Apprentissage de distance pour l'annotation d'images par plus proches voisins

Matthieu Guillaumin, Jakob Verbeek, Cordelia Schmid

Apprentissage de distance pour l'annotation d'images par plus proches voisins

Matthieu Guillaumin Thomas Mensink Jakob Verbeek Cordelia Schmid

Équipe-projet LEAR, INRIA Grenoble et Laboratoire Jean Kuntzmann

655, avenue de l'Europe, Montbonnot, F-38334 Saint-Ismier Cedex
Prénom.Nom@inria.fr

Résumé

L'annotation automatique d'image est un problème ouvert important pour la vision par ordinateur. Pour cette tâche nous proposons TagProp, un modèle par plus proche voisins pondérés. Celui-ci est entraîné de manière discriminative et exploite des images d'apprentissage pour prédire les labels des images de test. Les poids sont calculés à partir du rang ou de la distance entre l'image et son voisin. TagProp permet l'optimisation de la distance qui définit les voisinages en maximisant la log-vraisemblance des prédictions de l'ensemble d'apprentissage. Ainsi, nous pouvons régler de manière optimale la combinaison de plusieurs similarités visuelles qui vont des histogrammes globaux de couleur aux descriptions locales de forme. Nous proposons également de moduler spécifiquement chaque mot pour augmenter le rappel des mots rares. Nous comparons les performances des différentes variantes de notre modèle à l'état de l'art sur trois bases d'images. Sur les cinq mesures considérées, TagProp améliore significativement l'état de l'art.

Mots Clef

Annotation automatique d'image, recherche d'image par mot clef, apprentissage de distance, plus proches voisins.

Abstract

Image annotation is an important open problem in computer vision. For this task we propose TagProp, a weighted nearest neighbor model, discriminatively trained to exploit labeled training images for predicting tags of test images. Neighbor weights are based on neighbor rank or distance. TagProp can learn the metric that defines neighbors by maximizing the log-likelihood of the tag predictions in the training set. Hence, we can optimally combine several image similarity metrics that cover different aspects of image content, from global color histograms to local shape descriptors. We also propose to specifically modulate each word to boost the recall of rare words. We compare the performance of the different variants of our model to existing work for three challenging data sets. On all measures, TagProp notably improves over the state-of-the-art.

Keywords

Image auto-annotation, keyword-based image retrieval, metric learning, nearest neighbor model.

1 Introduction

L'annotation automatique d'image est un sujet effectif de recherche [7, 15, 16, 18]. Le but est de développer des méthodes qui peuvent prédire, pour une nouvelle image, les mots clef pertinents parmi un vocabulaire d'annotation. Ces prédictions de mots clef peuvent ensuite être utilisées soit pour proposer des étiquettes pour l'image, soit proposer des images à une requête par mot clef (ou combinaison de mots clef). De telles méthodes deviennent de plus en plus importantes étant donné l'augmentation des archives de contenus visuels disponibles, par exemple les sites internet de partage de photos ou vidéos, ou les applications de gestion d'images personnelles. Ces bases de grande taille motivent le développement de méthodes automatiques d'indexation, d'annotation et de requête. La quantité d'images possédant des annotations plus ou moins structurées allant également croissant, des techniques d'apprentissage machine peuvent en bénéficier en estimant de manière plus robuste des modèles de prédiction de labels.

Bien que le problème général soit difficile, la communauté a accompli des progrès significatifs grâce à l'apparition de bases standards d'images annotées. Dans la section suivante, nous détaillons les travaux existants qui se rapprochent le plus du nôtre. Les inconvénients principaux de ces méthodes sont de deux natures. Premièrement, les modèles proposés sont souvent estimés pour maximiser la vraisemblance générative des caractéristiques visuelles et des labels, ce qui n'est pas nécessairement optimal pour la tâche de prédiction d'annotation. Deuxièmement, beaucoup de modèles paramétriques ne sont pas assez riches pour capturer les dépendances subtiles entre le contenu visuel des images et les annotations. Des méthodes non paramétriques telles que les méthodes par plus proches voisins se sont avérées assez fructueuses pour cette tâche [5, 11, 13, 17, 22, 27]. Ceci est principalement dû à la grande « capacité » de ces classifieurs : ils s'adaptent

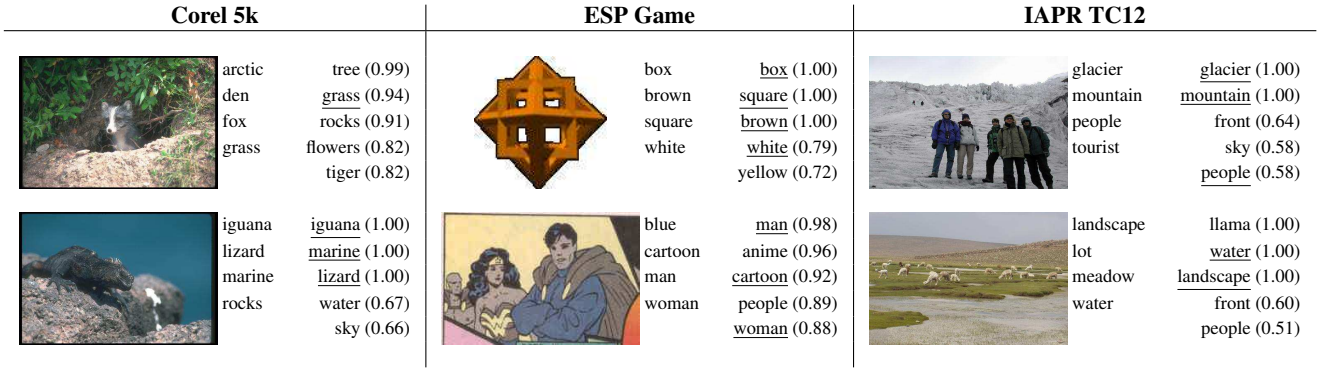

FIG. 1 – Exemples d'images test provenant des trois bases utilisées. A côté de chaque image sont données la vérité terrain (gauche) et les cinq labels les plus pertinents (soulignés lorsqu'ils sont corrects) selon la variante σML notre modèle TagProp (avec $K = 200$ voisins). Il est remarquable que les trois bases d'images sont très différentes et que la vérité terrain ne contient pas tous les labels pertinents (par exemple, « water » pour l'image en bas à gauche), et contient parfois des labels discutables (par exemple, « lot » pour l'image en bas à droite).

