
Indexation multimédia par dictionnaires visuels en environnement décentralisé

Une approche par protocoles Gossip

Jérôme Fellus¹, David Picard¹, Philippe-Henri Gosselin²

1. *ETIS - UMR CNRS 8051 - ENSEA - Université de Cergy-Pontoise
France*

jerome.fellus@ensea.fr,picard@ensea.fr

2. *INRIA, Texmex project, Campus de Beaulieu, Rennes
France*

philippe.gosselin@inria.fr

RÉSUMÉ. Pour permettre la recherche par le contenu de documents multimédias repartis sur de larges réseaux, nous proposons un système d'indexation basé sur l'apprentissage décentralisé et asynchrone de dictionnaires visuels. Nous proposons un algorithme décentralisé pour le calcul des dictionnaires basé sur un protocole d'agrégation Gossip, qui produit un dictionnaire local performant en chaque nœud du réseau. Nous fournissons une loi empirique pour déterminer les paramètres optimaux du système selon la taille du réseau ciblé, qui permettent d'obtenir des dictionnaires égaux entre nœuds pour un coût de communication faible. Une étude expérimentale met en évidence les capacités de passage à l'échelle et la qualité de recherche du système.

ABSTRACT. In order to allow content-based retrieval of multimedia documents spread over large networks, we propose an indexing system based on decentralized and asynchronous learning of visual codebooks. We propose a Gossip-based decentralized algorithm to compute an accurate visual codebook at each networking node. We provide an empirical law to define the optimal parameters, given the targetted network size, to get codebooks which are equal between nodes with low communication costs. An experimental study highlights the scaling abilities and the retrieval accuracy of our system.

MOTS-CLÉS : recherche par le contenu distribuée (D-CBR), indexation multimédia, dictionnaires visuels, clustering décentralisé, protocoles Gossip.

KEYWORDS: distributed content-based retrieval (D-CBR), multimedia indexing, visual codebooks, decentralized clustering, Gossip aggregation protocols.

DOI:10.3166/TS.32.39-64 © 2015 Lavoisier

Extended Abstract

This paper tackles content-based multimedia retrieval (CBR) in decentralized networking environments. State-of-the-art CBR approaches rely on a preliminary indexing step. They first extract from each multimedia document a set of low level feature vectors and fit a statistical model to their distribution. For each document, its own features are then projected onto this model to get a single vector called *signature*. Using simple vector metrics on signatures instead of raw content, the resulting similarity measure is both accurate and computationally efficient. This enables large-scale retrieval, classification, similarity search, etc.

In this paper, the statistical model of interest is a hard clustering of the feature space, called *visual codebook*. While the K -means algorithm is a well studied and easy-to-implement technique to train such visual codebooks, it requires all data to be available on a single machine. Yet, available amounts of multimedia data are indubitably growing faster than the computing and storage capacities of individual machines. This motivated recent developments in distributed content-based retrieval (D-CBR), that aims at porting centralized paradigms and algorithms involved in CBR to a set of connected processing units, each one holding a subset of input data.

While there exists many distributed equivalents of K -means straightforwardly leading to D-CBR systems, an important challenge is to design a fully decentralized and asynchronous D-CBR system targeting large and dynamic consumer networks (Internet, P2P overlays, ad-hoc sensor or mobile networks, ...).

In this paper, we present such a decentralized and asynchronous system, that uses a Sum-Weight Gossip averaging protocol to train the clustering model. Given a network of N nodes, our system allows each node to iteratively compute a codebook fitting the complete network data. The resulting codebooks are equivalent to first gathering the full data on a single machine and running a centralized K -means algorithm. Our approach only requires that each node iteratively sends small messages to a few random neighbors. The process is fully asynchronous, *i.e.*, nodes perform local computations and message emission without waiting for other nodes response at any time. A decentralized codeword shifting procedure is added to improve codebooks accuracy beyond a simple K -means. Importantly, all nodes converge to the same codebook, allowing each one to compute signatures for its own documents only using local computations.

Experimental evaluation of our codebook learning algorithm shows that all nodes find an accurate consensus codebook even when each node sends very few messages. In particular, for fully connected networks, the number of messages per node needed to reach a fixed error between nodes scales logarithmically with the network's size. This property demonstrates the suitability of our approach to very large networks. Experimental study of the whole D-CBR system confirms that obtained codebooks generate accurate indexes, as we achieved retrieval results equivalent to a centralized indexing scheme. Interestingly, retrieval precision is kept high even when the number of messages per node is insufficient to get a perfect consensus on local codebooks,

suggesting that rough modeling of the feature space can be sufficient to obtain good retrieval results in a decentralized context.

1. Introduction

Les performances des moteurs de recherche actuels permettent à des milliards d'utilisateurs d'accéder à des documents textuels au sein de collections extrêmement larges et variées. La recherche de contenus multimédias (images, vidéos, son, . . .), qui représentent une part grandissante des documents disponibles, est quant à elle un enjeu de premier plan. Essentiellement non textuels, ceux-ci requièrent des méthodes de recherche spécifiques faisant intervenir directement leur contenu visuel et/ou sonore.

Tout comme leurs homologues textuels, les systèmes de recherche par le contenu recourent à une phase préalable d'*indexation*. Les méthodes d'indexation actuelles nécessitant de modéliser l'ensemble des contenus disponibles, l'approche classique est de rapatrier tous les documents sur une unique machine très performante qui se charge de les indexer. Cette phase d'indexation est très gourmande en temps de calcul et en espace mémoire. Pour des collections à l'échelle du web, les capacités des machines les plus performantes sont dépassées. Aucune de ces méthodes ne permet ainsi d'indexer le flux continu de nouveaux contenus disponibles (*e.g.* en 2013, 100 heures de vidéo sont mises en ligne chaque minute sur Youtube).

En réalité, ces immenses masses de contenu proviennent d'une multitude de sources, plongées dans des réseaux grand public (Internet, réseaux pair-à-pair (P2P), réseaux mobiles. . .). De l'appareil mobile à l'imageur satellitaire en passant par le serveur de média à la demande, chaque source bénéficie souvent d'une disponibilité et de capacités de calcul et de stockage suffisantes pour indexer son propre contenu.

Afin de tirer parti de ces capacités et de cette organisation des données, de récents développements en recherche d'image par le contenu distribuée (D-CBIR) (Müller *et al.*, 2003; Marée *et al.*, 2010; Picard *et al.*, 2012) visent à répartir le processus d'indexation sur de multiples unités mises en réseau. Les meilleures méthodes d'indexation multimédia s'appuyant sur une modélisation globale des contenus, un des défis soulevés est de permettre le calcul de ce modèle de manière décentralisée et asynchrone lorsque les contenus sont repartis sur de larges réseaux dynamiques (Datta *et al.*, 2006; Fatta *et al.*, 2012).

Dans cet article, nous nous intéressons aux méthodes d'indexation par *dictionnaires visuels*, qui offrent à l'heure actuelle les meilleurs résultats. Nous proposons une stratégie d'indexation multimédia décentralisée qui offre une qualité et une rapidité de recherche au niveau de l'état de l'art, tout en tolérant des quantités de contenu inaccessibles aux approches centralisées. Celle-ci s'appuie sur le calcul décentralisé des dictionnaires visuels par des protocoles Gossip asynchrones.

Après un panorama succinct des méthodes existantes, nous présentons le système proposé puis une étude expérimentale qui met en évidence ses capacités de passage à

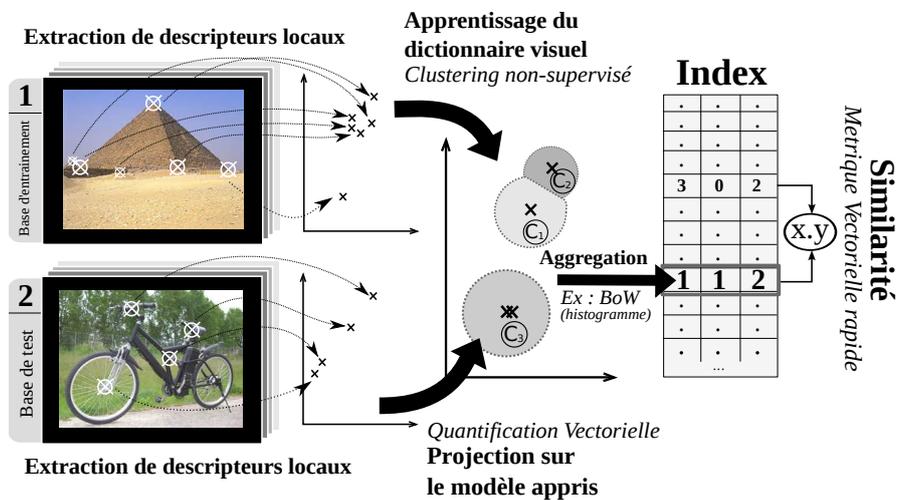


Figure 1. Indexation par dictionnaires visuels

l'échelle et ses performances vis à vis d'une méthode centralisée, avant de conclure sur les perspectives et les nouveaux enjeux soulevés.

2. État de l'art

Un système de recherche par le contenu s'appuie sur une *mesure de similarité* entre documents. Cette mesure de similarité correspond au choix d'un mode de représentation des documents et d'une métrique associée pour les comparer. Dans cette section, nous introduisons les mesures basées sur des dictionnaires visuels, puis nous présentons la méthode classique pour calculer ces dictionnaires visuels. Nous recensons ensuite quelques solutions distribuées existantes avant de nous concentrer sur l'approche retenue basée sur des protocoles Gossip.

2.1. Indexation par dictionnaires visuels

En indexation visuelle, les mesures de similarité les plus performantes sont basées sur la mise en correspondance de descripteurs locaux (SIFT (Lowe, 2004), SURF (Bay *et al.*, 2008), HoG, ...). On extrait de chaque document un ensemble de descripteurs appelé *sac de descripteurs*, via une détection de points d'intérêts ou sur une grille dense. Chaque sac peut contenir plusieurs milliers (images) à plusieurs millions (vidéos) de descripteurs. Pour manipuler de grandes bases, la solution la plus répandue est de transformer ces représentations volumineuses en simple vecteurs *signatures*, via une étape d'*indexation* qui peut être effectuée hors-ligne. La similarité entre deux signatures est alors très rapide à calculer.

