Le génie pour l'industrie

GTI770
Systèmes intelligents et apprentissage machine

TP04 — Développement d'un système intelligent

1. Contexte

L'explosion de la quantité de pièces musicales accessibles sur Internet a créé de nouveaux intérêts dans le domaine de la distribution de la musique, aussi bien pour les utilisateurs qui recherchent des moyens pratiques de gérer et d'utiliser leurs collections personnelles que pour les distributeurs et services de musique en flux continu (*streaming*) qui cherchent à donner accès à ses utilisateurs à la musique qui les intéresse. Pensons, par exemple, aux services de musique en ligne Spotify ou Apple Music qui conçoivent automatiquement des listes de lecture adaptées aux goûts de l'utilisateur.

Les recherches actuelles ont pour but d'extraire des descripteurs pertinents dans le cadre des applications visées, par exemple la recherche efficace de musique sur l'Internet ou encore l'identification du genre musical. Le défi repose surtout sur l'automatisation de ces tâches puisque le volume d'information à traiter ne permet pas une manipulation cas par cas. De plus, la manipulation du fichier audio dans son entièreté requiert énormément de ressources computationnelles et de bande passante.

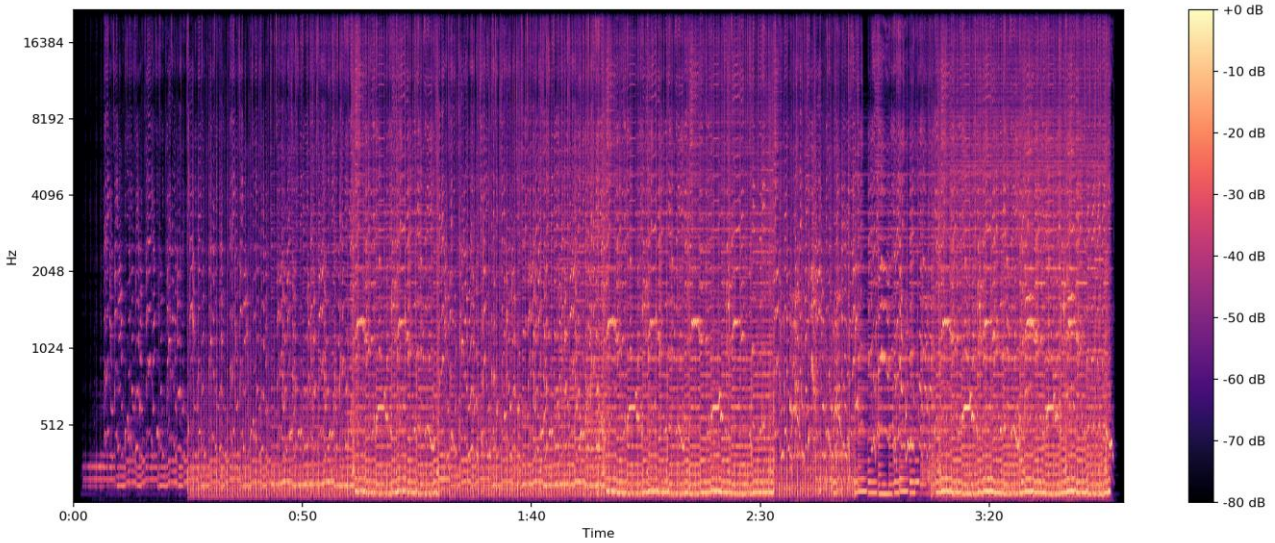
Différentes techniques existent pour extraire de manière automatique une information perceptive à partir de signaux acoustiques. Ainsi, certains descripteurs simples peuvent être extraits directement à partir du signal audio en utilisant des algorithmes spécifiques, tandis que d'autres descripteurs de hauts-niveaux nécessitent non seulement l'extraction de caractéristiques primaires à partir du signal (les descripteurs de bas-niveau), mais également la construction d'un modèle permettant de combiner ces paramètres pour obtenir le descripteur désiré. La modélisation finale consiste à lier les valeurs des descripteurs aux résultats perceptifs (genres musicaux étudiés) en utilisant des méthodes d'apprentissage automatiques.

Dans ce laboratoire, vous êtes amené à concevoir et implémenter un système intelligent qui résout un problème de classification de genres audio. Vous devrez, par le fait même, intégrer une notion supplémentaire acquise récemment dans le cours théorique, à savoir la combinaison de modèles d'apprentissage.