de manière flexible aux motifs des données quand celles-ci deviennent plus nombreuses. Toutefois, les méthodes existantes par plus proches voisins pour l'annotation d'image n'optimisent pas la distance qui définit les plus proches voisins pour maximiser les performances de prédiction. Elles utilisent soit une métrique fixe [5, 27], soit une combinaison ad hoc de plusieurs distances [17], malgré de nombreux travaux récents montrant les bénéfices que l'apprentissage de cette distance peut apporter à d'autres tâches de la vision par ordinateur comme la classification [12], la recherche d'images [10], ou l'identification visuelle [9].

Dans cet article nous présentons TagProp, pour Propagation de Tags, une approche originale par plus proches voisins qui prédit des labels en considérant une combinaison pondérée des présences et absences des labels dans le voisinage des images. Nos contributions sont les suivantes. Premièrement, les poids des voisins sont déterminés en fonction soit du rang de voisin, soit de la distance au voisin, et réglés automatiquement pour maximiser la vraisemblance des labels des images d'apprentissage. Avec les poids basés sur le rang, le k-ième voisin reçoit toujours le même poids, tandis que les poids basés sur la distance décroissent exponentiellement avec celle-ci. Notre modèle de prédiction de labels est conceptuellement simple, mais s'avère plus performant que l'état de l'art actuel sur les mêmes jeux de données. Ensuite, et contrairement aux travaux existants, notre modèle permet directement l'apprentissage de la métrique. Cela nous permet d'optimiser, par exemple, une distance de Mahalanobis entre caractéristiques visuelles (ou, de manière moins coûteuse, une combinaison de plusieurs distances) pour définir les poids des voisins pour la prédiction de labels. Enfin, TagProp incorpore des modèles de discrimination logistique spécifique à chaque mot. Ces modèles utilisent en entrée les prédictions du modèle de plus proches voisins et sont capables, à l'aide de seulement deux paramètres, de moduler les probabilités de présence des mots rares ou fréquents. Cela permet une augmentation significative du nombre de mots effectivement assignés à au moins une image.

Pour évaluer nos modèles et nous comparer aux travaux existants, nous utilisons trois bases d'images, Corel 5k, IAPR TC-12 et ESP Game, ainsi que plusieurs mesures usuelles de performance. La Figure 1 montre plusieurs exemples d'images avec leurs labels et les prédictions de notre modèle. Sur toutes les bases et toutes les mesures, nous obtenons des performances significativement supérieures aux travaux précédemment publiés.

L'article est composé comme suit. Dans la Section 2, nous donnons un aperçu des travaux existants. Ensuite, en Section 3, nous présentons nos modèles et les méthodes d'estimation de leurs paramètres. La Section 4 présente les trois bases d'images, les mesures d'évaluation ainsi que les représentations des images que nous utilisons dans nos expériences. Les résultats expérimentaux sont donnés dans la Section 5. Enfin, en Section 6 nous présentons nos conclusions et les directions futures de recherche.

2 État de l'art

Schématiquement, nous distinguons quatre groupes de méthodes pour l'annotation d'image : les modèles à thèmes, les modèles de mélanges, les modèles discriminatifs et les modèles locaux, dont ceux par «plus proches voisins».

Le premier groupe de méthodes est basé sur les modèles à thèmes tels que l'allocation de Dirichlet latente, l'analyse sémantique latente probabiliste et les processus de Dirichlet hierarchiques, voir par exemple [1, 20, 25]. Ces méthodes modélisent les images annotées comme des échantillons d'un mélange spécifique de thèmes, et chaque thème est une distribution sur les caractéristiques visuelles des images et les mots clef d'annotation. Pour chaque image, les proportions du mélange de thèmes doivent être estimées. Et pour chaque thème, la distribution des données (le plus souvent multinomiale pour les mots et Gaussienne pour les caractéristiques visuelles de différentes régions de l'image) doit l'être aussi. Les méthodes inspirées de la traduction par ordinateur [4] peuvent aussi être vues comme des modèles à thèmes, un pour chaque lexème visuel.

Une seconde famille de méthodes utilise les modèles de

mélanges pour définir une distribution jointe sur l'espace des caractéristiques visuelles (Gaussiennes) et des annotations (multinomiales ou Bernoulli). Pour annoter une nouvelle image, ces modèles calculent la probabilité conditionnelle des annotations étant données les caractéristiques visuelles en normalisant la vraisemblance jointe. Le nombre de composantes de caractéristiques visuelles pour le mélange est soit fixé [2], soit défini par des images d'apprentissage [5, 11, 13]. Ces modèles peuvent être considérés comme des estimateurs non-paramétriques de densité sur l'espace des co-occurrences d'image et de mots clef.