Nous nous intéressons ici aux méthodes d’indexation par *dictionnaires visuels*, qui donnent actuellement les meilleurs résultats. Un dictionnaire visuel \mathcal{M} est composé d’un ensemble de K descripteurs prototypes $\{\mu_1, \dots, \mu_K\}$ appelés *mots visuels*. Il définit un *quantificateur* $q_{\mathcal{M}}$ qui associe à un descripteur \mathbf{b} l’indice du mot visuel le plus proche :

$$q_{\mathcal{M}}(\mathbf{b}) = \arg \min_k d(\mathbf{b}, \mu_k), \quad (1)$$

avec $d(\cdot, \cdot)$ une distance entre descripteurs (*e.g.* euclidienne). Indexer un sac de descripteurs consiste alors à le projeter sur le dictionnaire visuel pour obtenir une signature vectorielle (*cf.* figure 1).

La méthode de projection la plus utilisée, nommée *Bag-of-Words* (BoW) (Sivic, Zisserman, 2003), en analogie avec les méthodes d’indexation textuelles, consiste à calculer la fréquence d’apparition des mots visuels. Une signature BoW est ainsi un histogramme de dimension K . Des extensions à BoW ont été proposées, en particulier en prenant en compte la similarité entre descripteur et mot visuel (*Soft Coding* (Gemert *et al.*, 2010)), ou en envisageant la projection comme un problème de reconstruction des descripteurs (*Sparse Coding* (Yang *et al.*, 2009), *Locality-constrained Linear Coding* (Wang *et al.*, 2010)). Afin de réduire la taille du dictionnaire tout en augmentant les performances, des méthodes de projection mesurant la déviation entre les descripteurs et le dictionnaire visuel (VLAD (Jégou *et al.*, 2010)) ou ses statistiques d’ordre supérieur (Fisher Vectors (Perronnin, Dance, 2007), VLAT (Picard, Gosselin, 2011)) ont été proposées avec succès. Elle offrent aujourd’hui les meilleurs résultats sur les jeux de données académiques. Dans tous les cas, ces techniques reposent sur l’apprentissage d’un dictionnaire visuel, généralement obtenu par des algorithmes de clustering non supervisé tels que K-Means.

2.2. Calcul de dictionnaires visuels par l’algorithme K-means

La qualité des signatures obtenues est directement liée à la capacité du dictionnaire à modéliser correctement l’espace des descripteurs. Un dictionnaire visuel est donc généralement optimisé à l’aide d’un algorithme de type K-means (Linde *et al.*, 1980; Lloyd, 1982) sur un ensemble représentatif de descripteurs.

Étant donné un ensemble $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ de n échantillons de \mathbb{R}^D , l’algorithme K-means produit un dictionnaire $\mathcal{M} = \{\mu_1, \dots, \mu_K\}$ de K vecteurs de \mathbb{R}^D , nommés *centroïdes*. Son objectif est de minimiser l’erreur de quantification moyenne (MQE) de \mathcal{M} sur \mathbf{X} , définie comme suit :

$$\text{MQE}_{\mathbf{X}}(\mathcal{M}) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x} \in \mathbf{X}} d(\mathbf{x} - q_{\mathcal{M}}(\mathbf{x})) \quad (2)$$

Pour ce faire, K-means adopte une stratégie de descente de gradient qui améliore itérativement un dictionnaire $\mathcal{M}(0)$ initialisé aléatoirement. A chaque itération τ , $\mathcal{M}(\tau)$ est amélioré à partir de $\mathcal{M}(\tau - 1)$ grâce à deux étapes d’optimisation :

Étape 1 : assignation des cellules. Étant donné $\mathcal{M}(\tau)$, affecter à chaque échantillon \mathbf{x} de \mathbf{X} l'indice de son plus proche voisin dans $\mathcal{M}(\tau)$. On obtient alors K cellules $\mathcal{C}_1(\tau), \dots, \mathcal{C}_K(\tau)$ définies comme la partition de Voronoi de \mathbf{X} engendrée par $\mathcal{M}(\tau)$:

$$\forall k, \quad \mathcal{C}_k(\tau) = \{\mathbf{x} \in \mathbf{X} / \arg \min_{\mu_k \in \mathcal{M}(\tau)} d(\mathbf{x}, \mu_k)\} \quad (3)$$

Étape 2 : mise à jour du dictionnaire. Étant donnée la partition $\{\mathcal{C}_k(\tau)\}_k$, calculer le nouveau dictionnaire $\mathcal{M}(\tau + 1)$ comme l'ensemble des barycentres des cellules $\mathcal{C}_k(\tau)$:

$$\forall k, \quad \mu_k(\tau + 1) = \frac{1}{|\mathcal{C}_k(\tau)|} \sum_{\mathbf{x}_i \in \mathcal{C}_k(\tau)} \mathbf{x}_i \quad (4)$$

En alternant les deux étapes d'assignation des cellules et de mise à jour du dictionnaire, il est prouvé que $\mathcal{M}(\tau)$ atteint à convergence un minimum local de la MQE. Pour éviter de converger vers un minimum local présentant des cellules vides ou déséquilibrées, des techniques telles que le codeword-shifting ont été proposées pour forcer la réassignation de cellules de faible densité à des régions de densité plus élevée (Patanè, Russo, 2000; Patanè, Russo, 2001; Lu, Chang, 2010) et ainsi produire de meilleurs dictionnaires.

Cet algorithme fait implicitement l'hypothèse que l'ensemble des échantillons est directement accessible à chaque instant, ce qui n'est pas le cas lorsque les documents à traiter sont dispersés sur un réseau. La solution naïve consistant à transférer tous les échantillons sur une unique machine n'est pas viable à grande échelle. En effet, le débit disponible sur le réseau et la puissance de calcul de la machine effectuant le traitement doivent croître proportionnellement au nombre de documents à indexer, ce qui est impensable en l'état actuel des technologies matérielles. Malgré les nombreuses propositions émises pour réduire l'accès aux données ou la complexité calculatoire (approches par streaming (Shindler *et al.*, 2011; Ailon *et al.*, 2009), par apprentissage compétitif (Kohonen, 1990), par plus proches voisins approximatés (Ailon *et al.*, 2009), etc...), aucune solution impliquant une unité de calcul unique ne permet pour le moment d'apprendre un dictionnaire visuel à très grande échelle. Nous nous tournons donc vers une stratégie distribuée pour calculer les dictionnaires visuels.

2.3. Apprentissage distribué de dictionnaires

La nécessité de modéliser de très grandes masses de données, qui a émergé au sein de nombreuses disciplines a conduit à la formulation de plusieurs solutions distribuées pour l'apprentissage de dictionnaires. Les environnements distribués sont de nature très diverse, aussi l'applicabilité d'une solution donnée dépend fortement du contexte de mise en œuvre. Nous présentons ici les contraintes liées à l'apprentissage distribué de dictionnaires pour une indexation large échelle, puis nous détaillons les différentes stratégies proposées dans la littérature pour répondre à ce problème.

2.3.1. Contraintes à respecter

L'environnement considéré est un réseau de N nœuds, où chaque nœud i héberge un jeu d'échantillons local $\mathbf{X}^{(i)} \subset \mathbb{R}^D$ et doit produire un dictionnaire local $\mathcal{M}^{(i)}$. L'objectif est que tous ces dictionnaires soient égaux à un unique dictionnaire \mathcal{M}^* , qualifié de *consensus*, qui minimise la MQE sur $\mathbf{X} = \bigcup_i \mathbf{X}^{(i)}$. Nous ciblons de larges réseaux grand public (web, réseaux mobiles, etc. . .), et plus spécifiquement les réseaux pair-à-pair (P2P). Le système doit satisfaire les six contraintes suivantes :

- C1** Les échantillons ne doivent pas être échangés sur le réseau, certains contenus étant non transférables, pour des raisons de taille, de quantité de documents, de confidentialité ou de protection de la vie privée.
- C2** Un mot visuel μ_k doit avoir le même indice k sur tous les nœuds pour permettre l'utilisation directe de métriques vectorielles sur les signatures (*e.g.* euclidienne, produit scalaire, . . .).
- C3** L'algorithme doit minimiser les coûts de communications.
- C4** Aucun nœud ne peut être considéré comme plus fiable qu'un autre. Le système doit conserver ses performances de modélisation en présence de fautes ou de connexions/déconnexions imprévisibles.
- C5** Le système doit éviter toute synchronisation entre les nœuds.
- C6** La charge de calcul doit être équitablement répartie selon les capacités hétérogènes des machines.

2.3.2. Stratégies existantes

Nous nous concentrons sur des méthodes ayant en commun deux éléments principaux : une procédure locale d'optimisation de dictionnaires, qui n'implique aucune communication, et un opérateur de combinaison par échange de dictionnaires entre nœuds. Ces méthodes satisfont donc **C1** car seuls les dictionnaires sont échangés. Toutefois, l'ordonnancement de ces deux éléments et la nature de l'opérateur de combinaison choisi détermine fortement la satisfaction des autres contraintes.

Une approche courante est de conserver les deux étapes itérées de K -means et les adapter à l'environnement distribué. L'étape 1 (assignation) se prête naturellement à un calcul indépendant en chaque nœud, mais l'étape 2 (mise à jour du dictionnaire) requiert le calcul distribué de moyennes vectorielles pondérées. Une autre stratégie consiste à calculer un dictionnaire local à chaque nœud puis combiner ces dictionnaires par échanges de messages pour construire le dictionnaire consensus.

