

Contributions aux communications multi-vues pour l'apprentissage collaboratif

Denis Maurel

10 Décembre 2018

Introduction

L'apprentissage machine (ou Machine Learning en anglais) est le domaine de recherche regroupant les algorithmes et les méthodes permettant **d'apprendre automatiquement** un résultat à partir d'un ensemble de données, aussi appelés **individus**. Les trois principaux sous-domaines du Machine Learning sont:

À noter que les domaines ne sont pas exclusifs (apprentissage semi-supervisé, apprentissage par renforcement profond...)

L'apprentissage machine (ou Machine Learning en anglais) est le domaine de recherche regroupant les algorithmes et les méthodes permettant **d'apprendre automatiquement** un résultat à partir d'un ensemble de données, aussi appelés **individus**. Les trois principaux sous-domaines du Machine Learning sont:

- **La classification:** apprentissage des correspondances entre une donnée et son label.

À noter que les domaines ne sont pas exclusifs (apprentissage semi-supervisé, apprentissage par renforcement profond...)

L'apprentissage machine (ou Machine Learning en anglais) est le domaine de recherche regroupant les algorithmes et les méthodes permettant **d'apprendre automatiquement** un résultat à partir d'un ensemble de données, aussi appelés **individus**. Les trois principaux sous-domaines du Machine Learning sont:

- **La classification:** apprentissage des correspondances entre une donnée et son label.
- **Le clustering:** détection de groupes d'individus similaires.

À noter que les domaines ne sont pas exclusifs (apprentissage semi-supervisé, apprentissage par renforcement profond...)

L'apprentissage machine (ou Machine Learning en anglais) est le domaine de recherche regroupant les algorithmes et les méthodes permettant **d'apprendre automatiquement** un résultat à partir d'un ensemble de données, aussi appelés **individus**. Les trois principaux sous-domaines du Machine Learning sont:

- **La classification:** apprentissage des correspondances entre une donnée et son label.
- **Le clustering:** détection de groupes d'individus similaires.
- **L'apprentissage par renforcement:** apprentissage d'un comportement permettant à un modèle de réagir à un environnement dynamique.

À noter que les domaines ne sont pas exclusifs (apprentissage semi-supervisé, apprentissage par renforcement profond...)

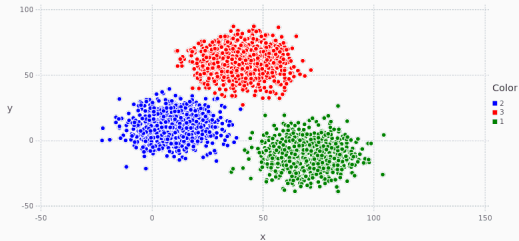
L'apprentissage machine (ou Machine Learning en anglais) est le domaine de recherche regroupant les algorithmes et les méthodes permettant **d'apprendre automatiquement** un résultat à partir d'un ensemble de données, aussi appelés **individus**. Les trois principaux sous-domaines du Machine Learning sont:

- **La classification**: apprentissage des correspondances entre une donnée et son label.
- **Le clustering**: détection de groupes d'individus similaires.
- **L'apprentissage par renforcement**: apprentissage d'un comportement permettant à un modèle de réagir à un environnement dynamique.

À noter que les domaines ne sont pas exclusifs (apprentissage semi-supervisé, apprentissage par renforcement profond...)

Clustering

- Tâche d'apprentissage **non supervisée** consistant à rassembler des groupes d'individus (a.k.a. clusters) de sorte à **maximiser la similarité intra-groupe** et à **minimiser la similarité inter-groupes**.



Clustering: similarité entre individus

- La notion de **similarité** est souvent confondue avec la notion de **distance**.
- La similarité doit être **adaptée à la nature des données**.

Euclidienne	$\ a - b\ _2 = \sqrt{\sum_i (a_i - b_i)^2}$
Manhattan	$\ a - b\ _1 = \sum_i a_i - b_i $
Maximum	$\ a - b\ _\infty = \max_i a_i - b_i $
Mahalanobis	$\sqrt{(a - b)^T S^{-1} (a - b)}$
Hamming	$\text{Hamming}(a, b) = \sum_i (1 - \delta_{a_i, b_i})$

Table 1: Exemples de distances

Clustering: types de partitions

Après un clustering, on obtient une partition de l'espace ainsi que les appartenances des individus à chaque groupe de cette partition. Ces appartenances peuvent être **dures**, **molles** ou **floues**.

	c ₁	c ₂	c ₃
x ₁	1	0	0
x ₂	0	1	0
x ₃	0	0	1
x ₄	0	0	1

(a) Clustering dur

	c ₁	c ₂	c ₃
x ₁	1	1	0
x ₂	0	1	1
x ₃	0	0	1
x ₄	0	0	1

(b) Clustering mou

	c ₁	c ₂	c ₃
x ₁	0.9	0.1	0
x ₂	0	0.8	0.2
x ₃	0	0.3	0.7
x ₄	0	0	1.0

(c) Clustering flou

Table 2: Les trois principaux types d'appartenances à des clusters

On peut regrouper les algorithmes de clustering en sous-catégories suivant l'approche qu'ils utilisent:

- **Méthodes hiérarchiques:** création d'un arbre de correspondance entre les individus (Agglomerative method).
- **Méthodes de quantification de vecteurs:** définition d'individus prototypes pour synthétiser les individus en entrée (K-Means).
- **Méthodes de densité:** estimation des clusters suivant les zones les plus densément peuplées de l'espace d'entrée (DBSCAN).
- **Méthodes stochastiques:** création de modèles probabilistes définissant la probabilité d'appartenance d'un individu à un cluster donné (GMM).