Ces deux familles de modèles génératifs peuvent être critiquées car elles maximisent la vraisemblance générative des données, ce qui n'est peut-être pas optimal pour la tâche de prédiction des labels. C'est pourquoi des modèles discriminatifs ont aussi été proposés [3, 7, 10]. Ces méthodes apprennent des classifieurs pour chaque mot clef, et utilisent ceux-ci pour décider si une image appartient ou non à la classe des images qui possèdent ce mot clef. Différentes méthodes d'apprentissage ont été proposées, dont les machines à vecteurs supports, l'apprentissage d'instances multiples et les machines à point de Bayes. Le problème de la recherche d'images par requêtes complexes, composées de plusieurs mots, est abordé dans [7].

Étant donnée l'augmentation de la quantité de données annotées disponibles, les méthodes locales deviennent plus intéressantes en tant qu'alternatives simples mais puissantes aux modèles paramétriques. Parmi ces techniques, on peut inclure la diffusion des labels sur un graphe de similarité [16, 22], ou l'apprentissage de modèles discriminatifs dans les voisinages d'images test [27]. Une approche ad hoc simple par plus proches voisins a été récemment proposée [17], et s'avère définir l'état de l'art en termes de performance. Les voisins y sont déterminés à l'aide d'une combinaison de plusieurs distances correspondant à différentes caractéristiques visuelles. Les auteurs ont également essayé d'apprendre cette combinaison de distances via un classifieur binaire qui tente de séparer les paires d'images possédant plusieurs mots clef en commun et celles n'en possédant aucun. Cette tentative s'est avérée infructueuse par rapport à la combinaison ad hoc considérée.

3 Modèles de prédiction de label

Notre but est de prédire la pertinence de mots clef pour des images. Cela permettra ensuite soit de proposer des mots clef pour annoter les images, soit de rechercher les images pertinentes pour des mots clef. Notre proposition est basée sur un modèle de plus proches voisins pondérés, inspiré de travaux récents et fructueux [5, 11, 13, 17] qui propagent les annotations de l'ensemble d'apprentissage aux nouvelles images. Nos modèles sont entraînés de manière discriminative plutôt qu'en utilisant un ensemble de validation [5] ou une technique ad hoc [17]. Nous supposons que des similarités ou distances visuelles entre images sont disponibles, en nous affranchissant temporairement de leurs définitions précises.

3.1 Prédiction par voisins pondérés

Pour modéliser les annotations des images, nous utilisons un modèle de Bernoulli pour chaque mot clef. Ce choix est naturel car les mots clef, à l'inverse du texte naturel où la fréquence d'apparition des mots est importante, sont soit présents, soit absents. Les dépendances entre les mots clef dans la base d'apprentissage ne sont pas modélisées explicitement, mais notre modèle les exploite implicitement.

Nous dénotons par $y_{im} \in \{-1, +1\}$ l'absence ou la présence du mot clef m pour l'image i, représentant ainsi les annotations des images. La prédiction $p(y_{im}=+1)$ de présence de ce label pour l'image i est une somme pondérée sur les images d'apprentissage, que l'on indexe par j :

$$p(y_{im} = +1) = \sum_j \pi_{ij} p(y_{im} = +1|j), \tag{1}$$

$$p(y_{im} = +1|j) = \begin{cases} 1-\epsilon & \text{si } y_{jm} = +1, \\ \epsilon & \text{sinon,} \end{cases} \tag{2}$$

où π_{ij} denote le poids de l'image j pour la prédiction des labels de l'image i. Nous imposons $\pi_{ij} \geq 0$ et $\sum_j \pi_{ij} = 1$. Nous utilisons ϵ pour éviter de prédire une probabilité nulle, et en pratique réglons $\epsilon = 10^{-5}$. Pour estimer les paramètres qui contrôlent les poids π_{ij}, nous maximisons la log-vraisemblance $\mathcal{L}$ des prédictions pour les images d'apprentissage, en mettant à zéro les poids des images pour leur propre prédiction, c'est-à-dire $\forall i, \pi_{ii} = 0$

$$\mathcal{L} = \sum_{i,m} c_{im} \ln p(y_{im}), \tag{3}$$

où c_{im} est un coût qui tient compte du déséquilibre entre présence et absence d'un mot clef. En effet, en pratique, il y a bien plus d'absences de label que de présences. De plus, si la plupart des labels présents sont pertinents, de nombreux labels pertinents sont omis dans l'annotation. Nous choisissons $c_{im} = 1/n^+$ lorsque $y_{im} = +1$, où $n^+ = \sum_{i,m} \max(y_{im}, 0)$ est le nombre total de labels positifs, et de la même manière $c_{im} = 1/n^-$ quand $y_{im} = -1$, où $n^- = \sum_{i,m} \max(-y_{im}, 0)$.

Poids basés sur le rang. Si l'image j est le k-ième voisin de l'image i, nous appelons k son rang. Les poids π_{ij} peuvent alors être définis constants à rang donné : $\pi_{ij} = \gamma_k$. La log-vraisemblance des données (3) est concave vis à vis des paramètres γ_k et ceux-cis peuvent être estimés par un algorithme de gradient projeté pour assurer la contrainte $\gamma_k \geq 0$. La dérivée de l'équation (3) par rapport à γ_k vaut

$$\frac{\partial \mathcal{L}}{\partial \gamma_k} = \sum_{i,m} \frac{c_{im} p(y_{im}|n_{ik})}{p(y_{im})}, \tag{4}$$

où n_{ik} dénote l'index du k-ième voisin de i. Le nombre de paramètres est K, la taille des voisinages. Nous appelons cette variante RK, pour « Rank-based ».