Dans (Durut *et al.*, 2012), les auteurs distribuent l'étape 2 en utilisant un nœud maître. Chaque nœud calcule des moyennes partielles, puis les transmet au nœud maître qui calcule la moyenne globale et la restitue aux autres nœuds à chaque itération. Dans (Januzaj *et al.*, 2004), chaque nœud calcule d'abord un dictionnaire local puis les dictionnaires locaux sont combinés par un nœud maître en effectuant un clustering de leurs centroïdes. De telles combinaisons centralisées ne satisfont pas les contraintes **C4** à **C6**, car elles impliquent un déséquilibre des charges de calcul et de

communications entre nœuds, une grande sensibilité aux fautes et une synchronisation globale.

Dans (Eisenhardt *et al.*, 2003), les auteurs envisagent l'agrégation le long d'un parcours défini du réseau (*e.g.* un arbre couvrant par protocole Echo-Probe). Bien que les coûts de communication soient réduits, les contraintes **C4** et **C5** ne sont pas satisfaites. En effet, tous les nœuds doivent attendre que l'agrégation soit terminée pour poursuivre et toute erreur de communication ou déconnexion d'un nœud particulier peut entraîner l'écroulement du système entier.

Une solution pour satisfaire les contraintes **C4** et **C6** est de concevoir un opérateur de combinaison décentralisé, *i.e.* qui utilise uniquement des interactions locales entre nœuds voisins. Dans les approches de (Bandyopadhyay *et al.*, 2006) et (Datta *et al.*, 2006), chaque nœud réalise l'étape 2 de K -means en estimant la moyenne globale grâce aux moyennes partielles récupérées de quelques nœuds voisins. L'erreur induite par cette estimation croît avec l'hétérogénéité de la distribution des données sur le réseau. Les dictionnaires produits peuvent alors être significativement différents entre les nœuds, ce qui ne satisfait pas notre objectif de départ.

Dans (Nikseresht, Gelgon, 2008) les auteurs proposent un algorithme décentralisé qui optimise des modèles statistiques plus fins tels que des mélanges de gaussiennes (GMM), mais qui s'adapte aisément au simple calcul de dictionnaires visuels. Chaque nœud optimise un dictionnaire local, puis les dictionnaires sont agrégés itérativement entre nœuds voisins. L'agrégation consiste à concaténer les modèles reçus en un seul modèle, puis réduire la complexité de ce modèle en minimisant une KL-divergence entre le modèle concaténé et le modèle réduit. Cette méthode donne d'excellents résultats et satisfait **C3** à **C6**, mais viole **C2** car la construction du modèle réduit détruit l'ordre des mots dans le dictionnaire.

Newscast-EM (Kowalczyk, Vlassis, 2004) est une version décentralisée de l'algorithme *Expectation-Maximization* (EM) (Dempster *et al.*, 1977). L'étape de maximisation, qui correspond à la moyenne pondérée de l'étape 2 de K -means est confiée à un protocole d'agrégation *Newscast*. *Newscast* (Jelasity *et al.*, 2003) est un protocole Gossip qui permet le calcul d'agrégats par une estimation itérative impliquant des interactions locales entre paires de nœuds. La convergence exponentielle vers le consensus est formellement garantie. Contrairement à (Nikseresht, Gelgon, 2008), **C2** est satisfaite, mais *Newscast* demande une synchronisation ponctuelle entre paires de nœuds, qui peut être complexe à maintenir lorsqu'un nœud est investi dans plusieurs interactions simultanées, et source d'erreurs si un nœud se déconnecte au cours d'une interaction. *Newscast-EM* ne satisfait donc pas **C5**.

Epidemic K-means (Fatta *et al.*, 2011 ; 2012) utilise également un protocole Gossip pour apporter une solution décentralisée à K -means. Pour réaliser la moyenne pondérée de l'étape 2, il recourt à un protocole *Sum-Weight* (Kempe *et al.*, 2003; Bénézit *et al.*, 2010; Iutzeler *et al.*, 2013) qui autorise un fonctionnement totalement asynchrone. Chaque nœud gère ainsi indépendamment ses rôles d'émetteur et de récepteur. Tout comme *Newscast*, les protocoles *Sum-Weight* ont une convergence ex-

Tableau 1. Avantages et inconvénients de méthodes de clustering distribué existantes vis-à-vis des contraintes C1 à C6

METHOD	C1	C2	C3	C4	C5	C6
DURUT12	✓	✓	✓			
JANUZAJ04	✓	✓	✓			
EISENHARDT03	✓	✓	✓			✓
BANDYOPADHYAY06	✓		✓	✓		✓
DATTA06	✓		✓	✓		✓
NIKSERESHT08	✓		✓	✓	✓	✓
KOWALCZYK04	✓	✓	✓	✓		✓
FATTA12	✓	✓	~	✓	~	✓

ponentiellement rapide et formellement établie (Kempe *et al.*, 2003; Iutzeler *et al.*, 2013). Dans (Fatta *et al.*, 2012), les auteurs ne fournissent pas de règle pour définir le nombre de messages échangés à chaque itération et le fixent arbitrairement, ce qui entraîne un coût de communication superflu.

Comme synthétisé dans le tableau 1, l’approche Epidemic K -means est l’alternative qui satisfait le plus de contraintes. Une méthode de détermination du nombre de messages à échanger doit être cependant définie pour améliorer son efficacité. Nous détaillons ci-après les protocoles d’agrégation Gossip permettant le calcul distribué de moyennes pondérées, sur lesquels repose notamment Epidemic K -means.

2.4. Calcul de moyennes par protocoles Gossip

L’estimation décentralisée de moyennes consiste à estimer la moyenne de N valeurs $\{v_1, \dots, v_N\}$ hébergées sur N nœuds du réseau, en ne recourant qu’à des échanges locaux entre nœuds voisins. Si t est une mesure du temps global, le but est donc de faire converger des estimées locales $v_i(t)$, $1 \leq i \leq N$ vers $\bar{v} = \frac{1}{N} \sum_i v_i(0)$. Ce problème peut être résolu par un protocole Gossip.

Les protocoles Gossip (ou épidémiques) ont été proposés pour répondre aux problématiques de dissémination d’information (Demers *et al.*, 1988) puis de calcul d’agrégats (Kempe *et al.*, 2003) en environnement décentralisé. Particulièrement robustes, ces protocoles sont adaptés à des réseaux point-à-point très larges à la connectivité hétérogène et changeante (notamment les réseaux P2P, mobiles ou ad hoc tels que les réseaux de capteurs). Pour calculer une moyenne, leur principe général est le suivant :

- Chaque nœud i maintient une estimée $v_i(t)$ initialisée à v_i .
- Chaque nœud envoie régulièrement des fractions de son estimée v_i à un ou plusieurs nœuds voisins, et garde la fraction restante.
- Lorsqu’un nœud i reçoit une valeur v , il l’ajoute à son estimée locale $v_i(t)$.

A chaque instant global t , chaque nœud i est donc susceptible d’envoyer une fraction $\kappa_{ij}(t) \geq 0$ de $v_i(t)$ à chaque nœud j et en conserve une fraction $\kappa_{ii}(t)$, l’absence de

message entre i et j étant traduite par $\kappa_{ij}(t) = 0$. Les coefficients $\kappa_{ij}(t)$ sont choisis aléatoirement suivant une distribution prédéfinie. Le processus stochastique linéaire ainsi défini s'écrit sous forme matricielle :

$$\mathbf{v}(t+1)^T = \mathbf{v}(t)^T \mathbf{K}(t), \quad \text{avec} \quad \begin{cases} \mathbf{K}(t) = [\kappa_{ij}(t)]_{ij} \\ \mathbf{v}(t) = (v_1(t), \dots, v_N(t)) \end{cases} \quad (5)$$

La distribution des matrices $\mathbf{K}(t)$, appelées *matrices de diffusion*, caractérise complètement un protocole donné. Elle doit respecter une contrainte de *conservation de masse* : $\forall t, i, \sum_j \kappa_{ij}(t) = 1$. Cette contrainte garantit que la moyenne des estimées courantes reste toujours égale à la moyenne initiale \bar{v} et équivaut à imposer $\mathbf{K}(t)\mathbf{1} = \mathbf{1}$. Ainsi, si tous les nœuds convergent vers la même estimée celle-ci vaudra \bar{v} .

$$\forall t, \quad \frac{1}{N} \mathbf{v}(t+1)\mathbf{1} = \frac{1}{N} \mathbf{v}(t)^T \mathbf{K}(t)\mathbf{1} = \frac{1}{N} \mathbf{v}(t)^T \mathbf{1} = \bar{v} \quad (6)$$

Un protocole est parfaitement *asynchrone* s'il n'y a pas plus d'un message échangé à chaque instant global t (le processus d'échange de message est ainsi Poissonien). Dans ce cas, toutes ses matrices de diffusion ont au plus un seul coefficient hors-diagonale non nul. De plus, la convergence n'est assurée que si l'état où toutes les estimées sont égales est un *point fixe* des matrices de diffusion (*i.e.* $\bar{v}\mathbf{1}^T \mathbf{K}(t) = \bar{v}\mathbf{1}^T$). Les trois contraintes de conservation de masse, de point fixe et d'asynchronisme ne peuvent pas être satisfaites simultanément (la seule matrice qui les satisfait est l'identité).

Certains protocoles tels que Newscast (Jelasity *et al.*, 2003) ou (Boyd *et al.*, 2006) choisissent de relâcher la contrainte d'asynchronisme en supposant que des messages symétriques entre paires de nœuds peuvent être échangés de manière atomique. Dans des réseaux dynamiques au routage précaire, cette hypothèse n'est pas viable.