Apparition d'un nouveau contexte: **un même ensemble d'individu** est décrit dans plusieurs base de données indépendantes appelées **vues**.

Problème: comment obtenir un clustering de cet ensemble d'individus ?

Deux approches:

Apparition d'un nouveau contexte: **un même ensemble d'individu** est décrit dans plusieurs base de données indépendantes appelées **vues**.

Problème: comment obtenir un clustering de cet ensemble d'individus ?

Deux approches:

- **Coopérative:** chaque vue effectue un clustering de ses données locales avant de transférer ses résultats à une entité tiers qui devra fusionner les résultats.

Apparition d'un nouveau contexte: **un même ensemble d'individu** est décrit dans plusieurs base de données indépendantes appelées **vues**.

Problème: comment obtenir un clustering de cet ensemble d'individus ?

Deux approches:

- **Coopérative:** chaque vue effectue un clustering de ses données locales avant de transférer ses résultats à une entité tiers qui devra fusionner les résultats.
- **Collaborative:** chaque vue effectue un premier clustering local, puis le modifie en fonction des résultats obtenus par les autres vues.

Apparition d'un nouveau contexte: **un même ensemble d'individu** est décrit dans plusieurs base de données indépendantes appelées **vues**.

Problème: comment obtenir un clustering de cet ensemble d'individus ?

Deux approches:

- **Coopérative:** chaque vue effectue un clustering de ses données locales avant de transférer ses résultats à une entité tiers qui devra fusionner les résultats.
- **Collaborative:** chaque vue effectue un premier clustering local, puis le modifie en fonction des résultats obtenus par les autres vues.

Exemple multi-vues

- **Vue 1:** Ensemble des achats récent d'un individus sur des sites d'e-commerce
- **Vue 2:** Salaire et régime alimentaire
- **Vue 3:** Contenu des derniers repas de chaque individu

Clustering collaboratif: définition

Le **clustering collaboratif** est un domaine récent désignant l'ensemble des méthodes permettant à **plusieurs algorithmes de clustering** opérant sur des **sources de données différentes** de collaborer pour **améliorer localement** leurs résultats.

- Les algorithmes utilisés peuvent être **différents**.
- Les vues doivent partager soit leurs descripteurs (clustering vertical), soit **leurs individus** (clustering horizontal) pour pouvoir être comparées.

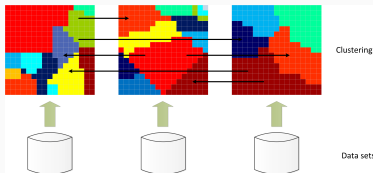


Figure 1: Illustration du principe de clustering collaboratif horizontal

Clustering collaboratif: processus

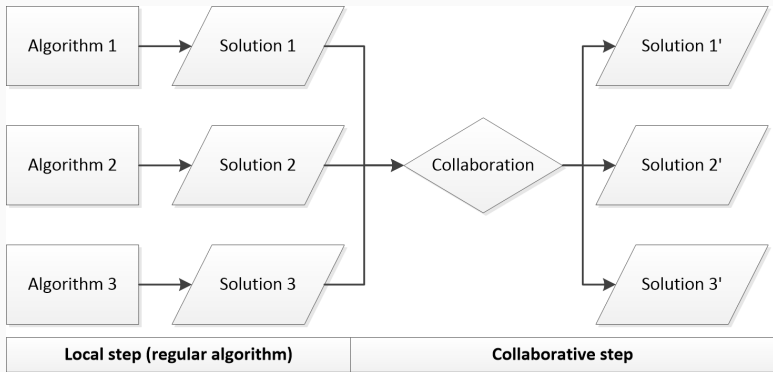


Figure 2: Processus de clustering collaboratif

Définition d'un algorithme de clustering collaboratif:

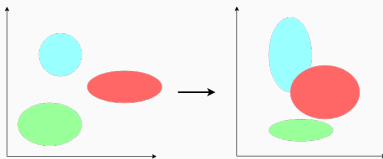
- Q_{local}^i est généralement basé sur **le critère de l'algorithme local** à optimiser.
- Q_{collab}^i se base sur **l'échange d'information entre vues**, typiquement les appartenances des individus aux clusters respectifs de chaque vue.
- α_i et β_j^i sont définis **à la main**. L'approximation $\forall j, \beta_j^i = \alpha_i^2$ est parfois utilisée car donnant de bons résultats en pratique.