Vous devrez concevoir non seulement vos modèles d'apprentissage, mais également un script permettant de classifier de nouvelles chansons. L'évaluation de ce travail sera basée sur la qualité de la conception, de même que sur la performance du système lors de l'évaluation. Sur le total de 100 points du laboratoire, 10 points seront consacrés à la performance de votre solution par rapport aux autres équipes. L'équipe ayant la meilleure performance de classification aura les 10 points, alors que les autres équipes auront 1 point de moins, en ordre décroissant de performance de leur modèle, jusqu'à 0.

Malheureusement, vous n'avez pas accès directement aux fichiers audio servant à entraîner votre modèle d'apprentissage. Vous n'avez seulement accès qu'à différents descripteurs de fichiers audio qui sont décrits en annexe.

Figure 1.1 : *Mel frequency cepstral coefficients* (MFCCs) de la chanson « *Rolling in the deep* » de l'artiste anglaise *Adele Laurie Blue Adkins* extraits avec la librairie *LibROSA*. 26 valeurs sont prélevées de ce spectrogramme pour former les MFCCs disponibles



dans vos vecteurs de primitives.

Afin de construire vos modèles, vous disposez des vecteurs de primitives qui ont été extraits de 179 555 fichiers audio MP3. Les échantillons se retrouvent en un seul fichier, où l'ensemble des 179 555 morceaux qui consistent en l'ensemble d'apprentissage sont étiquetés selon 25 genres musicaux.

Il y'a plusieurs aspects qui vous devez considérer avant de manipuler les données, par exemple :

- les primitives ne sont pas normalisées;
- le nombre de dimensions de certains vecteurs de primitive étant élevé, une méthode de réduction de dimensionnalité tel que PCA pourrait être envisagée.

Veuillez également prendre note que tous les fichiers sont dans le format. csv. Notez que les fichiers ont quelques informations additionnelles :

- *@attribute SAMPLEID numeric;*
- *@attribute TRACKID string.*

Le premier attribut est le numéro d'identification de l'ordre de l'échantillon, alors que le deuxième attribut est un identificateur des pièces musicales. Ils doivent donc être éliminés avant de faire l'apprentissage et l'évaluation.

Contrairement aux laboratoires précédents dans lesquelles vous travailliez avec des classificateurs indépendants sur un seul ensemble de données, vous devez dans ce présent laboratoire combiner des algorithmes d'apprentissage afin de classifier les échantillons d'un ensemble de données regroupant

plusieurs sortes de primitives (*feature sets*). Vous implémenterez ce système avec les librairies *TensorFlow* ou *scikit-learn*. Finalement, vous aurez à proposer une stratégie combinatoire de classificateurs (voir la section *Combinaison de modèles* ainsi que la section *Vote des modèles* dans les notes de cours théoriques) tel que *Vote de majorité, moyenne, produit, médiane*, etc. Comme présenté dans la portion théorique du cours, une stratégie de combinaison peut se résumé à ce diagramme :