Les protocoles Gossip dits « Sum-Weight » (Kempe *et al.*, 2003; Bénézit *et al.*, 2010; Iutzeler *et al.*, 2013) permettent de relâcher la contrainte de point fixe et autorisent le calcul de moyennes pondérées. Chaque estimée $v_i(t)$ est couplée à un poids $w_i(t)$ mis à jour selon les mêmes règles. En choisissant des poids initiaux $w_i(0)$ arbitraires et en fixant $v_i(0) = w_i(0)v_i$, les quotients $v_i(t)/w_i(t)$ convergent vers la moyenne des v_i pondérée par les $w_i(0)$ (Kempe *et al.*, 2003). En notant $\mathbf{w}(t) = (w_1(t), \dots, w_N(t))$ et $\frac{\mathbf{v}}{\mathbf{w}}$ le quotient terme à terme des vecteurs \mathbf{v} et \mathbf{w} on obtient :

$$\begin{cases} \mathbf{v}(t+1)^T = \mathbf{v}(t)^T \mathbf{K}(t) \\ \mathbf{w}(t+1)^T = \mathbf{w}(t)^T \mathbf{K}(t) \end{cases} \Rightarrow \frac{\mathbf{v}(t)}{\mathbf{w}(t)} \rightarrow \frac{\mathbf{v}(0)^T \mathbf{1}}{\mathbf{w}(0)^T \mathbf{1}} = \frac{\sum_i w_i(0)v_i}{\sum_i w_i(0)} \mathbf{1} \quad (7)$$

La vitesse à laquelle les estimées convergent vers la moyenne est étroitement liée à la conductance du graphe de connectivité du réseau (Shah, 2009). Lorsque ce graphe est une clique ou a des propriétés d'expandeur, l'erreur d'estimation suit une décroissance exponentielle rapide (Iutzeler *et al.*, 2013). Pour cette famille de réseaux, le

temps de convergence croît logarithmiquement avec le nombre de nœuds, ce qui permet d'envisager de très grands réseaux en conservant un coût de communication raisonnable.

3. Système d'indexation multimédia décentralisé et asynchrone

Le système d'indexation que nous proposons apporte une solution totalement décentralisée et asynchrone à l'indexation par le contenu de collections multimédias réparties sur les réseaux. Ce système repose sur le calcul en chaque nœud d'un dictionnaire visuel local qui modélise l'ensemble des documents du réseau. La mesure de similarité obtenue est cohérente sur l'ensemble du réseau, et permet une comparaison directe de documents provenant de sites distincts.

Dans cette section nous présentons l'architecture du système d'indexation proposé, puis nous détaillons l'algorithme décentralisé de calcul des dictionnaires visuels locaux.

3.1. Architecture du système d'indexation

Le système d'indexation que nous proposons est une extension décentralisée de la chaîne d'indexation par dictionnaire visuel classique. Sa mise en œuvre implique le choix d'un type de descripteur visuel (*e.g.* SIFT pour des images) et d'un opérateur d'agrégation (BoW, VLAD, VLAT, ...). Étant donné un réseau de N nœuds possédant chacun un jeu local de documents multimédias, la procédure d'indexation est la suivante :

1. Chaque nœud extrait un sac de descripteurs visuels locaux pour chacun de ses documents.
2. Chaque nœud calcule un dictionnaire visuel local qui modélise l'ensemble des contenus du réseau, grâce à l'algorithme de K -means décentralisé que nous présentons dans la section suivante.
3. Chaque nœud produit et stocke les signatures associées à ses documents en projetant chaque sac de descripteurs sur son dictionnaire local via la méthode d'agrégation choisie.

A l'issue de cette procédure, chaque document sur le réseau est associé à une signature compacte stockée sur le nœud qui l'héberge. Pour comparer deux documents, on applique directement une métrique vectorielle (*e.g.* L_2 , produit scalaire) sur leurs signatures.

3.2. Calcul du dictionnaire visuel par protocole Gossip

L'algorithme que nous proposons pour l'apprentissage décentralisé de dictionnaire visuel est une extension distribuée de K -means (*cf.* algorithme 1) inspirée de (Fatta *et*

Algorithme 1. Gossip K-Means avec codeword-shifting

```

1 Pour le nœud  $i$ 
  Input:
   $\mathbf{X}^{(i)} = \{\mathbf{x}^{(i)}\} \subset \mathbf{X}$  : Jeu de données d'entraînement pour le nœud  $i$ 
   $K$  : nombre de cellules désirées
   $\Delta_{min}$  : seuil d'amélioration relative de la MQE pour déclencher le
  codeword-shifting ou terminer
   $\sigma_{shift}$  : RSD des distorsions des cellules requise pour terminer
  Output: Un dictionnaire  $\mathcal{M} = [\mu_k]_k$  pour le nœud  $i$ 
2 begin
3   Initialiser  $\mathcal{M} = [\mu_k]_k$  aléatoirement ;  $MQE_{old} \leftarrow \infty$  ;
4   repeat
5      $(n_k)_k \leftarrow \mathbf{0}_K$  ;  $[\mathbf{s}_k]_k \leftarrow \mathbf{0}_{K \times D}$  ;  $(d_k)_k \leftarrow \mathbf{0}_K$ 
6     foreach  $\mathbf{x} \in \mathbf{X}_i$  do
7        $k \leftarrow \arg \min_{k'} \|\mathbf{x} - \mu_{k'}\|_2^2$ 
8        $n_k += 1$  ;  $\mathbf{s}_k += \mathbf{x}$  ;  $d_k += \|\mathbf{x} - \mu_k\|_2^2$ 
9     end
10    Lancer ProcEmission() et ProcReception() concurremment
11     $\mathcal{M} \leftarrow [\mathbf{s}_k/w_k]_k$  ;  $(d_k)_k \leftarrow (d_k/w_{K+1})_k$  ;  $MQE \leftarrow \frac{1}{K} \sum_k d_k$ 
12    while CodewordShifting()
13 end

```

Algorithme 2. ProcEmission

```

1  $\mathbf{w} \leftarrow (n_1, \dots, n_k, 1)$ 
2 for  $t \leftarrow 1$  to  $M$  do
3   Choisir un nœud voisin  $j \in \mathcal{N}_i$  aléatoirement
4    $([\mathbf{s}_k]_k, (d_k)_k, \mathbf{w}) \leftarrow \frac{1}{2}([\mathbf{s}_k]_k, (d_k)_k, \mathbf{w})$ 
5   Send  $([\mathbf{s}_k]_k, (d_k)_k, \mathbf{w})$  to  $j$ 
6 end

```

Algorithme 3. ProcReception

```

1 repeat
2   A réception d'un message  $([\mathbf{s}_k^{(j)}]_k, (d_k^{(j)})_k, \mathbf{w}^{(j)})$  d'un nœud  $j$ 
3    $([\mathbf{s}_k]_k, (d_k)_k, \mathbf{w}) += ([\mathbf{s}_k^{(j)}]_k, (d_k^{(j)})_k, \mathbf{w}^{(j)})$ 
4 while ProcEmission est active

```

al., 2012). Nous étendons (Fatta *et al.*, 2012) en ajoutant une procédure de codeword-shifting dérivée de (Patanè, Russo, 2001) ainsi qu'une règle pour calculer le nombre de messages que chaque nœud doit envoyer à chaque itération. Le protocole Gossip

asynchrone que nous proposons, plus simple que celui de (Fatta *et al.*, 2012), ne recourt pas à des échanges bilatéraux mais offre des performances équivalentes.

L'algorithme est totalement décentralisé dans la mesure où chaque nœud exécute un processus identique de manière complètement indépendante. Chaque nœud i débute avec un jeu d'échantillons local $\mathbf{X}^{(i)} \subset \mathbb{R}^D$ et un dictionnaire local $\mathcal{M}^{(i)}(0) = \{\mu_1^{(i)}(0), \dots, \mu_K^{(i)}(0)\}$. A la différence de (Fatta *et al.*, 2012), chaque dictionnaire local est initialisé aléatoirement sans coordination entre nœuds. La procédure exécutée sur un nœud i consiste en trois étapes itérées. A chaque itération τ de son processus local, un nœud effectue une première étape d'optimisation locale, qui n'implique aucune communication. Puis il estime le dictionnaire consensus courant par échange de messages au moyen d'un protocole d'agrégation Gossip. Si un critère de convergence est vérifié, il déclenche alors une procédure locale de codeword-shifting. Le dictionnaire courant sert alors d'initialisation à l'étape d'optimisation locale de l'itération $\tau + 1$. Ces trois étapes sont détaillées ci-après pour un nœud quelconque i .

3.2.1. Étape 1 : optimisation locale

Chaque échantillon local est assigné à la cellule qui a le mot visuel le plus proche dans le dictionnaire courant $\mathcal{M}^{(i)}(\tau)$. Simultanément, on calcule la taille, la somme et la distorsion de chaque cellule $\mathcal{C}_k^{(i)}$ (respectivement notées $n_k^{(i)}(\tau)$, $\mathbf{s}_k^{(i)}(\tau)$ et $d_k^{(i)}(\tau)$):

$$\mathcal{C}_k^{(i)}(\tau) = \{\mathbf{x} \in \mathbf{X}^{(i)} / k = \arg \min_{k'} \|\mathbf{x} - \mu_{k'}^{(i)}(\tau)\|_2^2\} \quad (8)$$

$$n_k^{(i)}(\tau) = |\mathcal{C}_k^{(i)}(\tau)| \quad \mathbf{s}_k^{(i)}(\tau) = \sum_{\mathbf{x} \in \mathcal{C}_k^{(i)}} \mathbf{x} \quad d_k^{(i)}(\tau) = \sum_{\mathbf{x} \in \mathcal{C}_k^{(i)}} \|\mathbf{x} - \mu_k^{(i)}(\tau)\|_2^2 \quad (9)$$

