j

Exemple avec des données discrètes-1

Données [2], question et premières probabilités

Age	Income	Gender	Missed Payment
Young	Low	Male	Yes
Young	High	Female	Yes
Medium	High	Male	No
Old	Medium	Male	No
Old	High	Male	No
Old	Low	Female	Yes
Medium	Low	Female	No
Medium	Medium	Male	Yes
Young	Low	Male	No
Old	High	Female	No

- Q : défaut de payement pour une jeune femme avec un revenu moyen ?
- $Y_1 \equiv'$ Missed Payment$' ='$ YES$'$, $Y_2 \equiv'$ Missed Payment$' ='$ NO$'$
- Comparer $\Pr[Y_1|\text{Age} = \text{Young}, \text{Income} = \text{Medium}, \text{Gender} = \text{Female}]$ et $\Pr[Y_1|\text{Age} = \text{Young}, \text{Income} = \text{Medium}, \text{Gender} = \text{Female}]$
- $\Pr[Y_1] = \frac{4}{10}$ et $\Pr[Y_2] = \frac{6}{10}$

Probabilités pour chaque attribut conditionnellement à Y_j

- $\Pr[\text{Age} = \text{Young}|Y_1] = \frac{2}{4}$, $\Pr[\text{Age} = \text{Young}|Y_2] = \frac{1}{6}$ ...
- $\Pr[\text{Income} = \text{Medium}|Y_1] = \frac{1}{4}$, $\Pr[\text{Income} = \text{Medium}|Y_2] = \frac{1}{6}$ ...
- $\Pr[\text{Gender} = \text{Female}|Y_1] = \frac{2}{4}$, $\Pr[\text{Gender} = \text{Female}|Y_2] = \frac{2}{6}$ ...

2. Yilmaz, E., Al-Rubaie, M., & Chang, J. M. (2019). Locally differentially private naive bayes classification. arXiv preprint arXiv:1905.01039.

Exemple avec des données discrètes-2

Evaluation de $\Pr[Y_j|\text{Age} = \text{Young}, \text{Income} = \text{Medium}, \text{Gender} = \text{Female}]$

- $Pr[Y_1] \times \Pr[\text{Age} = \text{Young}|Y_1] \times \Pr[\text{Income} = \text{Medium}|Y_1] \times \Pr[\text{Gender} = \text{Female}|Y_1] = \frac{4}{10} \times \frac{2}{4} \times \frac{1}{4} \times \frac{2}{4} = \frac{1}{40} \approx 0.025$
- $Pr[Y_2] \times \Pr[\text{Age} = \text{Young}|Y_2] \times \Pr[\text{Income} = \text{Medium}|Y_2] \times \Pr[\text{Gender} = \text{Female}|Y_2] = \frac{6}{10} \times \frac{1}{6} \times \frac{1}{6} \times \frac{2}{6} = \frac{1}{180} \approx 0.0056$

Réponse

La probabilité qu'elle ratte son paiement est beaucoup plus importante que celle opposée.

Exemple avec des données continues-1

Données [3], question et premières probabilités

sexe	taille (cm)	masse (kg)	point. (cm)
masc.	182	81.6	30
masc.	180	86.2	28
masc.	170	77.1	30
masc.	180	74.8	25
fém.	152	45.4	15
fém.	168	68.0	20
fém.	165	59.0	18
fém.	175	68.0	23

- Q : sexe d'une personne mesurant 183cm, pesant 59kg et dont les pieds mesurent 20cm ?
- $Y_1 \equiv'$ Sexe$' ='$ masc.$'$, $Y_2 \equiv'$ Sexe$' ='$ fém.$'$
- Comparer $\Pr[Y_1|\text{Taille} = 183, \text{Masse} = 59, \text{Point.} = 20]$ et $\Pr[Y_2|\text{Taille} = 183, \text{Masse} = 59, \text{Point.} = 20]$ et
- $\Pr[Y_1] = \Pr[Y_2] = \frac{1}{2}$