Poids basés sur la distance. L'autre possibilité est de définir les poids directement comme une fonction de la distance, plutôt que le rang. L'avantage est que cette fonction

peut être continue, ce qui est crutial si la distance est amenée à changer durant l'apprentissage. Les π_{ij} deviennent

$$\pi_{ij} = \frac{\exp(-d_{\boldsymbol{\theta}}(i,j))}{\sum_{j'} \exp(-d_{\boldsymbol{\theta}}(i,j'))}, \quad (5)$$

où $d_{\boldsymbol{\theta}}$ est une distance paramétrée par $\boldsymbol{\theta}$, que nous souhaitons optimiser. On remarquera que la décroissance de π_{ij} avec la distance est choisie comme exponentielle. Les choix pour $d_{\boldsymbol{\theta}}$ incluent les distances de Mahalanobis $d_{\mathbf{M}}$ paramétrés par une matrice symétrique définie positive $\mathbf{M}$, et les distances $d_{\mathbf{w}}(i,j) = \mathbf{w}^\top \mathbf{d}_{ij}$ où $\mathbf{d}_{ij}$ est un vecteur de distances de base entre les images i et j, et $\mathbf{w}$ contient des coefficients positifs pour la combination linéaire. Le nombre de paramètres dans ce cas est égal au nombre de distances de base qui sont combinées. Dans la suite, nous nous concentrons sur ce cas particulier. Lorsqu'une seule distance est utilisée, nous nommons la variante SD, et $\mathbf{w}$ est alors un scalaire qui contrôle la vitesse de décroissance des poids avec la distance et il s'agit du seul paramètre du modèle. Quand plusieurs distances sont utilisées, nous nommons la variante ML, pour « Metric Learning ».
La log-vraisemblance du modèle est maximisée sous la contrainte de positivité des éléments de $\mathbf{w}$. Avec la nouvelle définition des poids, le gradient de la log-vraisemblance, donnée en Eq. (3), par rapport à $\mathbf{w}$ vaut

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \sum_{i,j} M_i(\pi_{ij} - \rho_{ij})\mathbf{d}_{ij}, \quad (6)$$

où $M_i = \sum_m c_{im}$ et ρ_{ij} dénote la moyenne, pondérée sur les mots, des probabilités a posteriori du voisin j pour l'image i étant donnée son annotation :

$$\rho_{ij} = \sum_m \frac{c_{im}}{M_i} p(j|y_{im}). \quad (7)$$

Pour réduire la complexité de calcul lors de l'apprentissage du modèle, nous ne calculons pas les π_{ij} et ρ_{ij} pour toutes les paires d'images. Plutôt, pour chaque image i, nous calculons ces valeurs pour K images, avec K grand, et supposons que les π_{ij} et ρ_{ij} restants valent zéro. Précisément, nous gardons l'ensemble des K images qui est l'union de voisinages de taille k selon chacune des distances, et k est pris le plus grand possible. De cette manière, nous augmentons les chances de conserver toutes les paires d'images qui auront un π_{ij} grand, et ce, quelle que soit la combinaison de distances $\mathbf{w}$ apprise. De plus, une fois ces voisinages déterminés, la complexité de l'algorithme est linéaire avec le nombre d'images d'apprentissage.
Enfin, notons le lien entre notre modèle et l'approche d'apprentissage de distance de [6]. La métrique y est apprise de telle manière que les poids π_{ij} tels que définis par Eq. (5) soient les plus proches possibles, au sens de la divergence de Kullbach-Leibler (KL), de valeurs ρ_{ij} fixées comme objectif. Ces valeurs sont mises à zéro pour les paires de classes différentes et à un terme constant non nul pour les paires de la même classe. En écrivant l'optimisation de notre modèle via un algorithme de type EM, nous obtenons pour l'étape M une formule qui correspond précisément à la divergence KL entre les ρ_{ij} obtenus à l'étape E et les π_{ij}. Pour des ρ_{ij} fixés, cette divergence KL est convexe en $\mathbf{w}$. Selon sa paramétrisation, la log-vraisemblance peut ne pas être concave, auquel cas seul un maximum local est atteint.

3.2 Discrimination logistique par mot clef

Les approches par plus proches voisins ont cependant tendance à obtenir des scores de rappel relativement bas, ce qui peut s'expliquer de la manière suivante. Pour se voir attribuer une forte probabilité de présence d'un mot clef, il faut que ce mot clef soit présent pour la majorité des voisins qui possèdent un poids élevé. Cette situation est peu probable pour les mots clef rares. Même avec la présence de quelques occurrences d'un mot clef rare dans le voisinage, une probabilité faible de pertinence lui sera associée. Pour surmonter ce problème potentiel, nous proposons d'utiliser des modèles de discrimination logistique spécifiques à chaque mot, qui pourront augmenter ou diminuer les probabilités de présence des mots rares ou fréquents. En utilisant les prédictions x_{im} du modèle de plus proche voisin précédent, nous redéfinissons $p(y_{im} = +1)$ par

$$x_{im} = \sum_j \pi_{ij} y_{jm}, \quad (8)$$

$$p(y_{im} = +1) = \sigma(\alpha_m x_{im} + \beta_m), \quad (9)$$

où $\sigma(z) = (1 + \exp(-z))^{-1}$. Ces modèles ajoutent deux paramètres par mots à estimer. Les variantes ainsi modulées sont nommées σRK, σSD et σML, respectivement.
Pour des π_{ij} fixes, le modèle est un modèle de discrimination logistique, et la log-vraisemblance est concave en $\{\alpha_m, \beta_m\}$, qui peuvent être optimisés séparément. Le gradient de la log-vraisemblance des annotations d'apprentissage selon les paramètres $\boldsymbol{\theta}$ qui contrôlent les poids devient

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \sum_{i,m} c_{im}\alpha_m p(-y_{im}) y_{im} \frac{\partial x_{im}}{\partial \boldsymbol{\theta}}, \quad (10)$$

et, selon que les poids soient basés sur le rang ou la distance, respectivement

$$\frac{\partial x_{im}}{\partial \gamma_k} = y_{n_{ik}m}, \quad (11)$$

$$\frac{\partial x_{im}}{\partial \mathbf{w}} = \sum_j \pi_{ij} (x_{im} - y_{jm}) \mathbf{d}_{ij}. \quad (12)$$

En pratique, nous estimons les paramètres $\boldsymbol{\theta}$ et $\{\alpha_m, \beta_m\}$ en alternance. Nous observons une convergence rapide, typiquement après trois alternances de maximisation.

