Étiquetage automatique de musique : une approche de boosting régressif basée sur une fusion souple d'annotateurs

Rémi Foucard[1] Slim Essid[1] Mathieu Lagrange[2] Gaël Richard[1]

[1] TELECOM Paristech, CNRS-LTCI
37, rue Dareau
75014 Paris, France

[2] Ircam, STMS, CNRS, UPMC
1, place Igor Stravinsky
75004 Paris, France

remi.foucard@telecom-paristech.fr*

Résumé

*L'étiquetage automatique (*tagging*) de musique est le plus souvent traité comme un problème de classification. Dans ce paradigme, l'association d'une étiquette (*tag*) à un mot donné, est définie de manière "dure" : le tag est soit présent, soit absent. Pourtant, l'application d'un tag à un morceau n'est pas toujours évidente. En effet, pendant le processus d'annotation de la vérité-terrain, plusieurs annotateurs peuvent exprimer des doutes, ou être en désaccord les uns avec les autres. Dans cet article, nous proposons de fusionner les annotations des différents annotateurs de manière à conserver l'information sur cette incertitude. Cette fusion nous donne des scores continus, qui sont utilisés pour entraîner un algorithme de boosting régressif. Nos expériences montrent que la régression sur cette vérité-terrain "souple" mène à un apprentissage plus performant, et à de meilleures prédictions, par rapport à la traditionnelle classification binaire.*

Mots clefs

Apprentissage automatique, Music Information Retrieval, Analyse régressive, Tagging automatique, Boosting.

1 Introduction

Les *tags* constituent un outil très utile pour indexer des documents multimédias. Ce sont des labels sémantiques textuels qui décrivent n'importe quel aspect du document, aidant ainsi l'organisation et la structuration de bases de données. Les tags sont largement utilisés sur des services web sociaux comme Flickr ou Last.fm, où les utilisateurs sont invités à associer ces mots-clés aux documents qu'ils partagent ou apprécient.

Malheureusement, cette méthode de "tagging" par les utilisateurs pénalise les documents peu populaires. De plus, les caractéristiques décrites par les tags sont libres de choix, et de nombreux tags potentiellement utiles à la structuration de données peuvent être négligés par les utilisateurs. Pour éviter ces problèmes, des tags plus précis et fiables peuvent être obtenus en consultant des experts, comme le fait le *Music Genome Project*, mené par la radio en ligne Pandora[1]. Chaque morceau de leur catalogue est annoté par des professionnels, qui doivent écouter attentivement les chansons, et passer en revue de nombreux tags, souvent très précis (par exemple, *Jazz_Waltz_Feel*). Ce procédé délivre des annotations de haute qualité, mais se révèle également long et coûteux à mettre en œuvre.

Le tagging automatique est une autre méthode, qui permet d'étiqueter rapidement un grand nombre de documents. De nombreux travaux sur l'"autotagging" ont été publiés ces dernières années [1, 2]. Ils font appel à l'apprentissage automatique pour construire des règles de décision, estimant l'association ou non d'un tag avec un nouveau morceau.

La plupart du temps, cette tâche est considérée comme un problème de classification : un tag est soit présent soit absent. Pour un algorithme d'apprentissage, des associations tag/morceau positives et négatives peuvent être représentées par les scores-cibles 1 et 0. Mais en tant qu'êtres humains, notre réponse à un stimulus musical est plus subtile que cette description binaire. En effet, des annotateurs peuvent être incertains de la classe à choisir, ou peuvent même donner des réponses contradictoires les uns par rapport aux autres. Dans cet article, nous avons recours à la fusion d'annotateurs pour exprimer cette incertitude, ce qui permet de traduire une forme de consensus parmi les annotateurs plutôt que des scores binaires. Ces scores sont ensuite exploités par un algorithme de boosting régressif.