Probabilités pour chaque attribut X_i conditionnellement à Y_j : $\mathcal{N}(\mu_{i,j}, \sigma^2_{i,j})$

1. Calcul des paramètres de $\mathcal{N}$: moyene $\mu_{i,j}$ et variance $\sigma^2_{i,j}$

Sexe	μ_{taille}	σ^2_{taille}	μ_{masse}	σ^2_{masse}	$\mu_{point.}$	$\sigma^2_{point.}$
masc.	178	29.3	79.9	25.5	28.25	5.58
fém.	165	92.7	60.1	114	19	11.3

2. Avec la densité de probabilité $\frac{1}{\sqrt{2\pi\sigma^2_{i,j}}} \exp\left(-\frac{1}{2\sigma^2_{i,j}}(x-\mu_{i,j})^2\right)$, calcul de

$$\Pr[\text{taille} = 183|Y_1] = \frac{1}{\sqrt{2\pi \times 29.3}} \exp\left(\frac{-1}{2\times 29.3}(183-178)^2\right) dt \approx 0.0481$$

3. Classification naïve bayésienne, Wikipedia

Exemple avec des données continues-2

Valeurs numérique des robabilités pour chaque attribut X_i conditionnellement à Y_j

- $\Pr(taille = 183|Y_1) = 0.0481dt$, $\Pr(poids = 59|Y_1) = 0.0000146dp$ et $\Pr(point. = 20|Y_1) = 0.000381d$.
- $\Pr(taille = 183|Y_2) = 0.00721dt$, $\Pr(poids = 59|Y_2) = 0.0372dp$ et $\Pr(point. = 20|Y_2) = 0.114d$.

Evaluation de $\Pr[Y_j|taille = 183, poids = 59, point. = 20]$

- $Pr[Y_1] \times \Pr(taille = 183|Y_1) \times \Pr(poids = 59|Y_1) \times \Pr(point. = 20|Y_1) \approx 1.3404 \times 10^{-10}$
- $Pr[Y_2] \times \Pr(taille = 183|Y_2) \times \Pr(poids = 59|Y_2) \times \Pr(point. = 20|Y_2) \approx 1.52 \times 10^{-5}$

Réponse

La personne est probablement une femme.

Plan

Evaluer $\Pr[Y_j]$ et $\Pr[X_i|Y_j]$

Evaluer $\Pr[Y_j]$

- Histogramme des Y_j à construire
- Algorithme ϵ-DP pour les histogrammes : bruit laplacien de sensibilité 1.

Evaluer $\Pr[X_i|Y_j]$ pour un attribut X_i discret

- Pour chaque sortie Y_j à prédire : construction de l'histogramme de X_i
- Algorithme ϵ-DP pour les histogrammes : bruit laplacien de sensibilité 1.

Evaluer $\Pr[X_i|Y_j]$ pour un attribut X_i continu

- Pour chaque sortie Y_j à prédire : approché par une loi $\mathcal{N}(\mu_{i,j}, \sigma^2_{i,j})$
- Algorithme ϵ-DP pour la moyenne : bruit laplacien de sensibilité $\frac{U-L}{n+1}$ avec U la borne max, l la borne min, n le nbre de lignes déjà présentes
- Algorithme ϵ-DP [4] pour la variance σ^2 : bruit laplacien de sensibilité $\frac{n(U-L)}{n+1}$ avec U la borne max, l la borne min, n le nbre de lignes déjà présentes

4. Vaidya, Jaideep, et al. "Differentially private naive bayes classification." 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). Vol. 1. IEEE, 2013.