4 Bases d'images et évaluations

Dans cette section, nous présentons d'abord les bases d'images utilisées dans nos expérimentations, puis les différentes caractéristiques visuelles qui nous fournissent une collection de distances entre images, et enfin les mesures d'évaluation de performance pour l'annotation et la recherche d'images.

	Corel 5k	ESP Game	IAPR TC12
Vocabulaire	260	268	291
Nb. d'images	4493	18689	17665
Mots par image	3,4 (5)	4,7 (15)	5,7 (23)
Images par mot	58,6 (1004)	362,7 (4553)	347,7 (4999)

TAB. 1 – Statistiques des ensembles d'apprentissage des trois bases d'images. Pour les rapports mots-images, elles sont données sous le format « moyenne (maximum) ». Les statistiques sont similaires pour les ensembles de test.

4.1 Bases d'images

Nous utilisons trois bases d'images disponibles publiquement et qui ont été utilisées dans des travaux récents, ce qui permet une comparaison directe. La Table 1 résume certaines statistiques de ces bases, et des exemples sont présentés dans la Figure 1.

Corel 5k. Cette base d'images a été initialement utilisée dans [4]. Depuis, elle est devenue un étalon important pour la recherche d'image par mots clef et l'annotation d'image. Elle contient 5000 images annotées manuellement avec 1 à 5 mots. Le vocabulaire contient 260 mots. Un sous-ensemble de 499 images est utilisé comme test, et le reste pour l'apprentissage.

ESP Game. Cette base a été obtenue via un jeu en ligne où deux joueurs, qui ne peuvent communiquer par ailleurs, gagnent des points en s'accordant sur des mots décrivant les images qui leur sont proposées à l'affichage [24]. De cette manière, les joueurs sont encouragés à fournir aux images des labels importants et pertinents. Nous utilisons un sous-ensemble d'environ 20 000 images parmi les 60 000 disponibles, le même que celui utilisé par [17]. Cette base est très difficile, car elle contient une grande variété d'images : logos, dessins et photos personnelles.

IAPR TC12. Cet ensemble d'environ 20 000 images accompagnées de descriptions dans plusieurs langues a été initialement publiée pour la recherche documentaire multilingue [8]. Elle peut être transformée en un format comparable aux autres bases via l'extraction des noms communs en ayant recours à des techniques de traitement de langage naturel. Nous utilisons les mêmes annotations que [17].

4.2 Extraction des caratéristiques

Parmi les différents types de caractéristiques, nous utilisons deux types de descripteurs globaux : Gist [21], et des histogrammes de couleur formés de 16 cellules de quantification par canal, pour les espaces de couleur RVB, Lab et TSV. En tant que caractéristiques locales, nous utilisons des descripteurs SIFT ainsi qu'un descripteur robuste de teinte [23], tous les deux extraits sur une grille dense de points à plusieurs échelles, et sur des points d'intérêt de type Harris-Laplace. Chacune des caractéristiques locales est quantifiée en utilisant l'algorithme K-means sur les échantillons d'apprentissage. Les images sont ensuite représentées comme des histogrammes de «sacs de mots». Tous les descripteurs sauf Gist sont normalisés pour la norme 1 et aussi calculés selon une répartition spatiale [14]. L'image est divisée en trois régions horizontales et les histogrammes de chaque région sont concaténés pour former un nouveau descripteur global, qui encode ainsi de l'information sur l'agencement spatial de l'image. Pour limiter la taille des histogrammes, ici, nous n'utilisons que 12 cellules par canal pour les histogrammes de couleur. Cette division diffère sensiblement de l'approche par segmentation d'image utilisée dans certains travaux précédents.

Ainsi, nous obtenons 15 descriptions différentes, à savoir Gist, 6 histogrammes de couleur et 8 sacs de mots (2 caractéristiques $\times$ 2 descripteurs $\times$ 2 agencements). Nous choisissons les distances L2 pour Gist, L1 pour les histogrammes de couleur, et χ^2 pour les autres descriptions.

4.3 Mesures d'évaluation

Nous évaluons nos modèles selon plusieurs mesures usuelles de performance, utilisées dans les travaux précédents, qui évaluent les performances de recherche pour chaque mot, puis en calculent la moyenne.

Précision et rappel pour cinq mots par image.
Suivant [4], chaque image est annotée à l'aide des cinq mots les plus pertinents. Puis, la précision $\boldsymbol{P}$ et le rappel $\boldsymbol{R}$ sont moyennés sur les mots clef. **N+** denote le nombre de mots clef qui ont un rappel strictement positif. Ces mesures sont imparfaites car le nombre de mots clef pertinents selon la vérité terrain est souvent bien supérieur à cinq.

Précision à différents niveaux de rappel.
Comme [7], nous utilisons aussi les mesures **BEP** et **mAP**, qui opèrent à différents niveaux de rappel. **BEP** mesure pour chaque mot m la précision parmi les n_m images les plus pertinentes, où n_m est le nombre d'images annotées avec m dans la vérité terrain. **mAP** est obtenue en calculant pour chaque mot clef la moyenne des précisions mesurées au niveau de chaque image pertinente.