3.2.2. Étape 2 : estimation décentralisée du dictionnaire consensus

Le dictionnaire consensus courant est estimé en échangeant des messages d'agrégation avec les nœuds voisins. Celui-ci est défini comme le dictionnaire qui minimise la MQE sur \mathbf{X} compte tenu des cellules trouvées par tous les nœuds à cet instant. Le regroupement de ces cellules par indice déterminant une partition $\{\bigcup_j \mathcal{C}_1^{(j)}, \dots, \bigcup_j \mathcal{C}_K^{(j)}\}$ de l'ensemble des données $\bigcup_i \mathbf{X}^{(i)}$, le dictionnaire consensus $\mathcal{M}^*(\tau)$ est donc formé par les barycentres des $\bigcup_j \mathcal{C}_1^{(j)}, \dots, \bigcup_j \mathcal{C}_K^{(j)}$:

$$\forall k, \quad \mu_k^*(\tau) = \frac{1}{|\bigcup_i \mathcal{C}_k^{(i)}|} \sum_i \sum_{\mathbf{x} \in \mathcal{C}_k^{(i)}} \mathbf{x} = \frac{\sum_i \mathbf{s}_k^{(i)}}{\sum_i n_k^{(i)}} \quad (10)$$

Ces barycentres sont estimés via un protocole Gossip Sum-Weight asynchrone, à partir des sommes $\mathbf{s}_k^{(i)}(\tau)$ et des taille $n_k^{(i)}(\tau)$ des cellules obtenues à l'étape 1. Chaque nœud i maintient une matrice d'estimées $\mathbf{S}^{(i)} = [\mathbf{s}_1^{(i)} \dots \mathbf{s}_K^{(i)}]$ dont chaque colonne k est couplée à un poids $w_k^{(i)}$, ainsi qu'un vecteur de distorsions $\mathbf{d}^{(i)}$ de dimension K couplé à

un poids supplémentaire $w_{K+1}^{(i)}$. Leurs valeurs initiales sont fixées comme suit : $\mathbf{S}^{(i)} = [\mathbf{s}_1^{(i)}(\tau) \dots \mathbf{s}_K^{(i)}(\tau)]$, $\mathbf{d}^{(i)} = (d_1^{(i)}(\tau), \dots, d_K^{(i)}(\tau))$, et $\mathbf{w}^{(i)} = (n_1^{(i)}(\tau), \dots, n_K^{(i)}(\tau), 1)$.

Le nœud i lance alors une procédure d'émission et une procédure de réception concurremment :

Procédure d'émission (algorithme 2). De manière répétée, i divise tous les coefficients de sa matrice d'estimées $\mathbf{S}^{(i)}$, son vecteur de distorsions $\mathbf{d}^{(i)}$, et son vecteur de poids $\mathbf{w}^{(i)}$ par 2 et les envoie à un nœud voisin j choisi aléatoirement.

Procédure de réception (algorithme 3). A réception d'un message d'un nœud j , i ajoute les valeurs reçues aux coefficients de ses propres $\mathbf{S}^{(i)}$, $\mathbf{d}^{(i)}$ et $\mathbf{w}^{(i)}$.

Les quotients $\{\mathbf{s}_k^{(i)}/w_k^{(i)}\}_k$ convergent alors vers le dictionnaire consensus $\mathcal{M}^*(\tau)$ et les composantes vecteur $\mathbf{d}^{(i)}/w_{K+1}^{(i)}$ convergent vers la distorsion moyenne de chaque cellule $\frac{1}{N} \sum_i d_k^{(i)}$.

D'un point de vue global, notre protocole est parfaitement asynchrone, puisque chaque instant t correspond à un unique échange de message impliquant un émetteur i et un récepteur $j \neq i$, ce qui équivaut à appliquer l'équation (Eq. 5) à chaque coefficient des $(\mathbf{S}^{(i)}, \mathbf{d}^{(i)}, \mathbf{w}^{(i)})$ courants avec :

$$\mathbf{K}(t) = \mathbf{I} + \frac{1}{2} \mathbf{e}_i (\mathbf{e}_j - \mathbf{e}_i)^T \quad \text{avec} \quad \begin{cases} i \text{ choisi aléatoirement dans } \{1 \dots N\} \\ j \text{ choisi aléatoirement dans } \{1 \dots N\} \setminus i \end{cases} \quad (11)$$

3.2.3. Étape 3 : codeword-shifting

Algorithme 4. CodewordShifting

```

1 if  $0 \leq (MQE_{old} - MQE)/MQE_{old} < \Delta_{min}$  then
2   if  $\frac{1}{MQE} \sqrt{\frac{1}{K} \sum_k (d_k - MQE)^2} < \sigma_{shift}$  then return Faux;
3   else
4      $k_{max} = \arg \max_k d_k$ ;  $k_{min} = \arg \min_k d_k$ ;
5      $\mu_{k_{min}} \leftarrow \mu_{k_{max}} + \eta$ ;  $\mu_{k_{max}} \leftarrow \mu_{k_{max}} - \eta$ ;
6   end
7 end
8  $MQE_{old} \leftarrow MQE$ ;
9 return Vrai

```

Nous ajoutons une procédure de codeword-shifting locale aux nœuds, dérivée de l'approche centralisée de (Patanè, Russo, 2001). Elle permet la recherche de meilleurs dictionnaires lorsque l'algorithme de K -means a convergé vers une configuration stable mais sous-optimale (cellules vides ou déséquilibrées). Elle est déclenchée par un nœud i lorsque les deux conditions suivantes sont satisfaites :

1. Son estimée $\text{MQE}^{(i)}$ de l'erreur quadratique moyenne globale a convergé vers un minimum local (son amélioration relative est inférieure à Δ_{min}). $\text{MQE}^{(i)}$ est obtenue à partir des estimées de distorsions $\mathbf{d}^{(i)}$ calculées à l'étape 2.

2. L'erreur quadratique du modèle est mal répartie entre les cellules. Cette condition est satisfaite lorsque l'écart-type relatif de la distorsion des cellules $\text{RSD}^{(i)}$ est supérieur à σ_{shift} .

$$\text{MQE}^{(i)} = \frac{1}{K} \sum_{k=1}^K d_k^{(i)} \quad \text{et} \quad \text{RSD}^{(i)} = \frac{1}{\text{MQE}^{(i)}} \sqrt{\frac{1}{K} \sum_{k=1}^K \left(d_k^{(i)} - \text{MQE}^{(i)} \right)^2} \quad (12)$$

La procédure de codeword-shifting consiste à déplacer le centroïde associé à la cellule de plus faible distorsion k_{min} vers le centroïde associé à la cellule de plus forte distorsion k_{max} . On remplace alors $\mu_{k_{min}}$ par $\mu_{k_{max}} + \eta$, avec η un petit vecteur arbitraire. Bien que moins évoluée que (Patanè, Russo, 2001), cette fonctionnalité permet de relancer l'optimisation afin d'égaliser les distorsions des cellules. Les dictionnaires produits ont une erreur quadratique mieux répartie entre les cellules et gagnent ainsi en précision.

L'algorithme termine en un nœud i lorsque son estimée $\text{MQE}^{(i)}$ a convergé et que $\text{RSD}^{(i)}$ est passée sous le seuil σ_{shift} . Il retourne alors un dictionnaire $\mathcal{M}^{(i)}(\tau + 1)$ dont la MQE est localement optimale et dont l'erreur quadratique est équitablement répartie entre les cellules.

3.3. Paramètres de l'algorithme de calcul de dictionnaires

L'algorithme que nous proposons est réglé par l'ensemble des paramètres suivants :

- Δ_{min} définit le seuil de déclenchement du codeword-shifting. Si l'amélioration relative de la MQE est inférieure à ce seuil, l'algorithme considère qu'il a convergé vers un minimum local de la MQE. Plus Δ_{min} est proche de 1, plus la procédure est lancée fréquemment en un nœud, et inversement.

- σ_{shift} est un seuil qui permet de déterminer, lorsque l'algorithme a atteint un minimum local, si le dictionnaire obtenu à une répartition suffisamment équitable de l'erreur de quantification entre les cellules pour terminer, ou si cette répartition est inéquitable et nécessite un codeword-shifting. Une valeur proche de 0 force une répartition totalement équitable, tandis qu'une valeur proche de 1 relâche cette contrainte (la procédure de codeword-shifting tend alors à n'être jamais exécutée et Δ_{min} devient un critère d'arrêt).

- M est un paramètre du protocole d'estimation du consensus qui contrôle le nombre de messages envoyés par chaque nœud à chacune de ses itérations. Son rôle est critique, puisqu'une valeur trop faible peut empêcher la convergence de l'estimation vers le consensus et entraîner le dysfonctionnement de l'algorithme. A contrario, une valeur élevée assure que l'estimation décentralisée a bien le temps de converger vers le dictionnaire consensus, mais augmente les coûts de communication. Dans (Fatta *et*

al., 2012), le nombre de messages échangés à chaque itération est fixé arbitrairement, sans lien avec le nombre N de noeuds considérés. Pourtant, il est prouvé dans (Iutzeler *et al.*, 2013) que le nombre de messages requis pour atteindre une erreur d'estimation donnée a une dépendance logarithmique sur le nombre de noeuds du réseau lorsque le réseau possède certaines propriétés de connectivité (détaillées dans (Iutzeler *et al.*, 2013)). L'étude expérimentale présentée dans la section suivante détermine une loi empirique qui permet de fixer le paramètre M étant donnés le nombre de noeuds dans le réseau et la précision d'estimation attendue.

4. Expériences et résultats

Nous présentons dans cette section les résultats et l'analyse de deux campagnes d'évaluation. La première se concentre sur la qualité des dictionnaires produits par l'algorithme proposé, que nous évaluons grâce à des mesures de la littérature (Bandyopadhyay *et al.*, 2006; Datta *et al.*, 2009; Fatta *et al.*, 2012). La seconde étudie les performances du système d'indexation décentralisé dans son ensemble, via des mesures courantes de qualité de recherche en environnement centralisé (précision moyenne).

4.1. Dispositif expérimental

Afin d'évaluer les performances de notre système sur de grandes topologies de réseaux (plusieurs milliers de noeuds), nous avons mis en place une plateforme de simulation logicielle qui permet d'exécuter un nombre arbitraire de noeuds virtuels sur un ensemble de machines matérielles. Nous avons utilisé jusqu'à 20 machines matérielles (dont 4 calculateurs de 24 cœurs et 1 de 64 cœurs). Pour demeurer indépendant de l'ordonnancement des opérations par les cœurs de calcul, les messages d'agrégation sont déclenchés par une horloge Poissonienne globale. En empêchant ainsi l'échange simultané de plusieurs messages, le système simulé garantit un fonctionnement parfaitement asynchrone.