Algorithme de Vaidya

Algorithm 1 Computing differentially private parameters for Naïve Bayes

Require: ϵ, the privacy parameter for differential privacy
Require: Laplace(a, b) samples the Laplace distribution with mean a and scale b

for each attribute X_j **do**
  if X_j is categorical **then**
    sensitivity, $s \leftarrow 1$
    scale factor, $sf \leftarrow s/\epsilon$
    $\forall$ counts n_{kj}, $n'_{kj} = n_{kj}+$ Laplace$(0, sf)$
    Use n'_{kj} to compute $P(x_i|c_j)$
  else if X_j is numeric **then**
    compute sensitivity, s for mean μ_j as per equation 5
    scale factor, $sf \leftarrow s/\epsilon$
    $\mu'_j \leftarrow \mu_j+$ Laplace$(0, sf)$
    compute sensitivity, s for standard deviation σ_j as per equation 7
    scale factor, $sf \leftarrow s/\epsilon$
    $\sigma'_j \leftarrow \sigma_j$ + Laplace$(0, sf)$
    Use μ'_j and σ'_j to compute $P(x_i|c_j)$
  end if
end for
for each class c_j **do**
  count $nc'_j \leftarrow nc_j+$ Laplace$(0, 1)$
  Use nc'_j to compute the prior $P(c_j)$
end for

Remarques et implantation

Calcul du budget ϵ global

- A partager entre tous les attributs : chacun en consomme
- $\epsilon_i = \frac{\epsilon}{d+1}$: les attributs $X_1, \ldots, X_d$ et Y

Avec la biblotheque DiffprivLib [5]

```
# salaire sup à 50k$
import pandas as pd
from sklearn.model_selection import train_test_split
import diffprivlib.models as dp

...

def experimentDPGNB(eps,X,y):
    X_train, X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
    private_clf = dp.models.GaussianNB(epsilon=eps)
    private_clf.fit(X_train.values, y_train.values.ravel())
    y_pred = np.array(private_clf.predict(X_test))
    return accuracy_score(y_test,y_pred)
```

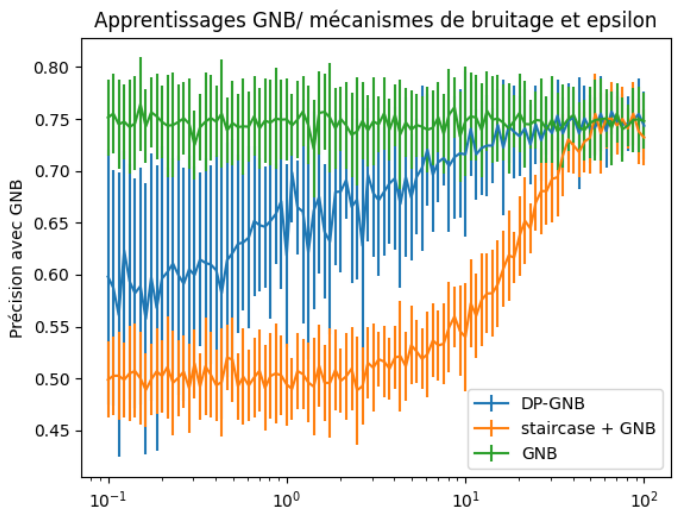
5. https://pypi.org/project/diffprivlib/

GNB vs $\mathcal{M}_{\text{Stair}}$ + GNB vs DP-GNB

Contexte expérimental

- Sur le jeu de données du diabete
- Outcome appris par GNB, par $\mathcal{M}_{\text{Stair}}$ + GNB, par DP-GNB
- Pour chaque $\epsilon \in [10^{-1}, 100]$ global, moyenne sur 20 apprentissages
  - GNB : sans PVP
  - $\mathcal{M}_{\text{Stair}}$ + GNB : nettoyage staircase + apprentissage
  - DP-GNB : `dp.models.GaussianNB`
- Affiché : précision moyenne de l'apprentissage et écart-type

Interprétation des résultats



- GNB : toujours plus précis (mais fuite de données)
- DP-GNB : toujours plus précis que $\mathcal{M}_{\text{Stair}}$ + GNB
- $\epsilon \in [0.1, 10]$: $\mathcal{M}_{\text{Stair}}$ + GNB proche de l'aléatoire
- $\epsilon \in [10, 30]$: intérêt de DP-GNB p.r. à GNB ?

Plan

Introduction puis formalisation

Exemple[6] de régression

Preg.	Gluc.	BloodP.	SkinThick.	Insul.	BMI	DPF	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
			...					