Clustering collaboratif incrémental

Objectif du clustering collaboratif: définir un ensemble de clusters suivant **la distribution** des données fournies en entrée.

Problème: il arrive que cette distribution **évolue au cours du temps**

Exemple: évolution du régime alimentaire d'un individu ou de la répartition des salaires à l'échelle d'une population.



Utilisation du clustering **incrémental**: les clusters sont appris au cours du temps afin de **s'adapter** aux éventuels changements de distribution. On utilise les N_{batch} derniers individus comme échantillon d'apprentissage.

Définition d'une méthode de clustering incrémental:

Définition d'une méthode de clustering incrémental:

- Choix de la méthode de clustering

Définition d'une méthode de clustering incrémental:

- Choix de la méthode de clustering
- Adaptation de la méthode de clustering pour de l'apprentissage incrémental

Définition d'une méthode de clustering incrémental:

- Choix de la méthode de clustering
- Adaptation de la méthode de clustering pour de l'apprentissage incrémental
- Adaptation du clustering collaboratif au modèle de clustering obtenu

Choix de la méthode de clustering

Dans notre cas, utilisation des **cartes auto-adaptatrices de Kohonen** (ou Self-Organizing Maps (SOM) en anglais) comme méthode de clustering.

- **1^{ère} contrainte:** des individus correspondants doivent appartenir à des **prototypes correspondants** ou à leurs voisinages proches.
- **2^{ème} contrainte:** **même topologie des cartes** pour toutes les vues pour les rendre comparables.

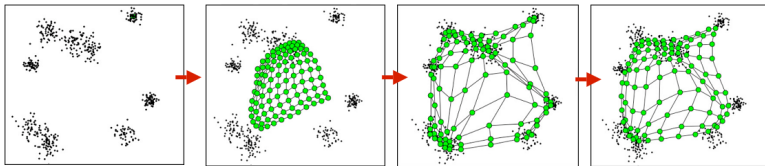
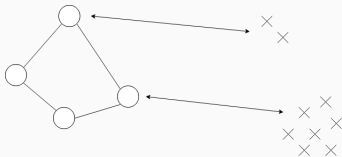


Figure 3: Exemple de SOM

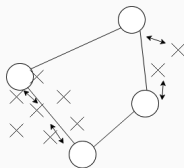
Cartes Auto-Adaptatrices (SOM)

- Méthode à base de **prototypes** (quantification de vecteurs)
- Permet la **visualisation** de données en hautes dimensions
- Notion de **voisinage**: utilisation d'une **fonction de température**.

$$\lambda(t) = \lambda_{\min} \left(\frac{\lambda_{\max}}{\lambda_{\min}} \right)^{\frac{1}{t}} \quad K_{i,j} = \exp \left(-\frac{d_1^2(i,j)}{\lambda(t)} \right)$$



(a) Température élevée



(b) Température faible

Les SOM incrémentales ont déjà été étudiées, mais les solutions proposées se basent toutes sur **l'ajout de prototypes** (Papliński (2012) et Deng and Kasabov (2000)).

→ **Non applicable au clustering collaboratif** du fait de la seconde contrainte: la topologie doit rester la même pour toutes les cartes.

Limitation: fonction de température dépendante du **temps**

Solution: rendre la fonction dépendante des **individus**

$$\lambda(t) = \lambda_{\min} \left(\frac{\lambda_{\max}}{\lambda_{\min}} \right)^{\frac{1}{t}} \rightarrow \tilde{\lambda}(B, W) = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \|x_i - \chi(x_i)\|_2$$

L'application des SOM au clustering collaboratif se fait en définissant les termes précédemment définis:

$$Q_{\text{local}}^m = \alpha_m \sum_{i=1}^N \sum_{j=1}^{|W|} K_{j,\chi(x_i)}^m \|x_i^m - \omega_j^m\|^2$$

$$Q_{\text{collab}}^m = \sum_{m'=1, m' \neq m}^P \beta_m^{m'} \sum_{i=1}^N \sum_{j=1}^{|W|} (K_{j,\chi(x_i)}^m - K_{j,\chi(x_i)}^{m'})^2 \|x_i^m - \omega_j^m\|^2$$

- $W \rightarrow$ la carte de prototypes
- $\chi(x_i) \rightarrow$ la fonction retournant le prototype le plus proche de x_i .
- $x_i^k \rightarrow$ l'individu i dans la vue m
- $\omega_j^m \rightarrow$ le prototype j de la SOM de la vue m .

Adaptation de notre version de SOM incrémentale au clustering collaboratif:

$$\lambda \rightarrow \tilde{\lambda}$$

$$K_{i,j}(\lambda) \rightarrow K_{i,j}(\tilde{\lambda}) \rightarrow \widetilde{K}_{i,j}$$

$$Q_{\text{local}}^m / Q_{\text{collab}}^m(K_{i,j}) \rightarrow Q_{\text{local}}^m / Q_{\text{collab}}^m(\widetilde{K}_{i,j}) \rightarrow \widetilde{Q}_{\text{local}}^m / \widetilde{Q}_{\text{collab}}^m$$

Le nouveau critère dépendant uniquement des N_{batch} derniers individus apparus, il est possible d'effectuer un **apprentissage collaboratif incrémental** sur l'ensemble des vues.