4.2. Évaluation de l'algorithme de calcul de dictionnaires visuels

Nous évaluons notre algorithme de calcul décentralisé de dictionnaires visuels à la fois sur des données synthétiques et des données réelles, grâce à trois indicateurs de qualité des dictionnaires locaux utilisés en environnement distribué :

La MQE sur l'ensemble des données du réseau (MQE_g) évalue les capacités globales de modélisation et de généralisation de l'algorithme.

$$\text{MQE}_g = \frac{1}{NK} \sum_{k=1}^{N,K} \sum_{\mathbf{x} \in \mathcal{C}_k^{(i)}} \left\| \mathbf{x} - \mu_k^{(i)} \right\|_2^2 \quad (13)$$

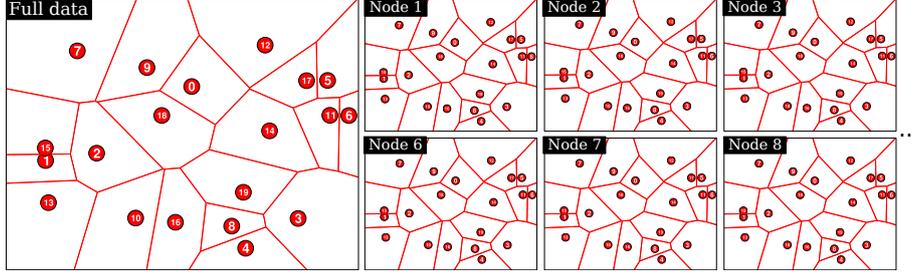


Figure 2. Dictionnaires produits par notre algorithme à partir de données locales synthétiques hétérogènes. Les dictionnaires modélisent correctement l'union des ensembles d'apprentissage locaux (cf. partie gauche)

L'erreur relative par rapport au consensus (*Relative Error to Consensus*, REC) mesure la cohérence entre nœuds en termes d'erreur euclidienne normalisée entre les dictionnaires locaux et le dictionnaire consensus.

$$\text{REC} = \frac{1}{N} \sum_{i=1}^N \frac{\|\mathcal{M}^{(i)} - \mathcal{M}^*\|_F^2}{\|\mathcal{M}^*\|_F^2} \quad (14)$$

Le pourcentage d'erreur d'assignation (*Percentage of Membership Mismatches*, PMM) mesure la cohérence des quantificateurs $q^{(i)}$ associés aux dictionnaires locaux, par rapport à celui obtenu avec le dictionnaire consensus q^* .

$$\text{PMM} = \frac{1}{N|\mathbf{X}|} \sum_{i=1}^N \left| \{ \mathbf{x} / q^{(i)}(\mathbf{x}) \neq q^*(\mathbf{x}) \} \right| \quad (15)$$

4.2.1. Expériences sur données synthétiques

Nous évaluons tout d'abord les capacités de notre algorithme à calculer des dictionnaires sur des jeux de données synthétiques. Pour chaque nœud, nous générons un jeu de données échantillonnés selon des lois normales, dont le nombre, les moyennes et les covariances sont choisis aléatoirement de sorte que la distribution des échantillons soit très hétérogène entre nœuds. Les dictionnaires locaux appris par notre algorithme décentralisé sur ces données sont comparées en termes de MQE_g à la MQE obtenue en exécutant un algorithme témoin de K -means centralisé sur l'ensemble des données regroupées. L'algorithme témoin est doté d'une procédure de codeword-shifting similaire à celle que nous proposons.

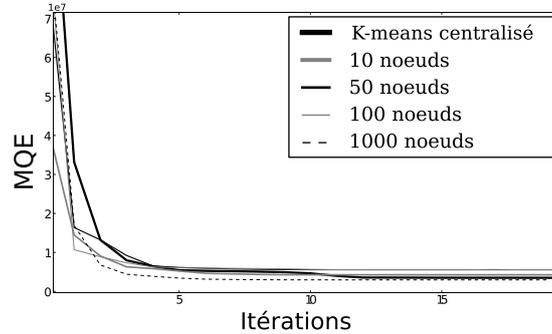


Figure 3. MQE obtenue sur des données similaires à celles de la figure 2

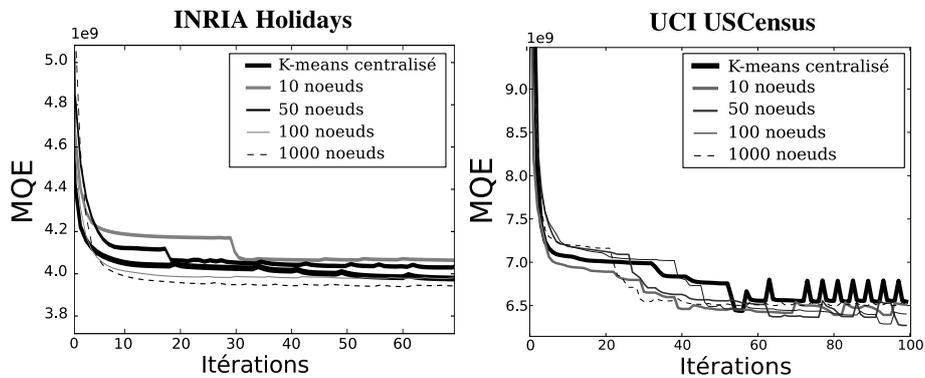


Figure 4. MQE_g obtenue sur les jeux de données UCI USCensus90 et INRIA Holidays avec notre algorithme ou un K-means centralisé sur la totalité des données

Dans un premier temps, M est fixé à une valeur arbitraire ($M = 30$ tel que préconisé dans (Fatta *et al.*, 2012)) et $K = 64$. Un exemple de dictionnaires obtenus est donné par la figure 2. La MQE_g en fonction du nombre d'itérations réalisée par les nœuds est présentée en figure 3. A convergence (après environ 150 messages échangés par nœud), notre algorithme atteint une faible REC (de l'ordre de 10^{-7}) et un PMM strictement nul. Les dictionnaires produits sont donc parfaitement cohérents, en termes de distance entre les mots visuels de même indice, et en termes d'assignation des descripteurs aux cellules. De plus, la figure 3 montre que la MQE obtenue est très proche de celle obtenue par un algorithme de K-means centralisé, en particulier lorsque le dictionnaire initial de l'algorithme centralisé est fixé à $\mathcal{M}^*(0)$. Notre procédure de codeword-shifting apporte un gain comparable à celui obtenu dans un contexte centralisé. Pour des réseaux entre 10 et 1 000 nœuds, les performances sont en pratique quasi indépendantes du nombre de nœuds impliqués et ne sont pas impactées par la nature hétérogène de la répartition des données.

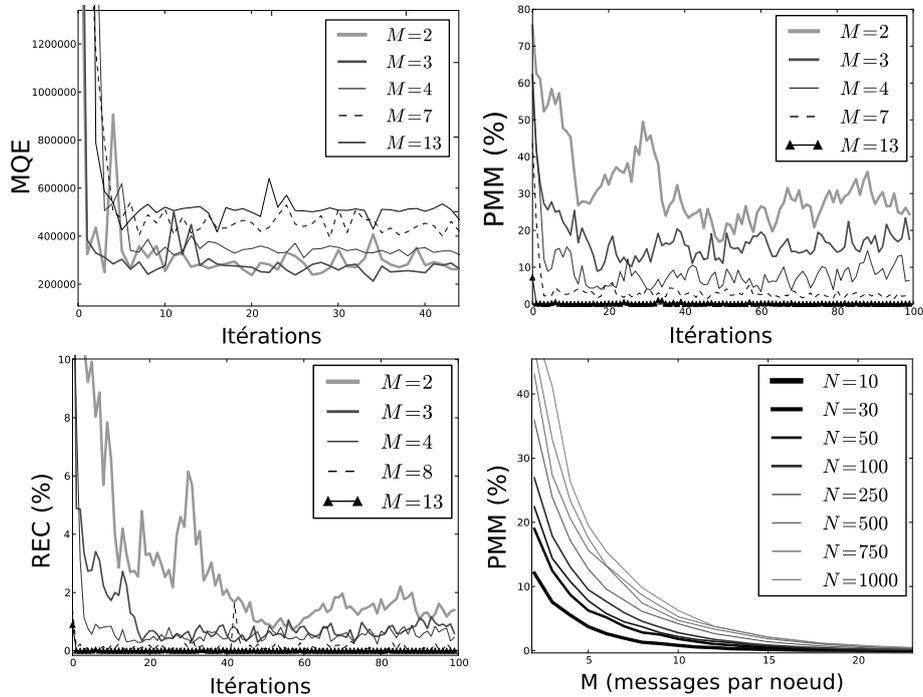


Figure 5. MQE_g , PMM, et REC obtenues pour différentes valeurs de M . En bas à droite : PMM à convergence de l'algorithme pour différentes tailles de réseau N

4.2.2. Expériences sur jeux de données réels

Les propriétés de passage à l'échelle de notre algorithme de calcul de dictionnaires visuels sont mises en évidence sur deux jeux de données de la littérature, à savoir UCI Census 90 et INRIA Holidays (Jégou *et al.*, 2008). Le premier contient environ 2,5 millions d'échantillons composés de 68 attributs, et le deuxième est un jeu d'environ 4,4 millions de descripteurs SIFT en 128 dimensions extraits de 1491 images. Dans les deux cas, les échantillons sont dispersés sur 10, 50, 100 puis 1 000 nœuds. Dans la mesure où les performances se sont avérées indépendantes de l'hétérogénéité de la distribution, nous ne présentons ici que les résultats obtenus pour les distributions les plus hétérogènes.

Pour ces jeux de données, la MQE_g obtenue au cours du temps est présentée dans la figure 4. Les performances en termes de MQE sont comparables à celles offertes par l'algorithme centralisé témoin.