- Connaissant une valeur pour chaque attribut numérique $X_1, \ldots, X_d$, de mesure : predire Outcome, $Y = Y_i$

Formalisation

- Base $D = \{(X_{11}, \ldots, X_{1d}, Y_1), \ldots, (X_{n1}, \ldots, X_{nd}, Y_n)$ de n tuples $t_1, \ldots t_n$
- Hypothèse : $Y_i = \omega_0 + \omega_1 X_{i1} + \omega_2 X_{i2} + \cdots + \omega_d X_{id} + e_i$, avec $i = 1, \ldots, n$
- e_i : erreur dans l'explication linéaire de Y_i à partir des $(X_{i1}, \ldots, X_{id})$
- $\omega = (\omega_0, \omega_1, \ldots, \omega_d)$: paramètres à estimer en minimisant $e_1, \ldots, e_n$

6. https ://www.kaggle.com/uciml/pima-indians-diabetes-database

Notation matricielle

Formalisation

$$\begin{cases} y_1 = \omega_0 + \omega_1 x_{1,1} + \ldots + \omega_d x_{1,d} + e_1 \\ y_2 = \omega_0 + \omega_1 x_{2,1} + \ldots + \omega_d x_{2,d} + e_2 \\ \ldots \\ y_n = \omega_0 + \omega_1 x_{n,1} + \ldots + \omega_d x_{n,d} + e_n \end{cases} \Leftrightarrow$$

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{1,1} & \cdots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,d} \end{pmatrix} \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_d \end{pmatrix} + \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix}$$

De manière compacte $y = X\omega + e$ avec :

- y de dimension : $(n, 1)$
- X de dimension : $(n, d+1)$
- ω de dimension : $(d+1, 1)$
- e de dimension : $(n, 1)$

Estimateur des moindres carrés ordinaires (MCO)

Objectif

- Modèle complet initial : $y_i = \omega_0 + \omega_1 x_{i,1} + \cdots + \omega_d x_{i,d} + e_i$
- Estimation finale des paramètres : $\hat{y}_i = \hat{\omega}_0 + \hat{\omega}_1 x_{i,1} + \cdots + \hat{\omega}_d x_{i,d}$
- Résidus estimés : différence entre la valeur de y observée et estimée

$$\hat{e}_i \equiv y_i - \hat{y}_i$$

Quelles valeurs de $\omega_0, \ldots, \omega_d$ minimisent la somme des carrés des résidus ?

- $f_D(\omega) = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_i - (\omega_0 + \omega_1 x_{i,1} + \cdots + \omega_d x_{i,d})\right)^2$ à minimiser
- Rechercher des solutions de $\frac{\partial(\sum \hat{e}_i^2)}{\partial \omega_j} = 0$, ($j = d + 1$ équations)
- Solution (facile) : $\hat{\omega} = (X^T X)^{-1} X^T Y$, X^T la transposée de X
- Attention : publier $\hat{\omega}$ fuite de l'information sur X et Y

Exemple

A partir des données $D = \{(1, 0.4), (0.9, 0.3), (-0.5, -1)\}$

- $y = X\omega + e$ avec $y = \begin{pmatrix} 0.4 \\ 0.3 \\ -1 \end{pmatrix}$, $X = \begin{pmatrix} 1 & 1 \\ 1 & 0.3 \\ 1 & -0.5 \end{pmatrix}$, $\omega = \begin{pmatrix} \omega_0 \\ \omega_1 \end{pmatrix}$ $e = \begin{pmatrix} e_0 \\ e_1 \end{pmatrix}$
- Fonction MCO : $f_D(\omega) = 3\omega_0^2 + 2.8\omega_0\omega_1 + 0.6\omega_0 + 2.06\omega_1^2 - 2.34\omega_1 + 1.25$
- Estimateur MCO : $\hat{\omega} = \begin{pmatrix} \hat{\omega}_0 \\ \hat{\omega}_1 \end{pmatrix} = (X^T X)^{-1} X^T Y = \begin{pmatrix} -\frac{564}{1055} \\ \frac{393}{422} \end{pmatrix}$
- Vérifications :
  - $\hat{y}_1 = -\frac{564}{1055} + 1\frac{393}{422} = \frac{837}{2110} \approx 0.397.$
  - $\hat{y}_2 = -\frac{564}{1055} + 0.9\frac{393}{422} = \frac{1280}{4220} \approx 0.306.$
  - $\hat{y}_3 = -\frac{564}{1055} - 0.5\frac{393}{422} = \frac{4221}{4220} \approx -1.0002.$
- Valeur minimum de $f_D(\omega)$: $\approx 2.36E - 5$