5 Résultats Expérimentaux

La Figure 1 montre des résultats qualitatifs. Dans cette section, nous présentons une évaluation quantitative de TagProp. Nous comparons avec l'état de l'art, d'abord en détaillant les résultats pour la base Corel 5k, puis, en Section 5.2, sur les bases IAPR TC12 et ESP Game. Enfin, nous présentons les résultats pour la recherche d'image par requête complexe dans la Section 5.3.

5.1 Résultats pour la base Corel 5k

Dans un premier jeu d'expérimentations, nous comparons les différentes variantes de TagProp aux résultats de [17], que l'on nomme JEC dans la suite. JEC utilise comme distance pour définir les voisinages une combinaison uniforme de plusieurs distances normalisées. Appliquée à nos 15 distances, nous lui donnons le nom JEC-15. Cette distance est également utilisée pour nos modèles à distance fixe, et pour le noyau Gaussien de la méthode σSVM-15, qui apprend un SVM ($C = 100$) par mot clef et module les scores de

	Résultats précédemment publiés									TagProp					
	CRM [13]	InfNet[19]	NPDE [26]	SML [2]	MBRM [5]	TGLM [16]	JEC [17]	JEC-15	σSVM-15	RK	σRK	SD	σSD	ML	σML
$\boldsymbol{P}$	0.16	0.17	0.18	0.23	0.24	0.25	0.27	**0.28**	0.23	0.28	0.26	0.30	0.28	0.31	**0.33**
$\boldsymbol{R}$	0.19	0.24	0.21	0.29	0.25	0.29	0.32	**0.33**	0.17	0.32	0.34	0.33	0.35	0.37	**0.42**
N+	107	112	114	137	122	131	139	**140**	79	136	143	136	145	146	**160**

TAB. 2 – Résultats, sur la base Corel 5k, en termes de $\boldsymbol{P}$, $\boldsymbol{R}$, et **N+** de TagProp (avec $K=200$), et d'une sélection de travaux déjà publiés. σSVM-15 et JEC-15 [17] dénotent l'utilisation de nos 15 distances. Pour nos variantes, RK et SD utilisent une combinaison uniforme des distances, ML apprend cette combinaison. σRK, σSD et σML sont leurs extensions modulées.

classification par des sigmoïdes dans l'esprit de l'Eq. (9).
En observant les résultats de la Table 2, nous pouvons faire les remarques suivantes. Premièrement, la méthode σSVM souffre d'un rappel faible. Ensuite, JEC-15 obtient des résultats très similaires à ceux publiés en [17]. Ainsi, toute différence ultérieure pourra être mise au crédit de la méthode de transfert de labels. Nos modèles à distance fixe pour definir les poids de voisins (soit par leur rang RK, soit directement SD), tout en étant moins ad hoc, ont des performances au moins aussi bonnes que JEC-15.
Les variantes avec apprentissage de distance (ML et en particulier σML) améliorent significativement les performances. Comparé à l'état de l'art sur les mêmes données, nous observons un gain de 5% en précision, 9% en rappel et 20 mots supplémentaires avec un rappel positif. Ces résultats montrent que le modèle de prédiction de labels par plus proches voisins bénéficie de l'apprentissage de la métrique. Il est intéressant d'observer que de précédentes tentatives avait échoué, cf [17]. La clef est, selon nous, dans l'optimisation directe de la distance dans le modèle de prédiction.
Dans la Figure 2, nous analysons en détail les bénéfices en termes de rappel des extensions sigmoïdales selon la fréquence des mots. Comme attendu, les améliorations sont plus grandes pour les mots rares et seuls les quelques mots les plus fréquents sont pénalisés, pour lesquels beaucoup d'images pertinentes sont malgré tout prédites.
Dans la Figure 3, nous montrons les performances en termes de $\boldsymbol{P}$, $\boldsymbol{R}$, **BEP** et **mAP** pour les variantes basées sur les distances, en fonction de la taille des voisinages. De manière systématique, pour toutes les tailles de voisinages, avec ou sans modulation, la combinaison apprise de distances est meilleure que la combinaison uniforme des mêmes distances. Nous observons également que la variante σML a un impact fort sur le rappel $\boldsymbol{R}$, mais qu'elle améliore aussi les performances selon les autres mesures. Les performances croissent avec la taille du voisinage considéré, au moins jusqu'à 100 voisins. Ceci est particulièrement vrai pour les variantes ML, parce que la modification de la distance remanie les plus proches voisins. Les voisinages doivent être pris assez larges au départ pour assurer la présence des voisins utiles a posteriori.
A partir de ces premières expériences, les variantes à poids basés sur les distances semblent les plus prometteuses, et nous les utilisons par la suite, avec $K=200$ voisins.

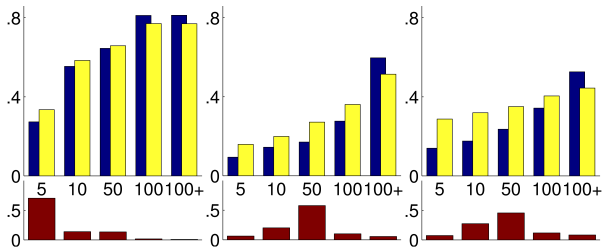

FIG. 2 – Pour les bases Corel5k, ESP Game et IAPR TC12 (de gauche à droite), pour ML (bleu) et σML (jaune), les rappels sont compartimentés par nombre d'images pertinentes : le premier groupe contient les mots avec moins de 5 images pertinentes, le second entre 6 et 10 images, et ainsi de suite. En bas, les histogrammes montrent les proportions de mots dans chaque groupe, En haut, ils montrent le rappel moyen pour les mots de chaque groupe.