Les règles de mise à jour sont obtenus par **descente de gradient** appliquée sur ce critère.

- Analyse de l'impact du clustering collaboratif sur l'apprentissage incrémental.
- Application directe de la phase collaborative pour le clustering collaboratif pour pouvoir comparer les résultats aux version locales.
- Variation de la taille du batch pour étudier l'impact sur l'apprentissage.

Test de notre méthode sur 4 jeux de données différents

- Spambase
- Waveform
- Wisconsin Breast Cancer Diagnosis (WDBC)
- Isolet

Pureté d'une SOM: pureté moyenne de ses prototypes. **Pureté d'un prototype:** classe majoritaire parmi les individus associés à ce prototype

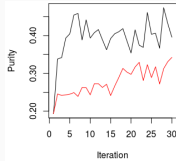
Erreur moyenne de quantification: distance moyenne entre un prototype et les individus qui y sont associés.

$$q_e = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \|x_i - \omega_{\chi(x_i)}\|^2$$

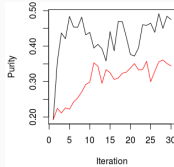
Table 3: Erreur de quantification moyenne pour chaque base de donnée. Les nombres en gras sont les plus petits pour chaque ligne

	Vue	SOM Incrémentales	Clustering Collaboratif Incrémentale
Spam Base	1	0.31	0.26
	2	0.18	0.19
	3	0.18	0.16
Waveform	1	0.18	0.23
	2	0.17	0.19
	3	0.24	0.30
WDBC	1	0.19	0.19
	2	0.16	0.19
	3	0.20	0.16
Isolet	1	2.15	1.27
	2	2.84	1.38
	3	2.85	1.37

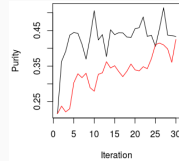
Expérimentations: résultats sur Isolet



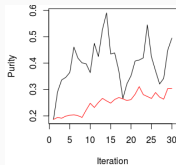
(a) $N_{\text{batch}} = 10$, View 1



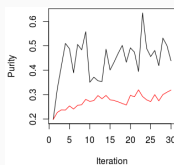
(b) $N_{\text{batch}} = 10$, View 2



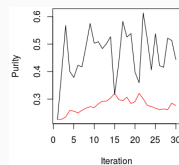
(c) $N_{\text{batch}} = 10$, View 3



(d) $N_{\text{batch}} = 3$, View 1



(e) $N_{\text{batch}} = 3$, View 2



(f) $N_{\text{batch}} = 3$, View 3

Figure 5: Évolution des puretés pour Isolet. Les lignes rouges représentent les SOM incrémentales tandis que les lignes noires représentent les SOM incrémentales collaboratives. Chaque itération correspond à l'arrivée d'une nouvelle donnée

- **Score en quantification à long termes comparables:** le clustering collaboratif n'endommage pas les résultats locaux.
- Le clustering collaboratif permet de **limiter l'impact du bruit** dans les vues.
- Les SOM incrémentales collaboratives **apprennent plus vite que les SOM incrémentales seules**: meilleure exploitation de l'information via le partage.
- Les SOM collaboratives sont globalement **plus instables** que les SOM incrémentales.
- La stabilité de l'apprentissage **augmente avec la taille du batch**: plus d'informations à exploiter.

Optimisation de paramètres pour le clustering collaboratif

Objectif: Apprendre automatiquement les α et les β en s'affranchissant de simplifications telles que $\beta = \alpha^2$.

Rappel du critère du clustering collaboratif:

$$\begin{aligned} Q^i &= \alpha_i Q_{\text{local}}^i(v_i) + Q_{\text{collab}}^i(v_i, v_{j \neq i}) \\ &= \alpha_i Q_{\text{local}}^i(v_i) + \sum_{j \neq i} \beta_j C_j^i(v_i, v_j) \end{aligned}$$

Nous avons proposé une nouvelle méthode de pondération permettant d'apprendre automatiquement les poids fixant les importances relatives des différents scores.

La définition des α et des β passe par la résolution d'un problème sous contrainte. Les contraintes sont les suivantes :

La définition des α et des β passe par la résolution d'un problème sous contrainte. Les contraintes sont les suivantes :

$$\bullet \beta^* = \operatorname{argmin}_{\beta} \sum_{i \neq j} \beta_{i,j} c_{i,j}$$

Optimisation de paramètres: définition du problème

La définition des α et des β passe par la résolution d'un problème sous contrainte. Les contraintes sont les suivantes :

- $\beta^* = \operatorname{argmin}_{\beta} \sum_{i \neq j} \beta_{i,j} c_{i,j}$
- $\forall j \quad \prod_{i \neq j} \beta_{i,j} = 1$

La définition des α et des β passe par la résolution d'un problème sous contrainte. Les contraintes sont les suivantes :

- $\beta^* = \operatorname{argmin}_{\beta} \sum_{i \neq j} \beta_{i,j} c_{i,j}$
- $\forall j \quad \prod_{i \neq j} \beta_{i,j} = 1$
- $\forall (i, j) \quad \beta_{i,j} > 0$

La définition des α et des β passe par la résolution d'un problème sous contrainte. Les contraintes sont les suivantes :

- $\beta^* = \operatorname{argmin}_{\beta} \sum_{i \neq j} \beta_{i,j} c_{i,j}$
- $\forall j \quad \prod_{i \neq j} \beta_{i,j} = 1$
- $\forall (i, j) \quad \beta_{i,j} > 0$
- Affranchissement des α : comme il s'agit de pondérations relatives, on peut fixer $\alpha = 1$.