Cet article est organisé de la manière suivante : tout d'abord, nous décrivons en Section 2 les algorithmes de tagging automatique, et les techniques habituelles pour construire une vérité-terrain. Puis, dans la Section 3, nous expliquons notre méthode de fusion d'annotateurs, et la Section 4 présente une méthode de boosting régressif pour exploiter correctement les scores obtenus. La Section 5 décrit une expérience qui valide notre approche. Pour finir, nos concluons et suggérons des recherches à venir dans la dernière section.

*Cette étude a été réalisée dans le cadre du Projet Quaero, financé par OSEO, Agence Française pour l'innovation.

[1] http ://www.pandora.com

2 Tagging automatique des signaux musicaux

L'autotagging est un important sujet de recherche dans le domaine du traitement du signal musical. Par conséquent, de nombreux travaux ont été publiés pour traiter ce problème [3].

La plupart du temps, le signal est découpé en courtes trames, qui peuvent se recouvrir, et sur lesquelles on calcule plusieurs descripteurs. Parmi les descripteurs courants, on peut citer : les Mel Frequency Cepstral Coefficients (MFCC) et leurs dérivées, les chromagrammes, les moments spectraux et le taux de passage par zéro. Le signal est ainsi représenté par une collection de vecteurs de descripteurs (un vecteur par trame).

Puis, un modèle d'apprentissage est utilisé pour construire une règle qui pourra décider automatiquement, d'après les descripteurs, si un tag considéré est présent ou non. Les Machines à vecteurs de support [4], Modèles de mélanges gaussiens [2] et le boosting [5] sont des modèles largement utilisés. Tous ces modèles sont utilisés pour l'apprentissage supervisé, et ont besoin d'une vérité-terrain sur les associations tag/morceau.

Plusieurs méthodes ont été proposées pour étiqueter les exemples d'apprentissage [6]. La première, qui est aussi la plus précise, est l'enquête-formulaire : des annotateurs (experts ou non experts) écoutent le fichier audio et répondent à des questions précises sur le contenu. Cette procédure garantit que les annotateurs ont passé tous les tags en revue. Un jeu d'annotation peut également être utilisé pour étiqueter des données [7]. Des méthodes moins coûteuses consistent à exploiter des tags sociaux ou des documents web [8].

La plupart du temps, les données d'anotation sont traitées pour obtenir des scores-cibles binaires. Ainsi, un tag peut seulement être présent ou absent. Cependant, la plupart des gens s'accordent sur le fait que la musique est une source complexe de données, et donc les concepts derrière chaque tag sont rarement assez précis et clairs pour être représentés par une catégorie binaire.

On trouve deux sources d'incertitude dans la vérité-terrain : l'incertitude individuelle des annotateurs, et les désaccords inter-annotateurs. Par exemple, des concepts comme l'émotion ou l'état d'esprit sont difficiles à catégoriser. En outre, les mots qui décrivent les émotions n'ont pas exactement le même sens selon les personnes. Ce problème peut être pris en compte en plaçant les émotions sur un espace continu en deux dimensions (valence-intensité). La reconnaissance d'émotion peut alors être formulée comme un problème de régression [9] ou de ranking [10]. La procédure d'annotation consiste à situer les morceaux dans l'espace bidimensionnel, où à les ordonner. Les scores-cibles sont obtenus en prenant la moyenne des réponses individuelles. Cette formulation convient très bien à la tâche de reconnaissance d'émotions, mais elle ne construit pas de catégories, et nécessiterait donc des étapes supplémentaires de traitement pour être adaptée au tagging automatique.

Dans [11], les auteurs tirent partie des corrélations entre les tags pour construire des catégories intermédiaires ordonnées, qui représentent plusieurs niveaux de confiance. Cependant, ces catégories sont fabriquées à partir des scores binaires. L'incertitude des annotateurs est donc seulement déduite, au lieu d'être directement utilisée.