Plan

Algorithme [7] ϵ-DP

Prétraitement de normalisation

- Normalisation des valeurs pour chaque attribut jusqu'à ce que $\sqrt{\sum_{i=1}^{d} x_{i,d}^2} \leq 1$, pour $1 \leq i \leq n$
- Normalisation des valeurs de yi : doivent appartenir à $[-1, 1]$

Quoi bruiter ?

- Bruit laplacien ajouté à $\hat{\omega}$: sensibilité non bornée !
- Bruit laplacien à chaque coefficient de la fonction MCO $f_D(\omega)$: sensibilité bornée par $\Delta = 2(1 + d)^2$ (admis)

Mécanisme qui bruite $f_D(\omega)$

Le mécanisme qui ajoute un bruit laplacien centré d'échelle $\frac{\Delta}{\epsilon}$, $\Delta = 2(1 + d)^2$, aux coefficients de $f_D(\omega)$ est ϵ-DP (admis).

7. Zhang, J., Zhang, Z., Xiao, X., Yang, Y., & Winslett, M. (2012). Functional mechanism : regression analysis under differential privacy. arXiv preprint arXiv :1208.0219.

Exemple (suite)

A partir des données $D = \{(1, 0.4), (0.9, 0.3), (-0.5, -1)\}$

- Fonction MCO : $f_D(\omega) = 3\omega_0^2 + 2.8\omega_0\omega_1 + 0.6\omega_0 + 2.06\omega_1^2 - 2.34\omega_1 + 1.25$
- $d = 1$, $\Delta = 2(1 + d)^2 = 8$, $\epsilon = 10$
- $\overline{f_D}(\omega) = 1.39\omega_0^2 + 1.18\omega_0\omega_1 - 0.29\omega_0 + 2.14\omega_1^2 - 2.54\omega_1 + 1.25$
- Estimateur MCO : $\hat{\omega} = \begin{pmatrix} -0.17 \\ 0.64 \end{pmatrix}$
- Vérifications :
  - $\hat{y}_1 = -0.17 + 1 \times 0.64 \approx 0.47$
  - $\hat{y}_2 = -0.17 + 0.9 \times 0.64 \approx 0.41$
  - $\hat{y}_3 = -0.17 - 0.5 \times 0.64 \approx -0.49$

Avec Diffprivlib [8]

```
def experimentDPLR(eps,X,y):
    X_train, X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)

    private_clf = dp.models..LinearRegression(epsilon=eps)
    private_clf.fit(X_train.values, y_train.values.ravel())

    y_pred = np.array([0 if a_ <0.5 else 1 for a_ in reg.predict(X_test.values)])
    return accuracy_score(y_test,y_pred)
```

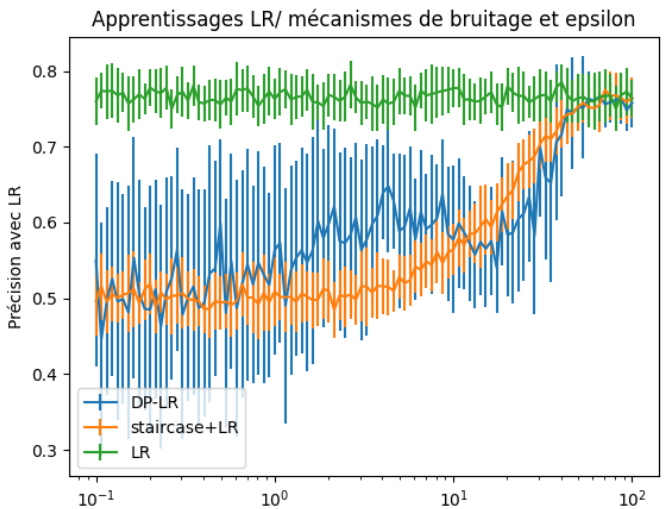
8. https://github.com/IBM/differential-privacy-library/blob/main/notebooks/linear_regression.ipynb