La définition des α et des β passe par la résolution d'un problème sous contrainte. Les contraintes sont les suivantes :

- $\beta^* = \operatorname{argmin}_{\beta} \sum_{i \neq j} \beta_{i,j} c_{i,j}$
- $\forall j \quad \prod_{i \neq j} \beta_{i,j} = 1$
- $\forall (i, j) \quad \beta_{i,j} > 0$
- Affranchissement des α : comme il s'agit de pondérations relatives, on peut fixer $\alpha = 1$.

Objectif: mettre des poids plus élevés sur les meilleurs accords, tout en gardant une certaine partie des vues en désaccord pour faire évoluer l'information

Critère + contrainte d'égalité + contrainte d'inégalité

→ **méthode de Karush-Kuhn-Tucker (KKT)**

$$\mathcal{L}(\beta, \nu, \lambda) = \sum_{j=1}^J \sum_{i \neq j}^J (\beta_{i,j} c_{i,j} - \nu_j \ln \beta_{i,j} - \lambda_{i,j} \beta_{i,j}).$$

$$\frac{\partial \mathcal{L}}{\partial \beta_{i,j}} = 0, \quad \frac{\partial \mathcal{L}}{\partial \nu} = 0, \quad \frac{\partial \mathcal{L}}{\partial \lambda} = 0$$

Après résolution, on obtient:

$$\forall (i, j), \quad i \neq j \quad \beta_{i,j} = \frac{(\prod_{k \neq j} c_{k,j})^{\frac{1}{j-1}}}{c_{i,j}}$$

L'importance d'une vue externe relativement à une vue locale est proportionnelle au rapport entre la moyenne géométrique de toutes les dissimilarités par rapport à la dissimilarité entre les deux vues .

Après résolution, on obtient:

$$\forall (i, j), \quad i \neq j \quad \beta_{i,j} = \frac{(\prod_{k \neq j} c_{k,j})^{\frac{1}{j-1}}}{c_{i,j}}$$

L'importance d'une vue externe relativement à une vue locale est proportionnelle au rapport entre la moyenne géométrique de toutes les dissimilarités par rapport à la dissimilarité entre les deux vues .

→ **plus 2 vues sont similaires, plus elles collaboreront.**

Algorithm 1: Algorithme topologique de collaboration horizontale

Initialisation: Initialiser toutes les cartes de prototypes W aléatoirement.

Étape locale: Initialisation des cartes

forall Vue **do**

| Minimiser la fonction objectif des cartes auto-adaptatrices standards.

end

Étape collaborative:

forall Vue **do**

| Pour w fixé, mettre à jour β

$$\beta^* = \operatorname{argmin}_{\beta} Q(w, \alpha, \beta)$$

| Pour β fixés, mettre à jour les prototypes de toutes les cartes:

$$w^* = \operatorname{argmin}_w Q(w, \alpha, \beta)$$

end

Deux axes d'analyse:

- Analyse de **l'évolution du critère collaboratif** avec et sans apprentissage automatique des poids (critère: différence relative).
- Analyse des **valeurs prises par les β** en fin d'apprentissage. (critère: différences relatives entre les poids).

Deux axes d'analyse:

- Analyse de **l'évolution du critère collaboratif** avec et sans apprentissage automatique des poids (critère: différence relative).
- Analyse des **valeurs prises par les β** en fin d'apprentissage. (critère: différences relatives entre les poids).

Ce que l'on s'attend à trouver:

- Une **amélioration de la valeur du critère collaboratif** grâce à notre méthode
- **L'identification automatique des vues bruitées**, menant à des valeurs de β proches de 0.

Deux axes d'analyse:

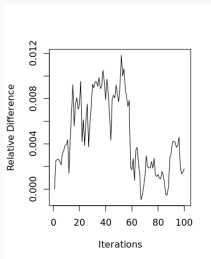
- Analyse de **l'évolution du critère collaboratif** avec et sans apprentissage automatique des poids (critère: différence relative).
- Analyse des **valeurs prises par les β** en fin d'apprentissage. (critère: différences relatives entre les poids).

Ce que l'on s'attend à trouver:

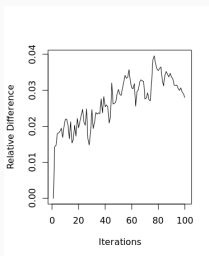
- Une **amélioration de la valeur du critère collaboratif** grâce à notre méthode
- **L'identification automatique des vues bruitées**, menant à des valeurs de β proches de 0.