3 Fusion souple d'annotateurs

3.1 Motivation

Nous exploitons dans cette étude la base de données CAL500 [2], annotée par la méthode du formulaire. Chaque morceau a été annoté par au moins trois personnes. Pour beaucoup de tags dans le formulaire, les annotateurs choisissent entre plusieurs niveaux de confiance. Par exemple, pour annoter une chanson donnée, chaque type d'émotion peut être noté de 1 à 5. Comme expliqué dans la section précédente, cette manière de produire les données génèrera probablement une incertitude, tant au niveau de chaque annotateur, que collectivement à cause de leurs désaccords.

Nous proposons une fusion d'annotateurs qui produit des scores continus, afin d'être plus flexible et d'exprimer ces doutes. Malheureusement, les scores souples fournis avec la base de données ne sont pas pleinement documentés et leur méthode de construction à partir des annotations individuelles demeure, dans certains cas, difficile à comprendre. Pour ces raisons, nous avons construit nos propres scores souples, à partir des réponses individuelles des annotateurs.

3.2 Procédure de fusion

Pour commencer, chaque réponse possible est convertie en une valeur $v \in [0, 1]$. Les valeurs consécutives sont régulièrement espacées, de plus, $v = 0$ et $v = 1$ doivent toujours être possibles. Par exemple, il y a quatre réponses possibles pour le tag *Instrument-Trompette* : *Absent*, *Peut-être*, *Présent* et *Au-premier-plan*. Ces choix sont convertis respectivement en : 0, 0.33, 0.67 et 1.

Ensuite, pour un tag et un morceau donnés, on compte plusieurs manières de fusionner les réponses individuelles. Pour les scores binaires de CAL500, un tag est considéré comme "positif" si 80% des sujets jugent que le tag s'applique [12]. Parmi les autres méthodes de fusion, on trouve : le vote majoritaire (le score correspond alors à la catégorie la plus choisie), ou bien prendre la moyenne (éventuellement seuillée) des annotations individuelles. Le vote majoritaire et la moyenne seuillée ne reflètent pas l'incertitude ; c'est pourquoi nous choisissons de prendre la moyenne des scores individuels (comme le font Yang *et al.* dans [9, 10], mais pour un type différent de vérité-terrain). Soit V_s le score souple correspondant au morceau considéré. On a donc :

$$V_s = \frac{1}{K} \sum_{k=1}^{K} v_k, \tag{1}$$

où v_k est la valeur correspondant au choix de l'annotateur k, et K est le nombre d'annotateurs.
Pour les tags "négatifs" (par exemple *Émotion-PAS-joyeux*), la valeur est simplement $V = 1 - P$, où P est la valeur associée au tag "positif" correspondant.
Pour valider les scores souples obtenus par ce procédé, nous mesurons leur accord avec les scores binaires fournis par le Computer Audition Lab avec CAL500. Le coefficient Kappa de Cohen [13] est conçu pour ce genre d'évaluation. Il prend en entrée deux ensembles d'annotations binaires. Il est calculé selon l'expression :

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{2}$$

où $Pr(a)$ désigne le taux d'accord entre les deux ensembles d'annotations, et $Pr(e)$ est la probabilité d'obtenir un accord par hasard sur un élément donné, calculée en imaginant que l'on ne connaîtrait que la répartition globale sur chaque classe, de chaque ensemble d'annotations.
De manière à obtenir des valeurs comparables, nous construisons donc de nouveaux scores binaires V_h, correspondant aux scores souples reconstruits V_s. Les nouveaux scores binaires sont obtenus par seuillage de nos valeurs souples :

$$V_h = \begin{cases} 1 & \text{si} \quad V_s > t \\ 0 & \text{sinon} \end{cases} . \tag{3}$$

Le seuil qui produit l'accord le plus élevé ($t = 0.64$) donne un Kappa de Cohen moyen de $\kappa = 0.80$ entre les deux ensembles de labels. D'après [13], cette valeur traduit un accord excellent. C'est donc ce seuil que l'on utilisera pour construire V_h.