4.2.3. Réduction des coûts de communication

Nous évaluons également les performances des dictionnaires locaux produits lorsque M est réduit. L'algorithme est exécuté sur un même réseau de 100 nœuds

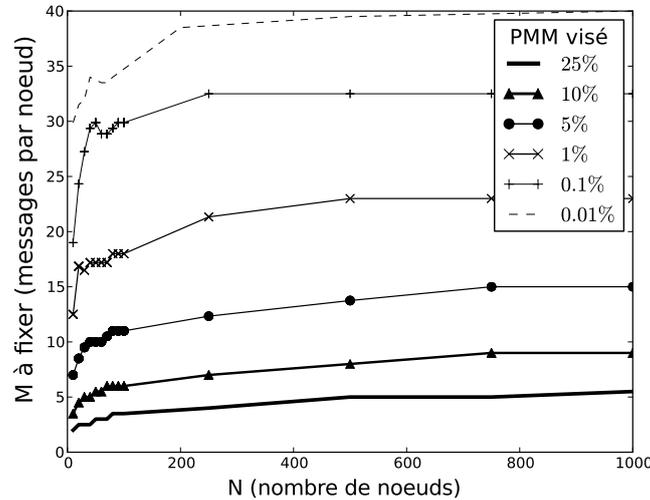


Figure 6. Loi empirique de la valeur de M à fixer pour obtenir un PMM donné sur un réseau de N nœuds

avec des jeux de données synthétiques (générés comme précédemment), pour différentes valeurs de M . La MQE_g , le PMM et la REC obtenus en fonction des valeurs de M sont rassemblés dans la figure 5. On remarque que les performances de notre algorithme décentralisé se maintiennent même lorsque le nombre de messages échangé est très faible. Plus on réduit M , plus les dictionnaires locaux modélisent finement le jeu de données local au détriment de la cohérence entre nœuds. Dans le cas où M est très faible ($M = 2$ par exemple), l'erreur euclidienne entre les dictionnaires est au départ importante, mais elle tombe progressivement sous les 2 %. Notre méthode est donc capable de produire des dictionnaires de qualité avec un coût de communication relativement modique.

4.2.4. Détermination du nombre de messages M

Les expériences menées sur plusieurs tailles de réseau N et plusieurs valeurs de M (cf. figure 5) nous permettent de définir une loi empirique qui, compte tenu du nombre de nœuds impliqués et de la cohérence souhaitée sur les dictionnaires en termes de PMM, détermine le nombre de messages M que chaque nœud doit envoyer à chacune de ses itérations. Cette loi empirique, représentée dans la figure 6 a été estimée sur les 4 millions de descripteurs de la base Holidays, en moyennant les résultats de PMM obtenus pour 2 schémas de répartition des échantillons et 5 initialisations différentes des dictionnaires locaux. Elle s'applique à des réseaux de 10 à 1 000 nœuds pour des valeurs désirées du PMM de 25 % à 0,01 %.

On constate que le nombre de messages à envoyer par nœuds a une croissance sous-linéaire en fonction de la taille du réseau, ce qui atteste des capacités de passage à l'échelle de notre algorithme sans entraîner de surcoût de communications significatif.



Figure 7. Images de la base Holidays (Inria)

tif. On constate notamment que pour de grands réseaux (1 000 nœuds ou plus), seuls 23 messages sont nécessaires à un nœud pour estimer le dictionnaire consensus avec moins de 1 % d'erreur (en termes de PMM). Comparés aux 999 messages nécessaires pour effectuer un rapatriement complet des dictionnaires des nœuds voisins et en calculer le consensus, notre algorithme implique un coût de communication beaucoup plus faible, tout en autorisant un fonctionnement asynchrone. L'algorithme proposé est donc particulièrement adapté au calcul de dictionnaires sur de très grands réseaux.

4.3. Système d'indexation complet

Le système d'indexation proposé est évalué sur la base d'images Holidays, constitué de 1 491 images répartis en 500 requêtes (*cf.* figure 7). L'objectif associé à cette base est la recherche de quasi-copies (*i.e.* une même scène dans des conditions de prise de vue différente : angle, éclairage, occultations, etc...). Chaque image est décrite par un sac de descripteurs SIFT (quelques milliers par image) de dimension $D = 128$. Nous avons ainsi choisi d'utiliser la méthode d'agrégation VLAD (Jégou *et al.*, 2012) pour calculer les signatures. Elle consiste à quantifier l'ensemble \mathbf{X} de descripteurs d'une image sur le dictionnaire local $\{\mu_1^{(i)}, \dots, \mu_K^{(i)}\}$ puis accumuler, pour chaque cellule obtenue C_k , les déviations entre les descripteurs quantifiés dans C_k et le mot visuel $\mu_k^{(i)}$. La signature VLAD d'une image est donc un vecteur \mathbf{v} de dimension KD tel que

$$\mathbf{v} = \mathbf{v}_1 \dots \mathbf{v}_K \quad \text{avec} \quad \mathbf{v}_k = \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \mu_k^{(i)}). \quad (16)$$

Cette signature est connue pour ne nécessiter que des dictionnaires de petite taille (de 64 à 1 024 cellules). Les performances de nos dictionnaires sont évaluées en termes de mAP (précision moyenne) sur l'ensemble de la base Holidays, par rapport à un dictionnaire de référence de 64 cellules calculé de manière centralisée avec l'algorithme témoin, qui réalise une mAP d'environ 55 %.

Les images de la base Holidays ont été dispersées sur les nœuds de manière hétérogène (toutes les variantes d'une même instance étant hébergées sur un même nœud). Nous avons évalué la capacité du système à construire des dictionnaires locaux cohérents ainsi que ses performances d'indexation en termes de précision moyenne de recherche (mAP) sur la totalité de la base Holidays.

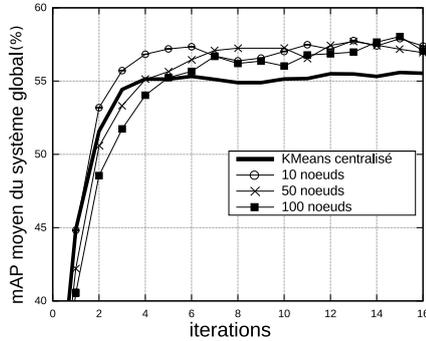


Figure 8. Le dictionnaire commun construit par notre système fournit des résultats de recherche (mAP) équivalents voire meilleurs qu'une solution centralisée

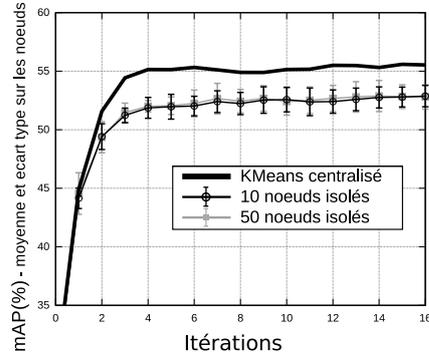


Figure 9. Des dictionnaires locaux entraînés isolément sont moins performants qu'une approche centralisée

	K = 4	K = 8	K = 16	K = 32	K = 64
<i>référence</i>	42,7	46,5	49,8	53,3	55,1
10 nœuds	44,2	47,9	51,6	54,2	57,0
50 nœuds	42,7	45,9	50,9	55,7	57,2
100 nœuds	44,7	47,3	51,8	53,4	56,0

Figure 10. Performances (% mAP) du système pour différentes tailles de dictionnaire K et nombre de nœuds N

Les dictionnaires locaux offrent des résultats d'indexation équivalents voire supérieurs à ceux d'un dictionnaire de référence calculé par l'algorithme centralisé, tel que le montre la figure 8. De telles performances signifient qu'un nœud qui ne dispose localement que d'informations parcellaires sur les contenus manipulés est capable de modéliser l'ensemble des données grâce à notre méthode décentralisée d'apprentissage de dictionnaires. Il modélise notamment des concepts visuels qu'il n'héberge pas localement, là où des systèmes centralisés entraînés isolément sur chaque nœud montrent une mauvaise généralisation comme l'atteste la figure 9.

Les expériences menées pour différents nombres de nœuds M et tailles de dictionnaire K , dont les résultats sont rassemblés en figure 10, montrent que le système décentralisé maintient des performances supérieures à la méthode de référence et présente une mAP très correcte pour de petits dictionnaires.

Dans le but d'optimiser les coûts de communication, la figure 11 montre que les dictionnaires produits en utilisant de faibles valeurs de M obtiennent une mAP aussi élevée que lorsque M est grand. Les simulations menées sur des réseaux de 50 nœuds

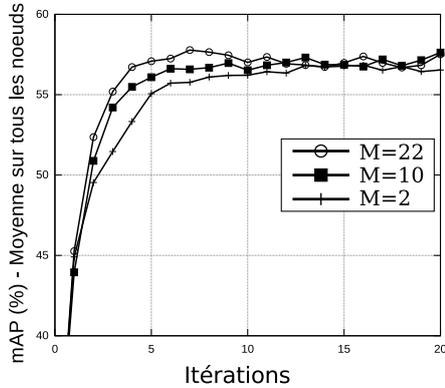


Figure 11. Influence de la réduction des coûts de communication sur la précision moyenne

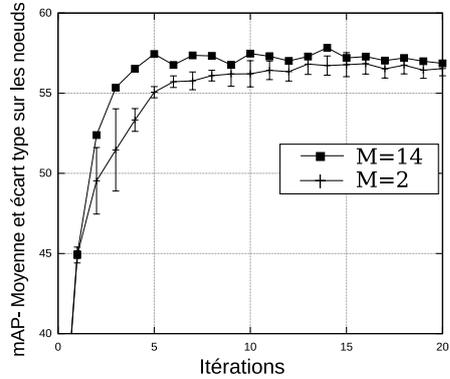


Figure 12. Influence de la réduction des coûts de communication sur l'écart type de la précision moyenne

avec $K = 64$ révèlent que bien qu'un nombre plus important d'itérations soit nécessaire pour atteindre les meilleurs résultats, les coûts totaux de communication sont fortement réduits. Le système global maintient malgré tout des performances de recherche asymptotiques équivalentes, même lorsque chaque nœud n'envoie que 2 messages par itération. Notons cependant que la légère dégradation de la cohérence des dictionnaires naturellement provoquée par la réduction de M se traduit par une variance plus élevée des résultats de recherche entre nœuds, visible dans la figure 12.