À noter: pour chaque base de données, **une vue entièrement composée de bruit** a été rajoutée pour tester le second point.

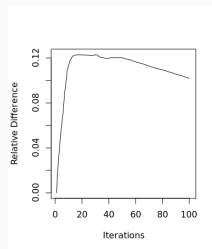
Figure 6: Différences relatives du critère collaboratif avec et sans optimisation des β tout au long du processus d'apprentissage.



(a) Spambase



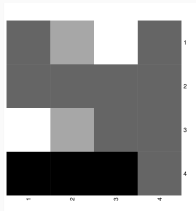
(b) Isolet



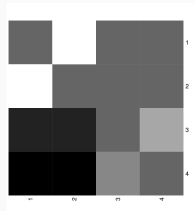
(c) VHR Strasbourg

- **Amélioration du critère** de manière significative dans la majorité des cas
- Tous les jeux de données **ne sont pas traités de la même manière** (dépend de la quantité d'informations à partager).

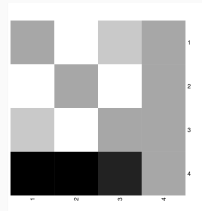
Figure 8: Cartes de chaleur des matrices de β pour chaque jeu de données. $M(i, j)$ correspond à l'importance accordée à la Vue j par la Vue i .
Blanc = forte collaboration - noir = faible collaboration - diagonale représente $\beta = 1$.



(a) WDBC



(b) Waveform



(c) Isolet

- **Identification des vues bruitées.**
- Apparition de **méta-clusters** (des clusters de clusters). Les vues auront tendance à se regrouper en sous-groupe mutuellement d'accord.

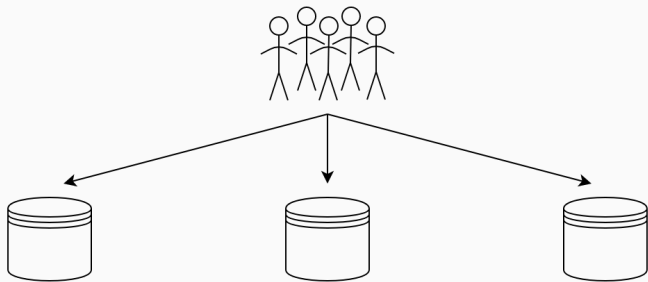
- Proposition d'un **système de pondération automatique des vues externes** pour le clustering collaboratif.
- Amélioration de la valeur du critère collaboratif
- Identification des vues bruitées
- (BONUS) Apparition de méta-clusters de vues

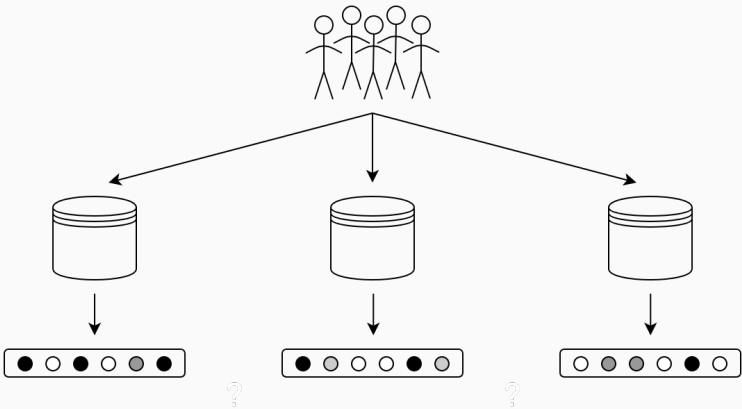
- Proposition d'un **système de pondération automatique des vues externes** pour le clustering collaboratif.
- Amélioration de la valeur du critère collaboratif
- Identification des vues bruitées
- (BONUS) Apparition de méta-clusters de vues