4 Boosting régressif pour tagging souple

Les scores souples obtenus par la fusion d'annotateurs sont ensuite utilisés pour entraîner un système de boosting régressif.

4.1 Descripteurs

Dans notre système, les données audio sont représentées par les descripteurs suivants : les 15 descripteurs psychoacoustiques recommandés dans [9] (loudness, dissonance tonale, ...)[2], auxquels on ajoute les classiques 13 premiers MFCC (sans l'énergie), chroma, taux de passage par zéro, dispersion, asymétrie et kurtosis spectraux. Ces descripteurs sont présentés dans le Tableau 1. Ils sont d'abord calculés sur des trames de 23 ms, avec un recouvrement de 50%. La dernière étape avant l'apprentissage consiste à intégrer ces trames temporellement, en prenant leur moyenne sur 2 s.

4.2 Boosting régressif

Sur ces descripteurs, nous appliquons deux algorithmes de boosting. Le boosting est une technique d'apprentissage

[2]Ces descripteurs ont été extraits à l'aide de Psysound (`http ://www/psysound.org/`).

Descripteur	Dim.	Description
Centroïde spectral	1	Centroïde du spectre
Diffusion spectrale	1	Diffusion de l'énergie spectrale
Asymétrie spectrale	1	Asymétrie du spectre
Kurtosis spectral	1	"Aplatissement" du spectre
Taux de passage par zéro	1	Fréquence du changement de signe
Loudness	1	Intensité sonore perçue
Sharpness	2	Contenu en hautes fréquences
Largeur tonale	1	Platitude de la fonction de Loudness
Volume	1	Taille du son perçue
Dissonance spectrale	2	Rugosité des composantes spectrales
Dissonance tonale	2	Rugosité des composantes tonales
Tonalité pure	1	Audibilité des tons spectraux
Tonalité complexe	1	Audibilité des tons virtuels
Multiplicité	1	Nombre de tons perçus
Tonalité	1	Tonalité du morceau
Accord	1	Accord instantané
MFCC	13	Description cepstrale
Chroma	12	Énergie pour chaque classe de notes

Tableau 1 – *Descripteurs utilisés par les systèmes d'apprentissage*

initialiser les poids des exemples $w_i \longleftarrow \frac{1}{2m}, \frac{1}{2l}$, resp. pour $y_i = 0, 1$, où m et l sont respectivement le nombre d'exemples positifs et négatifs ;

pour $r = 1, \ldots, R$ **faire**
Entraîner un classifieur $T_r(x)$ sur les données d'apprentissage, avec les poids w_i;

`// Taux d'erreur pondérée`
$\epsilon_r \longleftarrow \frac{1}{\sum_i w_i} \sum_i w_i I(y_i \neq T_r(x_i))$;

`// Coefficient associé à` T_r
$\alpha_r \longleftarrow \log \frac{1}{\beta_r}$, où $\beta_r = \frac{\epsilon_r}{1-\epsilon_r}$;

`// Mise à jour des poids des exemples`
pour chaque *exemple* x_i *bien classifié par* T_r **faire**
$w_i \longleftarrow w_i \beta_r$
fin
fin

Sorties : $H(x) = I(\sum_r \alpha_r T_r(x) \geq \frac{1}{2} \sum_r \alpha_r)$

Algorithme 1 : Adaboost.

initialiser les valeurs-cibles des exemples $m_i \longleftarrow y_i$;

pour $r = 1, \ldots, R$ **faire**
Entraîner un régresseur $T_r(x)$ sur les données d'apprentissage, avec les cibles m_i ;

`// Mise à jour des valeurs cibles`
pour chaque *exemple* x_i **faire**
$m_i \longleftarrow m_i - T_r(x_i)$
fin
fin

Sorties : $H(x) = \sum_r T_r(x)$

Algorithme 2 : Boosting régressif avec erreur quadratique.