LR vs $\mathcal{M}_{\text{Stair}}$ + LR vs DP-LR

Contexte expérimental

- Sur le jeu de données du diabete
- Outcome appris par LR, par $\mathcal{M}_{\text{Stair}}$ + LR, par DP-LR
- Pour chaque $\epsilon \in [10^{-1}, 100]$ global, moyenne sur 20 apprentissages
  - LR : sans PVP
  - $\mathcal{M}_{\text{Stair}}$ + LR : nettoyage staircase + apprentissage
  - DP-LR : `dp.models.LinearRegression`
- Affiché : précision moyenne de l'apprentissage et écart-type

Interprétation des résultats



- LR : toujours plus précis (mais fuite de données)
- DP-LR : variance élevée
- $\epsilon \in [0.1, 10]$: DP-LR et $\mathcal{M}_{\text{Stair}}$ + LR $\approx$ aléatoire
- $\epsilon \in [10, 30]$: $\mathcal{M}_{\text{Stair}}$ + LR plus précis

Plan

Plan

Rappels

Regroupement par k-moyenne

- Soit $D = \{x^1, x^2, \ldots, x^N\}$ un ensemble de données avec d attributs, *i.e.* chaque $x^j = (x_1^j, \ldots, x_d^j)$, $1 \leq j \leq N$ est de dimension d.
- Objectif : partitioner D en k groupes O^1, ..., O^k, de centroids o^1, ..., o^k minimisant $\sum_{j=1}^{k} \sum_{x \in O^j} ||x - o^j||^2$
- Publication : centroids $\{o^1, \ldots, o^k\}$ et cardinalités $|O^1|$, $|O^2|$, ..., $|O^k|$

Algorithme simplifié

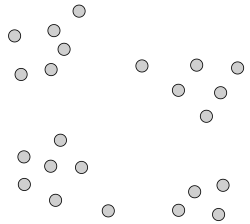
1. choisir des centroïds initiaux o^1, ..., o^k (au hasard ?)
2. répéter jusqu'à convergence, au plus t fois :
   - 2.1 associer chaque x au groupe O^j de centroïd o^j le plus proche :

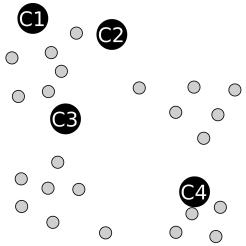
$$O^j = \left\{ x : \left\| x - o^j \right\| \leq \left\| x - o^{j'} \right\| \ \forall\, j' = 1, \ldots, k, j' \neq j \right\}$$

   - 2.2 m.-à-j. du centroïd de chaque groupe : $o^j = \frac{1}{|O^j|} \sum_{x \in O^j} x$

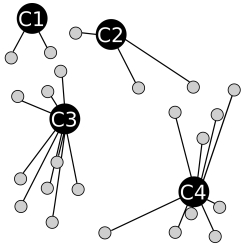
Exemple, synthèse

Exemple visuel [9]