5.2 Résultats pour ESP Game et IAPR TC12

Nous appliquons ensuite TagProp aux bases ESP Game et IAPR TC12. La Table 3 résume l'état de l'art pour ces bases et montre les performances de nos variantes SD et ML, ainsi que JEC-15 et σSVM-15. De nouveau, nous observons que nos modèles améliorent l'état de l'art, mais ici de manière plus importante pour la précision que pour le rappel : 18% (resp. 17%) sur $\boldsymbol{P}$, 6% (resp. 2%) sur $\boldsymbol{R}$ et 16 (resp. 15) sur **N+** sur IAPR (resp. ESP).
Notons que le modèle σSVM obtient ici des performances proches. Cela montre la sensibilité de cette méthode à la nature de la base utilisée. Elle est également plus complexe, car composée d'autant de classifieurs que de mots clef.
Comme pour Corel, nous montrons sur la Figure 4 l'influence de la taille des voisinages sur nos différentes mesures pour les variantes σSD et σML. Nous pouvons de nouveau constater l'avantage à apprendre la distance.

5.3 Résultats pour des requêtes complexes

Jusqu'ici, nous avons évalué les performances de requêtes simples, composées d'un mot clef, comme la plupart des travaux existants. Cependant, un système réaliste de requête d'image devrait aussi permettre les requêtes à mots clef multiples. Nous présentons ici nos performances en termes de **BEP** et **mAP** sur la base Corel en incluant ces re-

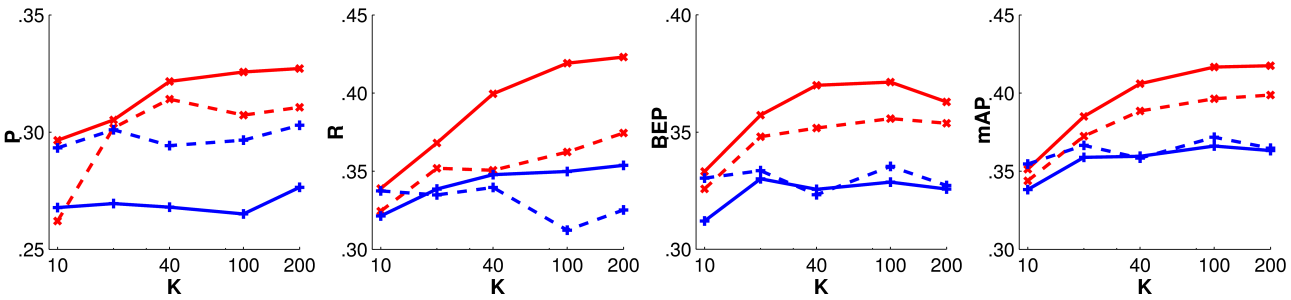

FIG. 3 – Performances sur la base Corel pour nos variantes de poids basés sur la distance, en termes de P, R, **BEP**, et **mAP** en fonction de la taille de voisinage K. Les courbes pointillées correspondent aux variantes ML (rouge) et SD (bleu). Les courbes pleines représentent les extensions sigmoïdales : σML (rouge) et σSD (bleu).

quêtes complexes. Pour permettre la comparaison directe, nous utilisons le même protocole que [7] : seuls les 179 mots qui apparaissent au moins deux fois dans l'ensemble de test sont conservés parmi les 260 que contient Corel. Une image est considérée pertinente pour une requête complexe si elle est annotée par tous les mots de cette requête. Nous considérons les 2241 requêtes composées d'un ou plusieurs mots qui ont au moins une image pertinente dans l'ensemble de test. Ces requêtes peuvent être divisées en requêtes «difficiles» (1820) pour lesquelles seules une ou deux images sont pertinentes, et «faciles» (421) pour lesquelles trois images ou plus sont pertinentes.

La pertinence des images pour une requête complexe est obtenue par multiplication des probabilités (indépendantes selon notre modèle) pour chacun des mots de la requête. Dans la table Table 4, nous résumons les résultat obtenus et les comparons avec ceux de PAMIR, qui surpasse nombre d'alternatives [7]. Nos résultats sont environ 10% meilleurs en **mAP** comme en **BEP** pour tous les types de requêtes.

6 Conclusion

Nous avons présenté de nouveaux modèles pour l'annotation d'image et la recherche d'images par mots clef. Ces modèles combinent une approche par plus proches voisins pondérés avec des capacités d'apprentissage de distance. Nous ajoutons une modulation logistique spécifique à chaque mot pour prendre en compte les différences de fréquence entre mots.

Nous avons évalué nos modèles de manière extensive sur trois bases d'images annotées en utilisant cinq mesures de performance. De ces résultats, nous concluons que la variante σML de TagProp, qui allie poids basés sur la distance et apprentissage de distance, obtient les meilleurs performances, avec un bon rappel et de très fortes précisions sur l'ensemble des bases utilisées, comme résumé dans la Table 5. Ces performances sont supérieures aux travaux précédemment publiés sur le sujet et significativement supérieures au même modèle sans apprentissage de la métrique, σSD. Cela contraste avec les tentatives précédentes d'apprentissage de distance pour les mêmes tâches, par exemple dans [17], car cette tentative n'incluait pas l'apprentissage dans le modèle de prédiction. Les modu-

		IAPR			ESP Game		
		P	R	N+	P	R	N+
MBRM [17]		0.24	0.23	223	0.18	0.19	209
JEC [17]		0.28	0.29	250	0.22	0.25	224
JEC-15		0.29	0.19	211	0.24	0.19	222
σSVM-15		0.48	0.25	227	0.44	0.24	228
TagProp	SD	**0.50**	0.20	215	0.48	0.19	212
	σSD	0.41	0.30	259	0.39	0.24	232
	ML	0.48	0.25	227	**0.49**	0.20	213
	σML	0.46	**0.35**	**266**	0.39	**0.27**	**239**