qui entraîne itérativement plusieurs versions complémentaires d'un classifieur "faible" (supposé de médiocre performance). La version la plus connue du boosting est probablement Adaboost, qui est décrit dans l'Algorithme 1. Cette version utilise des poids pour donner de l'importance à des exemples particuliers. À chaque itération r, les poids des exemples correctement classifiés par le classifieur faible T_r, sont diminués. On mettra ainsi l'accent sur les autres exemples lors des itérations suivantes.
Au départ, le boosting est un algorithme de classification, mais il a été généralisé pour faire de la régression avec des fonctions de coût différentiables [14]. Dans le cas de l'erreur quadratique, il n'y a pas de système de pondération. Ce sont les valeurs-cibles de l'apprentissage qui vont changer à chaque itération. Pour le régresseur T_r, les cibles sont les résidus des predictions :

$$res_{i,r} = y_i - \sum_{k=1}^{r-1} T_k(x_i) \tag{4}$$

où y_i est le score-cible de l'exemple i, et x_i est le vecteur de description correspondant. L'algorithme de boosting régressif pour l'erreur quadratique est présenté dans l'Algorithme 2.
Pendant la phase de test, un score unique S_n est obtenu pour chaque morceau n en prenant la moyenne des prédictions $H(x)$ de l'algorithme, sur toutes les trames du morceau.

5 Validation de l'approche

Nous menons une expérience pour démontrer l'utilité de nos scores souples fusionnés par rapport aux scores binaires, et l'efficacité de notre système de régression. Pour ce faire, nous exécutons deux systèmes de prédiction sur les mêmes données audio : l'un est entraîné sur les labels binaires V_h, et l'autre sur les scores souples V_s.

5.1 Cadre expérimental

L'expérience est réalisée sur la base de données CAL500. Cette base contient 500 chansons pop, avec des tags décrivant l'émotion, l'instrumentation, le genre, *etc*. Nous utilisons les 61 mêmes tags que dans [12]. Les tests sont menés avec une validation croisée de 10 folds, gardant 450 morceaux pour l'apprentissage, et 50 pour le test. Pour réduire la complexité, nous n'utilisons que 30 s de chaque morceau : entre les instants 30 s et 60 s.
Nous entraînons un système de classification Adaboost sur les labels binaires recréés V_h, et un système de régression avec les scores souples V_s. Chacun d'entre eux va être entraîné avec 500 itérations de boosting. Nos classifieurs faibles sont des arbres de décision à deux branches, comme dans [1].
Le but de cette étude est de prouver qu'une analyse régressive sur des scores souples est utile, même dans le cas où la tâche finale est une classification. C'est pourquoi, pour comparer les prédictions des deux systèmes sur l'ensemble de test, nous mesurons leur capacité à prédire la vérité-terrain binaire. Nous utilisons deux mesures de ranking différentes pour l'évaluation de ces prédictions. Les mesures de ranking évaluent une liste d'exemples ordonnés en fonction de leur score prédit S_n. Cette liste est comparée à la vérité-terrain binaire. Un classement parfait mettrait tous les morceaux positifs en haut de la liste. Notre première mesure est la "Mean Average Precision" (MAP). Elle peut être obtenue en parcourant la liste ordonnée de haut en bas, et en prenant la moyenne des précisions obtenues à chaque exemple vraiment positif. Nous utilisons également la courbe "Receiver Operating Characteristic" (ROC). Cette courbe représente le taux de vrais positifs en fonction du taux de fausses alarmes, calculés à chaque élément de la liste ordonnée. L'aire sous la courbe (AROC) sera notre seconde mesure.