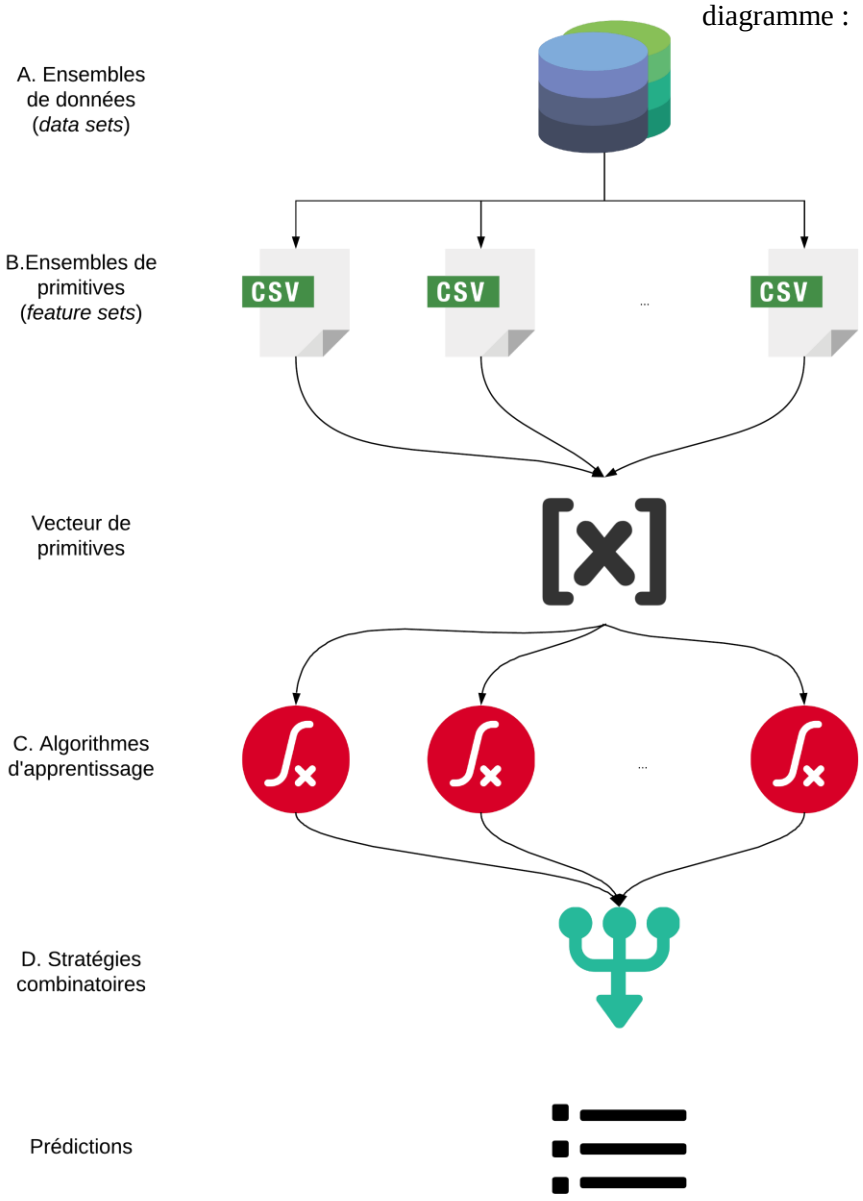


Figure 1.2 : Digramme explicatif d'un *ensemble*.

2. Objectifs

Les principaux objectifs de ce laboratoire sont :

1. Produire un code source utilisant les technologies Python et des techniques d'apprentissage machine permettant de classer des échantillons de donnés automatiquement;
2. De construire un véritable système intelligent dynamique avec une combinaison de modèles et d'algorithmes d'apprentissage;
3. Se familiariser avec le langage de programmation *Python* et les outils et librairies scientifiques d'apprentissage machine;

3. Matériel fourni

Vous disposez du matériel suivant :

- Plusieurs ensembles de données d'entrainement comportant qui présentent différents ensembles de primitives audio;

.

4. Manipulations

1. Téléchargez le fichier de primitives précalculées à partir de https://drive.google.com/file/d/1rEOofbBpiwkVBIgCAEKSn6ao2urzqbGU/view?usp=sharing
2. Analysez les données. Définissez votre méthodologie de prétraitement et de validation des données. Vous aurez à justifier et décrire cette méthodologie dans votre rapport.
    - Si vous jugez pertinent d'inclure une méthode de réduction de dimensionalité, celle-ci dois faire partie de votre méthodologie et vous devrez justifier en conséquence.
3. Choisissez 3 feature sets (ensemble de primitives) et entrainez un algorithme de votre choix par feature set. Entraînez-les en choisissant judicieusement vos hyperparamètres que vous prendrez en note. Consignez les scores de précision (*accuracy*) et F1 dans un tableau et faites un graphique permettant d'évaluer les différents algorithmes d'apprentissage sur ces trois ensembles de données. Discutez des résultats dans le rapport.
   Le meilleur classificateur issu de cette première étude sera votre classificateur de base (*baseline classifier*). Il vous servira de base de comparaison afin de comparer la performance entre un classificateur unique et votre futur ensemble de classificateurs.
4. Réalisez une combinaison de modèles d'apprentissage. Pour ce faire, vous pouvez combiner des modèles précédemment élaborés ou en générer d'autres. **Au moins 3 modèles** doivent faire partie de votre combinaison. Votre règle de décision peut être inspirée des notes de cours ou de la librairie *scikit-learn* et de sa documentation, pourvu que vous justifiiez le choix de celle-ci plus tard dans le rapport. Vous êtes également libre d'inclure dans vos modèles une méthode classique de la théorie des ensembles telle que *bagging*, *boosting*, *Random Forest.*
5. Produisez un diagramme afin d'illustrer votre stratégie d'ensemble qui permet clairement de voir quels ensembles de primitives vous avez choisis d'inclurent, quels classificateurs font partie de votre solution et quelle stratégie d'agrégation vous avez choisi. Joignez-le à votre rapport.
6. Comparez les performances entre votre algorithme de base (*baseline)* et votre combinaison de modèles précédemment conçu en 5. Discutez de ces résultats dans le rapport. Présentez dans le rapport un tableau des résultats et un graphique permettant de visualiser les résultats.
7. Exportez vos modèles. Cet exemple vous aidera à exporter vos modèles à l'aide de la librairie *scikit-learn*, alors que ce lien vous aidera avec TensorFlow.
8. Produisez un script permettant de lancer un processus d'inférence (prédiction) sur votre système. Ce script devrait être appelé de la manière suivante :