TAB. 3 – Comparaison des performances (P, R, et **N+**) de l'état de l'art [17] sur les bases ESP et IAPR et de TagProp avec poids basés sur les distances, avec $K = 200$ voisins.

		mAP	Simples	Complexes	Faciles	Difficiles	**BEP**
PAMIR [7]		0.26	0.34	0.26	0.43	0.22	0.17
TagProp	SD	0.32	0.40	0.31	0.49	0.28	0.24
	σSD	0.31	0.41	0.30	0.49	0.27	0.23
	ML	**0.36**	0.43	**0.35**	0.53	**0.32**	**0.27**
	σML	**0.36**	**0.46**	**0.35**	**0.55**	**0.32**	**0.27**

TAB. 4 – Comparaison de TagProp (avec $K = 200$) et PAMIR en termes de **mAP** et **BEP**. Les performances **mAP** sont divisées en requêtes simples et complexes, et en faciles et difficiles. Seuls les 179 mots apparaissant dans au moins deux images de test sont utilisés, comme dans [7].

TagProp	P	R	N+	**BEP**	**mAP**
Corel 5K	32.7%	42.3%	160	36.3%	41.8%
IAPR	46.0%	35.2%	266	40.9%	39.9%
ESP-Game	39.2%	27.4%	239	31.3%	28.1%

TAB. 5 – Résumé des performances de TagProp (variante σML avec $K = 200$) pour les trois bases.

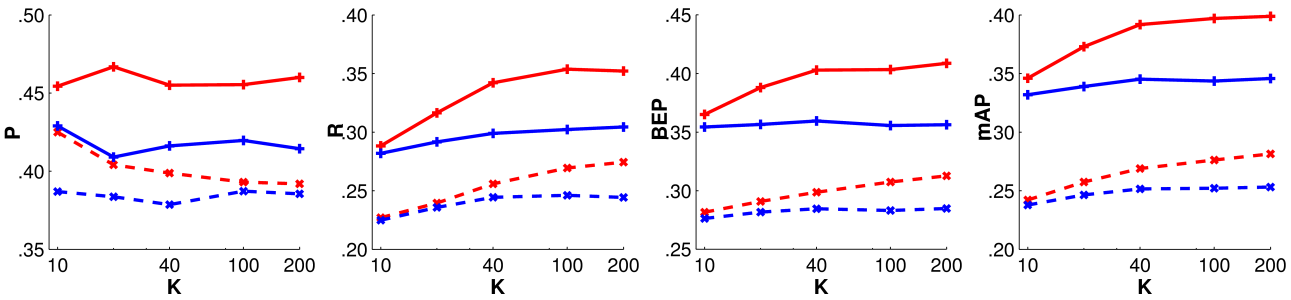

FIG. 4 – Performances des variantes σSD (bleu) et σML (rouge) de TagProp en termes de $\boldsymbol{P}$, $\boldsymbol{R}$, **BEP**, et **mAP**, sur les bases ESP (courbe pleine) et IAPR (pointillés) en fonction de la taille du voisinage K.

lations spécifiques aux mots améliorent les performances globales, et notoirement le rappel des mots rares.
Pour le futur, nous envisageons d'étendre le modèle à l'annotation de regions d'images, pour permettre la détection et la reconnaissance d'objets.

Remerciements

Ces travaux ont été soutenus par le projet européen de recherche CLASS et le projet ANR R2I.

Références

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, 3 :1107–1135, 2003.

[2] G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *PAMI*, 29(3) :394–410, 2007.

[3] C. Cusano, G. Ciocca, and R. Schettini. Image annotation using SVM. In *Proceedings Internet imaging (SPIE)*, volume 5304, 2004.

[4] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation : Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[5] S. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[6] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *NIPS*, 2006.

[7] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *PAMI*, 30(8) :1371–1384, 2008.

[8] M. Grubinger. *Analysis and Evaluation of Visual Information Systems Performance*. PhD thesis, Victoria University, Melbourne, Australia, 2007.

[9] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you ? Metric learning approaches for face identification. In *ICCV*, 2009.

[10] T. Hertz, A. Bar-Hillel, and D. Weinshall. Learning distance functions for image retrieval. In *CVPR*, 2004.

[11] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, 2003.

[12] R. Jin, S. Wang, and Z.-H. Zhou. Learning a distance metric from multi-instance multi-label data. In *CVPR*, 2009.

[13] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *NIPS*, 2003.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features : spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[15] J. Li and J. Wang. Real-time computerized annotation of pictures. *PAMI*, 30(6) :985–1002, 2008.

[16] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recognition*, 42(2) :218–228, 2009.

[17] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*, 2008.

[18] T. Mei, Y. Wang, X. Hua, S. Gong, and S. Li. Coherent image annotation by learning semantic distance. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[19] D. Metzler and R. Manmatha. An inference network approach to image retrieval. In *CIVR*, 2004.

[20] F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation : Constraining the latent space. In *ACM Multimedia*, 2004.

[21] A. Oliva and A. Torralba. Modeling the shape of the scene : a holistic representation of the spatial envelope. *IJCV*, 42(3) :145–175, 2001.

[22] J. Pan, H. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *ACM SIGKDD*, 2004.

[23] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *ECCV*, 2006.

[24] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *ACM SIGCHI*, 2004.

[25] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical Dirichlet process model. In *Workshop on Multimedia Data Mining ACM SIGKDD*, 2008.

[26] A. Yavlinsky, E. Schofield, and S. Rüger. Automated image annotation using global features and robust nonparametric density estimation. In *CIVR*, 2005.

[27] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN : Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, pages 2126–2136, 2006.