Méthode de fusion	MAP	AROC
Binaire	0.46	0.67
Souple	0.50	0.71

Tableau 2 – *Performances sur CAL500 avec les fusions d'annotateurs binaire et souple.*

5.2 Résultats

La performance des deux systèmes est présentée dans le Tableau 2. On observe clairement que le système régressif donne de meilleures prédictions par rapport au système de classification. La différence entre les deux systèmes est évaluée grâce à un *t*-test de paires, en validation croisée [15]. Ce test estime que la différence de performances est significative, avec une confiance de 99%. Cela signifie que l'information sur l'incertitude, portée par les scores souples, est réellement utile pour des systèmes d'apprentissage.
Il est important de remarquer que le système régressif ne nécessite pas davantage de données d'annotation que l'autre système : la seule différence entre les deux est le traitement de ces annotations. Et les résultats montrent qu'il y a effectivement une perte d'information utile lorsque ce traitement aboutit à des scores binaires.
La performance du système de classification binaire a été

comparée à des systèmes de référence sur les mêmes données dans [5].

6 Conclusion

Dans cette article, nous avons décrit une manière de fusionner les annotations qui préserve l'information sur l'incertitude de l'association tag/morceau. Nous avons proposé d'utiliser un boosting régressif pour apprendre les scores obtenus par cette fusion. Nos tests montrent que les scores souples, combinés à l'apprentissage régressif, conduisent à un meilleur apprentissage des tags.
De futurs travaux pourront porter sur l'exploitation des corrélations entre tags, dont l'utilité pour le tagging d'audio a été démontrée [11]. En effet, dans l'étude ici présentée, les tags ont été considérés comme des concepts indépendants. Cependant, des tags comme *Morceau-Très_dansant* et *Utilisation-Lors_d'une_fête* ont de bonnes chances d'apparaître en même temps sur de nombreux morceaux. Cette corrélation pourrait être exploitée par des méthodes comme la régression multivariée.

Références

[1] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, et B. Kégl. Aggregate features and ADABOOST for music classification. *Machine Learning*, 65(2-3) :473–484, 2006.

[2] D. Turnbull, L. Barrington, D. Torres, et G. Lanckriet. Semantic Annotation and Retrieval of Music and Sound Effects. *TASLP*, 16(2) :467–476, Février 2008.

[3] T. Bertin-Mahieux, D. Eck, et M. Mandel. Automatic Tagging of Audio : The State-of-the-Art. Dans Wenwu Wang, éditeur, *Machine Audition : Principles, Algorithms and Systems*. IGI Publishing, 2010.

[4] C. Xu, N. Maddage, X. Shao, F. Cao, et Q. Tian. Musical genre classification using support vector machines. Dans *ICASSP*, pages 429–432, 2003.

[5] R. Foucard, S. Essid, M. Lagrange, et G. Richard. Multi-scale temporal fusion by boosting for music classification. Dans *ISMIR*, 2011.

[6] D. Turnbull, L. Barrington, et G. Lanckriet. Five Approaches to Collecting Tags for Music. Dans *ISMIR*, pages 225–230, 2008.

[7] E. Law et L. von Ahn. Input-agreement : a new mechanism for collecting data using human computation games. Dans *CHI*, pages 1197–1206, New York, NY, USA, 2009. ACM.

[8] C. Laurier, M. Sordo, J. Serrà, et P. Herrera. Music mood representations from social tags. Dans *ISMIR*, pages 381–386, 2009.

[9] Y. Yang, Y. Lin, Y. Su, et H. Chen. A Regression Approach to Music Emotion Recognition. *TASLP*, 16(2) :448–457, 2008.

[10] Y. Yang et H. Chen. Ranking-Based Emotion Recognition for Music Organization and Retrieval. *TASLP*, (99), 2010.

[11] Y. Yang, Y. Lin, A. Lee, et H. Chen. Improving Musical Concept Detection by Ordinal Regression and Context Fusion. Dans *ISMIR*, pages 147–152, 2009.

[12] L. Barrington, M. Yazdani, D. Turnbull, et G. Lanckriet. Combining Feature Kernels for Semantic Music Retrieval. Dans *ISMIR*, pages 614–619, 2008.

[13] J. Landis et G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1) :159–174, 1977.

[14] T. Hastie, R. Tibshirani, et J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 3e édition, 2009.

[15] T.G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7), 1998.