   ```
   python3 [-- arguments] [input_feature_vectors.csv] output.csv
   ```

   Si votre script nécessite des arguments, pensez à les documenter dans un fichier README.md, fourni dans le même dossier que le script.

   Le fichier en sortie devrait ressembler à ceci :

```
METAL_ALTERNATIVE
PUNK
ROCK_NEO_PSYCHEDELIA
POP_INDIE
HIP_HOP_RAP
POP_CONTEMPORARY
ROCK_HARD
HIP_HOP_RAP
POP_CONTEMPORARY
.
.
FOLK_INTERNATIONAL
METAL_HEAVY
```

9. À l'aide de ce même script, inférez les valeurs du fichier fourni ne comportant pas d'étiquettes. Vous devez inclure le fichier produit en sortie dans votre remise. Comme mentionné précédemment, une partie de l'évaluation du laboratoire sera basée sur les résultats obtenus.

5. Notebook

Votre notebook devra contenir les réponses aux questions suivantes. Il devra notamment avoir une analyse détaillée des résultats de classification obtenus par les différents modèles et leurs variations d'hyperparamètres.

5.1 Questions du rapport

1. Avec les liens fournis en l'annexe de cet énoncé et avec vos trouvailles faites sur Internet par le biais de vos recherches personnelles, faites, à titre d'introduction, une revue de la littérature. Expliquez en quelques paragraphes comment nous réussissons aujourd'hui à classifier différents sons et pièces musicales automatiquement afin de bien comprendre le sujet sur lequel vous travaillez.
2. Quelle est la configuration (machine, matériel, versions logiciel) de votre environnement? Quelle a été votre approche de partitionnement des données? Quels ensembles de primitives avez-vous choisis? Quelle méthode de validation avez-vous utilisée afin de confectionner vos modèles? Quelles étapes supplémentaires avez-vous eu à effectuer en prétraitement (normalisation, balancement, réduction de dimensionnalité, etc.)?
3. Quels sont les trois modèles d'apprentissage que vous avez décidé d'étudier à titre de classificateur de base? Exprimez les raisons qui vous ont mené à un tel choix. Si vous avez décidé d'implémenter un réseau de neurones, décrivez la structure de votre modèle d'apprentissage par réseau de neurones. Ajoutez tous graphiques ou représentation pertinente afin de décrire votre modèle, par exemple, un graphe *TensorBoard* si applicable ou un texte descriptif.

4. Pour vos trois modèles, présentez les hyperparamètres d'apprentissage et les ensembles de données utilisées ayant menés à votre meilleur résultat de précision (*accuracy*) et *F-measure*. Comment ces classificateurs ont-ils performés sur cet ensemble de données ? Discutez des résultats.
5. Présentez la conception de votre solution finale au problème (votre solution reposant sur la théorie des ensembles). Présentez ici le diagramme nécessaire afin de présenter convenablement votre combinaison de modèles, les ensembles de primitives choisies ainsi que la stratégie de combinaison. Faites une discussion expliquant vos décisions de conception. Faites des liens avec l'implémentation et présentez le code clé de celle-ci.
6. Consignez dans un tableau les hyperparamètres finaux de vos modèles faisant partie de votre ensemble. Présentez le score de précision (*accuracy*) et F1 final de votre ensemble. Présentez une discussion faisant l'analyse des résultats de votre système final. Décrivez les problèmes et difficultés rencontrées. Décrivez les performances de votre ensemble et tentez d'expliquer ces résultats.
7. Formulez quelques pistes d'amélioration de la solution développée.