**Est-il possible d'étendre l'idée de collaboration
à d'autres problèmes que le clustering ?**

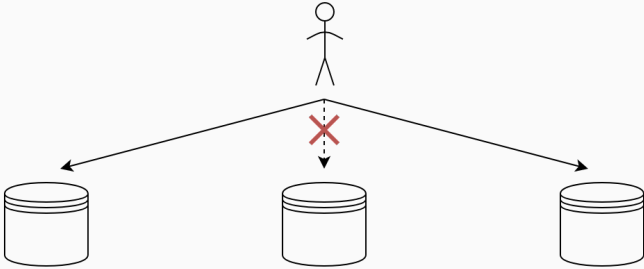
Système de reconstruction collaboratif

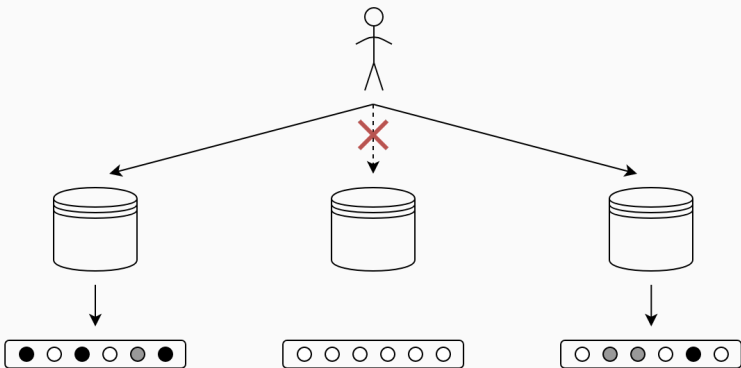


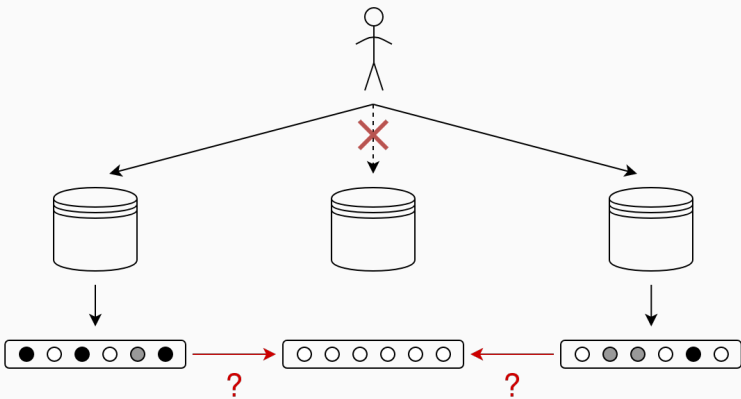












Reconstruction collaborative: définition

Comment définir le problème de **reconstruction collaborative** ?

- Application du **paradigme collaboratif** à un autre problème
- Le clustering collaboratif nécessite la **description d'un même individu dans chacune des vues**
- En pratique, les données souvent sont **soit manquantes, soit incomplètes**.

?			?		?					
		?			?		?		?	
	?	?	?	?			?			?
?					?				?	?

La reconstruction collaborative devrait permettre d'obtenir **une approximation** d'un individu dans une vue connaissant **l'information présente dans les autres vues**.

1. **Compresser et anonymiser** les données
2. **Transférer de l'information** d'une vue externe à une vue locale
3. **Combiner les informations** en provenance de différentes sources

1. **Compresser et anonymiser** les données
2. **Transférer de l'information** d'une vue externe à une vue locale
3. **Combiner les informations** en provenance de différentes sources

Ce que nous proposons:

1. **Des réseaux de neurones** (autoencodeurs).

1. **Compresser et anonymiser** les données
2. **Transférer de l'information** d'une vue externe à une vue locale
3. **Combiner les informations** en provenance de différentes sources

Ce que nous proposons:

1. **Des réseaux de neurones** (autoencodeurs).
2. **D'autres réseaux de neurones** (perceptrons multi-couches).

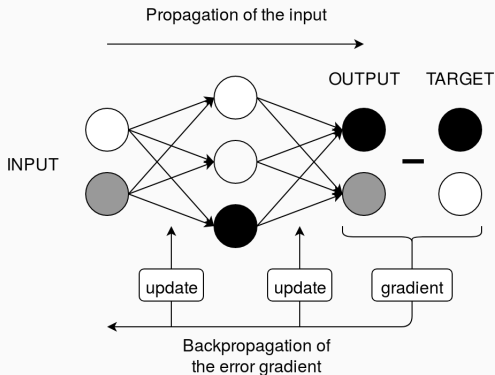
1. **Compresser et anonymiser** les données
2. **Transférer de l'information** d'une vue externe à une vue locale
3. **Combiner les informations** en provenance de différentes sources

Ce que nous proposons:

1. **Des réseaux de neurones** (autoencodeurs).
2. **D'autres réseaux de neurones** (perceptrons multi-couches).
3. **Une nouvelle méthode de combinaison.**

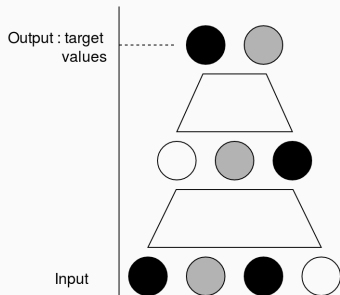
Réseaux de neurones: backpropagation

- Méthode utilisée pour **mettre à jour des paramètres** en utilisant la méthode de **descente de gradient**.
- L'erreur est propagée **de la sortie vers l'entrée** du réseau.