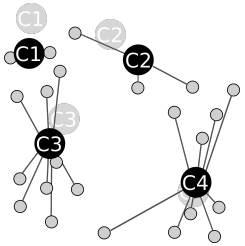

0a. Données d'entrée



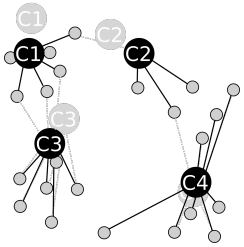
0b. intialisation



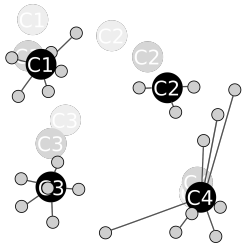
1a. assignation



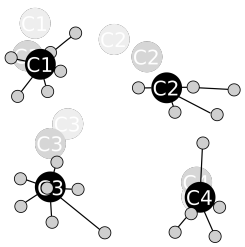
1b. calcul des points moyens



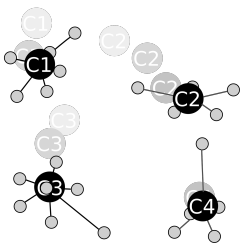
2a. assignation



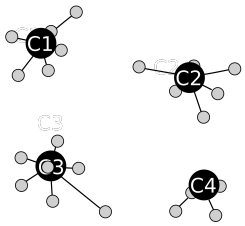
2b. calcul des points moyens



3a. assignation



3b. calcul des points moyens



4a. assignation
clusters stables (fin)

Points clefs en terme de fuite de données

1. choix des centroïds initiaux au hasard : pas de fuite
2. association des x au groupe O^j : calcul interne, pas de fuite
3. m.-à-j. (publications) des o^j avec $\frac{1}{|O^j|}\sum_{x\in O^j} x$ et cardinalités ($|O^j|$) : fuite

9. Wikipedia: K-moyennes

Plan

Sensibilité et budget ϵ

Sensibilités pour chaque itération

- Comptage $Q_C(D) = (|O^1|, \ldots, |O^k|)$:
  - $D' = D \cup \{x\}$: augmente de 1 une des cardinalités $|O^j|$, $1 \leq j \leq k$
  - $\rightsquigarrow \Delta_{Q_C} = 1$
- Somme $Q_S(D) = \left(\sum_{x \in O^1}(x_1, \ldots, x_d), \ldots, \sum_{x \in O^k}(x_1, \ldots, x_d)\right)$
  - $D' = D \cup \{x\}$: un seul O^j parmi k est augmenté de x
  - Seule une des sommes parmi k est impactée
  - Hypothèse : les d domaines egaux à $[0, r]$
  - $\rightsquigarrow \Delta_{Q_S} = d.r$

Budget ϵ pour chaque itération parmi t

- budget global : ϵ
- t itérations : composition séquentielle $\rightsquigarrow \frac{\epsilon}{t}$ par itération
  - budget pour le comptage : $\frac{\epsilon}{2t}$
  - budget pour la somme : $\frac{\epsilon}{2t}$

Algorithme simplifié ϵ-DP

Points clefs

1. choix **inchangé** des centroïds initiaux $o^1, \ldots, o^k$
2. répétition jusqu'à convergence, au plus t fois
   - 2.1 association **inchangée** de chaque x au groupe O^j de centroïd o^j le plus proche :

$$O^j = \left\{ x : \left\| x - o^j \right\| \leq \left\| x - o^{j'} \right\| \ \forall\, j' = 1, \ldots, k, j' \neq j \right\}$$

   - 2.2 m.-à-j. **bruitée** du centroïd o^j de chaque groupe :
     - 2.2.1 cardinalité : $|O^j|' = |O^j| + Lap(0, \frac{2t}{\epsilon})$
     - 2.2.2 somme : $\Sigma' = \sum_{x \in O^j} x + Lap(0, \frac{2d.r.t}{\epsilon})$
     - 2.2.3 $\rightsquigarrow$ nouveau centroïd : $o'^j = \frac{1}{|O^j|'} \Sigma'$
3. Publication : centroids $\{o'^1, \ldots, o'^k\}$ et cardinalités $|O^1|', \ldots, |O^k|'$

K-means avec DiffPrivLib

```
import pandas as pd
import diffprivlib.models as dp

df = pd.read_csv("iris.csv")
X = df.drop("Class", axis=1)
y = df[["Class"]]

dp_km = dp.KMeans(n_clusters=3,  epsilon=1.0)
dp_km.fit(X)
print(dp_km.cluster_centers_)
print(dp_km.labels_)
```