Annexe 1 : Sources d'information pertinentes

LIENS DIVERS

Documentation de la librairie *scikit-learn* : http://scikit-learn.org/stable/documentation.html

Documentation de la librairie *Google TensorFlow* : https://www.tensorflow.org/api_docs/python/

TensorBoard: Graph Visualization: https://www.tensorflow.org/get_started/graph_viz

Proceedings of the International Society for Music Information Retrieval Conference
http://www.ismir.net/conferences.html

Proceedings of the Sound and Music Conference: http://www.smc-conference.org
IEEE Explorer—http://ieeexplore.ieee.org

Science direct Elsevier: http://www.sciencedirect.com

Million Song Dataset: http://labrosa.ee.columbia.edu/millionsong

TU-WIEN MSD benchmark dataset: http://www.ifs.tuwien.ac.at/mir

Music Information Retrieval, Vienna University of Technology: http://www.ifs.tuwien.ac.at/mir/audiofeatureextraction.html

Librairie LibROSA: https://librosa.github.io/librosa/master/index.html

JAudio: A feature extraction library, McGill University: http://www.music.mcgill.ca/~cmckay/papers/musictech/jAudio_ISMIR_2005.pdf

LIVRES

Aurélien Géron. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow—Concept, Tools, and Techniques to Build Intelligent Systems.* 566 p. ISBN-13: 978–1491962299

Annexe 2 : Description des ensembles de primitives

Tableau A2.1 : Description des ensembles de primitives

Ensemble de primitives	Description (en anglais seulement)	Dimensions
Modulation Frequency Variance Descriptor	This descriptor measures variations over the critical frequency bands for a specific modulation frequency (derived from a rhythm pattern). Considering a rhythm pattern, i.e. a matrix representing the amplitudes of 60 modulation frequencies on 24 critical bands, an MVD vector is derived by computing statistical measures (mean, median, variance, skewness, kurtosis, min and max) for each modulation frequency over the 24 bands. A vector is computed for each of the 60 modulation frequencies. Then, an MVD descriptor for an audio file is computed by the mean of multiple MVDs from the audio file's segments, leading to a 420-dimensional vector.	420
Temporal Rhythm Histograms	Statistical measures (mean, median, variance, skewness, kurtosis, min and max) are computed over the individual Rhythm Histograms extracted from various segments in a piece of audio. Thus, change and variation of rhythmic aspects in time are captured by this descriptor.	420
Statistical Spectrum Descriptor	The Sonogram is calculated as in the first part of the Rhythm Patterns calculation. According to the occurrence of beats or other rhythmic variation of energy on a specific critical band, statistical measures are able to describe the audio content. Our goal is to describe the rhythmic content of a piece of audio by computing the following statistical moments on the Sonogram values of each of the critical bands: mean, median, variance, skewness, kurtosis, min- and max-value	168
MARSYAS	Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals) is an open source software framework for audio processing with specific emphasis on Music Information Retrieval applications.	124
JMIR Derivatives	Derivatives of low level spectral features.	96

Tableau A2.1 : Description des ensembles de primitives

Ensemble de primitives	Description (en anglais seulement)	Dimensions
Rhythm Histogram	The Rhythm Histogram features we use are a descriptor for general rhythmics in an audio document. Contrary to the Rhythm Patterns and the Statistical Spectrum Descriptor, information is not stored per critical band. Rather, the magnitudes of each modulation frequency bin of all critical bands are summed up, to form a histogram of "rhythmic energy" per modulation frequency. The histogram contains 60 bins which reflect modulation frequency between 0 and 10 Hz. For a given piece of audio, the Rhythm Histogram feature set is calculated by taking the median of the histograms of every 6 second segment processed.	60
JMIR MFCCs	In sound processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping can allow for better representation of sound, for example, in audio compression.	26
JMIR LPC	Linear predictive coding.	20
JMIR Spectral	Low level spectral features.	16
JMIR Moments	This feature consists of the first five statistical mo- ments of the spectrograph. This includes the area (ze- roth order), mean (first order), Power Spectrum Density (second order), Spectral Skew (third order), and Spectral Kurtosis (fourth order). These features describe the shape of the spectrograph of a given window	10