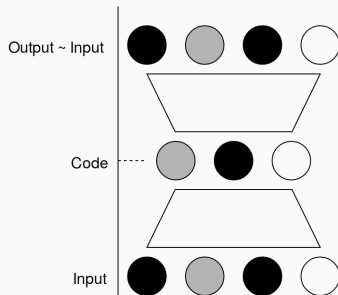
Perceptron multi-couches

- entrées et sorties **différentes**
- apprentissage **supervisé**



Autoencodeur

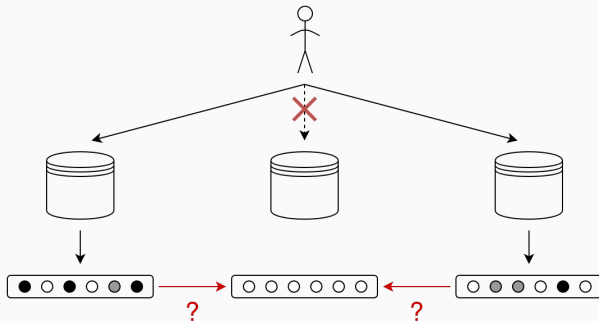
- entrées et sorties **identiques**
item apprentissage **non**
supervisé



Reconstruction collaborative: rappel

Rappel de la problématique

- Un individu décrit dans **toutes les vues sauf une**
- Utilisation des informations externes pour obtenir **une approximation locale**



Reconstruction collaborative: système

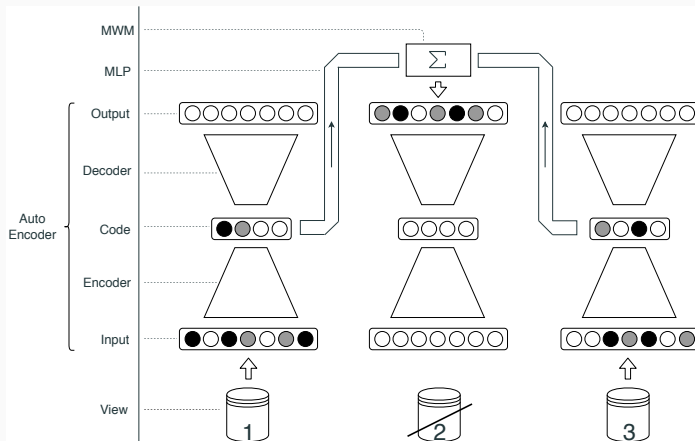


Figure 10: Architecture d'un système de reconstruction collaborative

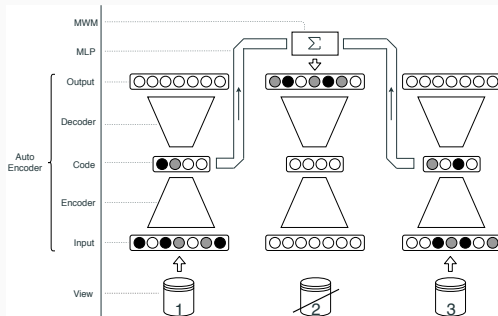
Reconstruction collaborative: système

Autoencodeurs

- Un par vue: encoder chaque espace
- Rend difficile la reconstruction des données originales sans le décodeur

Perceptron multi-couches

- Décode les représentations externes
- Passe d'un espace codé à l'espace local



- Pour chaque individu et avec N vues, on peut avoir jusqu'à $N - 1$ **reconstructions différentes**.
- Comment les combiner ?

- Pour chaque individu et avec N vues, on peut avoir jusqu'à $N - 1$ **reconstructions différentes**.
- Comment les combiner ?
- Habituellement, combinaison **pondérée** des différentes reconstructions.
 - Un scalaire par vue
 - Différentes façons d'apprendre les poids

- Pour chaque individu et avec N vues, on peut avoir jusqu'à $N - 1$ **reconstructions différentes**.
- Comment les combiner ?
- Habituellement, combinaison **pondérée** des différentes reconstructions.
 - Un scalaire par vue
 - Différentes façons d'apprendre les poids
- MAIS cela implique une hypothèse forte: chaque vue contient exactement **la même information** sur les individus → irréaliste.
- À la place d'un poids unique, nous utilisons un **masque**.

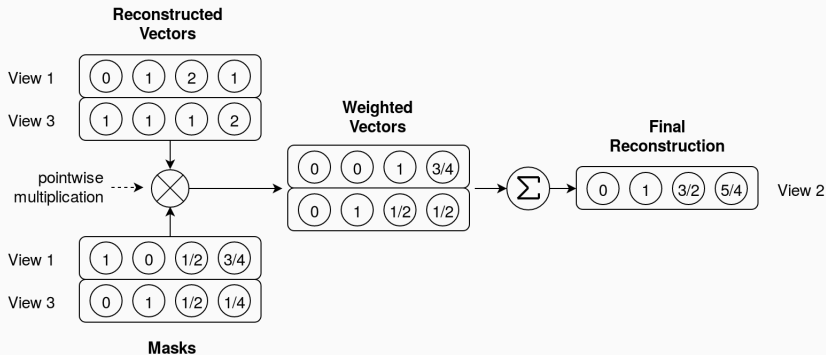


Figure 11: Méthode de pondération par masque (Masked Weighting Method (MWM) en anglais).

Perspectives

Conclusion

References

- D. Deng and N. Kasabov. Esom: An algorithm to evolve self-organizing maps from online data streams. In *Neural Networks*, volume 6, pages 3–8. IEEE, 2000.
- A. P. Papliński. Incremental self-organizing map (isom) in categorization of visual objects. In *ICONIP*, pages 125–132. Springer, 2012.