5. Conclusion

Dans cet article, nous proposons un système complet d'indexation multimédia décentralisé qui permet l'indexation conjointe de multiples collections de documents multimédias répartis sur de larges réseaux (des milliers de nœuds ou plus). Chaque machine est capable de produire des signatures pour ses propres documents qui offrent une précision d'indexation équivalente aux méthodes centralisées usuelles, via un processus local qui n'implique qu'une communication réduite et asynchrone avec les machines voisines dans le réseau.

Le système proposé permet ainsi de traiter des bases multimédias composées de milliards de documents, sous réserve que ces documents soient répartis sur un nombre suffisant d'unités de traitement, connectées via un réseau ou un bus mémoire.

Notre système utilise un algorithme d'apprentissage de dictionnaires locaux décentralisé et asynchrone qui s'appuie sur des protocoles Gossip pour produire des dictionnaires cohérents entre nœuds, et qui intègre une procédure décentralisée de *codeword-shifting* pour obtenir des dictionnaires de meilleure qualité. Appliqué à des bases d'images réparties, ses performances se sont révélées équivalentes à un système centralisé entraîné sur l'ensemble des données du réseau. Le paramétrage de l'algo-

rithme est facilité par une loi empirique pour déterminer le nombre de messages à échanger entre nœuds compte tenu de la taille du réseau considéré.

Les méthodes d'indexation centralisées les plus compétitives font appel à des techniques additionnelles de compression (PCA), de normalisation, etc... qui requièrent elles-aussi une agrégation sur l'ensemble des données. Leur décentralisation fait l'objet de nos travaux actuels.

Bibliographie

- Ailon N., Jaiswal R., Monteleoni C. (2009). Streaming k-means approximation. *Advances in Neural Information Processing Systems*, vol. 22, p. 10–18.
- Bandyopadhyay S., Giannella C., Maulik U., Kargupta H., Liu K., Datta S. (2006). Clustering distributed data streams in peer-to-peer environments. *Inf. Sci.*, vol. 176, n° 14, p. 1952–1985.
- Bay H., Ess A., Tuytelaars T., Gool L. V. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding*, vol. 110, n° 3, p. 346–359.
- Bénézit F., Blondel V., Thiran P., Tsitsiklis J., Vetterli M. (2010). Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *Information theory proceedings (isit), 2010 IEEE international symposium on*, p. 1753–1757.
- Boyd S., Ghosh A., Prabhakar B., Shah D. (2006). Randomized gossip algorithms. *Information Theory, IEEE Transactions on*, vol. 52, n° 6, p. 2508–2530.
- Datta S., Giannella C., Kargupta H. (2006). K-means clustering over a large, dynamic network.
- Datta S., Giannella C., Kargupta H. (2009, October). Approximate distributed k-means clustering over a peer-to-peer network. *IEEE Trans. on Knowl. and Data Eng.*, vol. 21, n° 10, p. 1372–1388. Retrieved from <http://dx.doi.org/10.1109/TKDE.2008.222>
- Demers A., Greene D., Houser C., Irish W. (1988). Epidemic algorithms for replicated database maintenance. *SIGOPS Oper. Syst. Rev.*, vol. 22, n° 1, p. 8–32. Retrieved from <http://doi.acm.org/10.1145/43921.43922>
- Dempster A. P., Laird N. M., Rubin D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, p. 1–38.
- Durut M., Patra B., Rossi F. (2012). A discussion on parallelization schemes for stochastic vector quantization algorithms. In *20-th european symposium on artificial neural networks, computational intelligence and machine learning (esann 2012)*, p. 477–482.
- Eisenhardt M., Muller W., Henrich A. (2003). Classifying documents by distributed p2p clustering. *GI Jahrestagung*, p. 286–291.
- Fatta G. D., Blasa F., Cafiero S., Fortino G. (2011, December). Epidemic k-means clustering. In *2011 IEEE 11th international conference on data mining workshops*, p. 151–158. IEEE. Retrieved from <http://centaur.reading.ac.uk/27053/> (Print ISBN: 9781467300056 Issue Date: 11-11 Dec. 2011 On page(s): 151 - 158)

- Fatta G. D., Blasa F., Cafiero S., Fortino G. (2012, September). Fault tolerant decentralised k-means clustering for asynchronous large-scale networks. *Journal of Parallel and Distributed Computing*. Retrieved from <http://centaur.reading.ac.uk/29421/>
- Gemert J. C. van, Veenman C. J., Smeulders A. W. M., Geusebroek J. M. (2010). Visual word ambiguity. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, n° 7, p. 1271–1283. Retrieved from <http://www.science.uva.nl/research/publications/2010/vanGemertTPAMI2010>
- Iutzeler F., Ciblat P., Hachem W. (2013). Analysis of sum-weight-like algorithms for averaging in wireless sensor networks. *Signal Processing, IEEE Transactions on*, vol. 61, n° 11, p. 2802–2814.
- Januzaj E., Kriegel H.-P., Pfeifle M. (2004). Dbdc: Density based distributed clustering. In *Advances in database technology-edbt 2004*, p. 88–105. Springer.
- Jégou H., Douze M., Schmid C. (2008, oct). Hamming embedding and weak geometric consistency for large scale image search. In A. Z. David Forsyth Philip Torr (Ed.), *European conference on computer vision*, vol. I, p. 304–317. Springer. Retrieved from <http://lear.inrialpes.fr/pubs/2008/JDS08>
- Jégou H., Douze M., Schmid C., Pérez P. (2010, June). Aggregating local descriptors into a compact image representation. In *Ieee international conference on computer vision and pattern recognition*, p. 3304–3311.
- Jégou H., Perronnin F., Douze M., Sánchez J., Pérez P., Schmid C. (2012). Aggregating local image descriptors into compact codes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 1, p. 3304-3311. Retrieved from [http://hal.inria.fr/inria-00633013/en/\(QUAERO\)](http://hal.inria.fr/inria-00633013/en/(QUAERO))
- Jelasity M., Kowalczyk W., Van Steen M. (2003). *Newscast computing*. Tech. Rep.. Technical Report IR-CS-006, Vrije Universiteit Amsterdam, Department of Computer Science, Amsterdam, The Netherlands.
- Kempe D., Dobra A., Gehrke J. (2003). Gossip-based computation of aggregate information. In *Proceedings of the 44th annual ieee symposium on foundations of computer science*, p. 482–. Washington, DC, USA, IEEE Computer Society. Retrieved from <http://dl.acm.org/citation.cfm?id=946243.946317>
- Kohonen T. (1990). The self-organizing map. *Proceedings of the IEEE*, vol. 78, n° 9, p. 1464–1480.
- Kowalczyk W., Vlassis N. A. (2004). Newscast em. In *Advances in neural information processing systems*, p. 713–720.
- Linde Y., Buzo A., Gray R. (1980). An algorithm for vector quantizer design. *IEEE Transaction on Communication*, vol. 28, p. 84–94.
- Lloyd S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, vol. 28, p. 129–137.
- Lowe D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, vol. 2, n° 60, p. 91–110.
- Lu T.-C., Chang C.-Y. (2010). A survey of vq codebook generation. *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, n° 3, p. 190–203.

- Marée R., Denis P., Wehenkel L., Geurts P. (2010). Incremental indexing and distributed image search using shared randomized vocabularies. In *Proceedings of the international conference on multimedia information retrieval*, p. 91–100.
- Müller W. T., Eisenhardt M., Henrich A. (2003, December). Efficient content-based P2P image retrieval using peer content descriptions. In *Internet imaging v. edited by santini, simone; schettini, raimondo. proceedings of the spie, volume 5304, pp. 57-68 (2003).*, p. 57–68.
- Nikseresht A., Gelgon M. (2008). Gossip-based computation of a gaussian mixture model for distributed multimedia indexing. *Multimedia, IEEE Transactions on*, vol. 10, n° 3, p. 385–392.
- Patanè G., Russo M. (2001, November). The enhanced LBG algorithm. *Neural Networks*, vol. 14, n° 9, p. 1219–1237.
- Patanè G., Russo M. (2000). Elbg implementation. *International Journal of Knowledge based Intelligent Engineering Systems*, vol. 2, p. 2–4.
- Perronnin F., Dance C. R. (2007, June). Fisher kernels on visual vocabularies for image categorization. In *Ieee international conference on computer vision and pattern recognition*, p. 1–8.
- Picard D., Gosselin P.-H. (2011, September). Improving image similarity with vectors of locally aggregated tensors. In *Ieee international conference on image processing*. Brussels, Belgique. Retrieved from <http://hal.archives-ouvertes.fr/hal-00591993/en/>
- Picard D., Revel A., Cord M. (2012). An application of swarm intelligence to distributed image retrieval. *Information Sciences*, vol. 192, p. 71–81.
- Shah D. (2009). *Gossip algorithms*. Now Publishers Inc.
- Shindler M., Meyerson A., Wong A. (2011). Fast and accurate k-means for large datasets. In *Proc. of twenty-fifth annual conference on neural information processing systems (nips)*.
- Sivic J., Zisserman A. (2003). Video google: A text retrieval approach to object matching in videos. In *Ieee international conference on computer vision*, vol. 2, p. 1470–1477.
- Wang J., Yang J., Yu K., Lv F., Huang T., Gong Y. (2010). Locality-constrained linear coding for image classification. In *Ieee international conference on computer vision and pattern recognition*, p. 3360–3367.
- Yang J., Yu K., Gong Y., Huang T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, p. 1794–1801.