Application of SVM to Blend Prediction **Julian Vanasse**
MPCL 2020, April 29

1 Introduction

Presented here is a brief overview articulating the path the author is taking regarding the classification of orchestral blend in multitrack audio stems. The method proposed necessitates the creation of several data-types, containers of information that will modularize the classification process. These objects will mirror the ground-truth data, which comes to us in the form of score annotations indicating which pairs of measures blend with one another. Described below is the framing of this problem first as a binary classification scheme using the Support Vector Machine (SVM) approach, which can be extended to multiclass classification.

Note on terminology: as in previous discussions of this project, the author distinguishes the two uses of harmonic as harmonic *(partial)* to denote when the sinusoidal elements of a sound are integrally related and harmonic *(tonal)* to denote when pitch relationships fall into the tonal interval relationships such as 8ve or fifth. For reasons that will become clear below, this process deals much with the feature space concerning individual voices and feature space concerning pairs of voices. These are distinguished by terms *singular* and *relative* respectively. As this project moves forward there will be inevitable confusion between musical terminology and mathematical or statistical learning terminology. The musical *measure* will be mistaken for the *measure* space, the musical *score* will be mistaken for success rate *score* of a learning process. An attempt has been made to be as clear as possible when mixups are possible.

2 Problem Statement

The ground-truth is derived from expert annotation indicating where blend occurs in orchestral excerpts. More precisely, the annotations indicate which pairs of measures blend over what intervals. Consider a typical excerpt, Schubert's Symphony No. 9 (D 944), IV, mm. 543-558 with annotations in in Figure 1. These annotations indicate that the flute and clarinet voices blend over mm. 543-558, and the violin I, violin 2, cello and oboe blend over mm. 547-548 and mm. 555-556. Each blend annotation is also paired with a rating indicating the strength of the effect from 1 to 5.

The annotations are found in a more data-science friendly form as Excel files indicating which voices are blending and over what measures of the excerpt. Luckily, the audio has been synthesized with a click track permitting the indexing of the voices by measures, which are also the smallest unit of time in the blend annotations. The task then is to design a map that takes as input two audio vectors $x_{m,i}$ and $x_{m,j}$ for measure number m and voices i and j and result in a blend rating scalar value

$$Y(x_{m,i}, x_{m,j}) \in \{0, 1, 2, 3, 4, 5\}$$

where 0 indicates no blend and the other integral values in the range of Y are blend ratings.

The first step will be not solving this problem but solving a weaker version: binary detection of blend or non-blend.

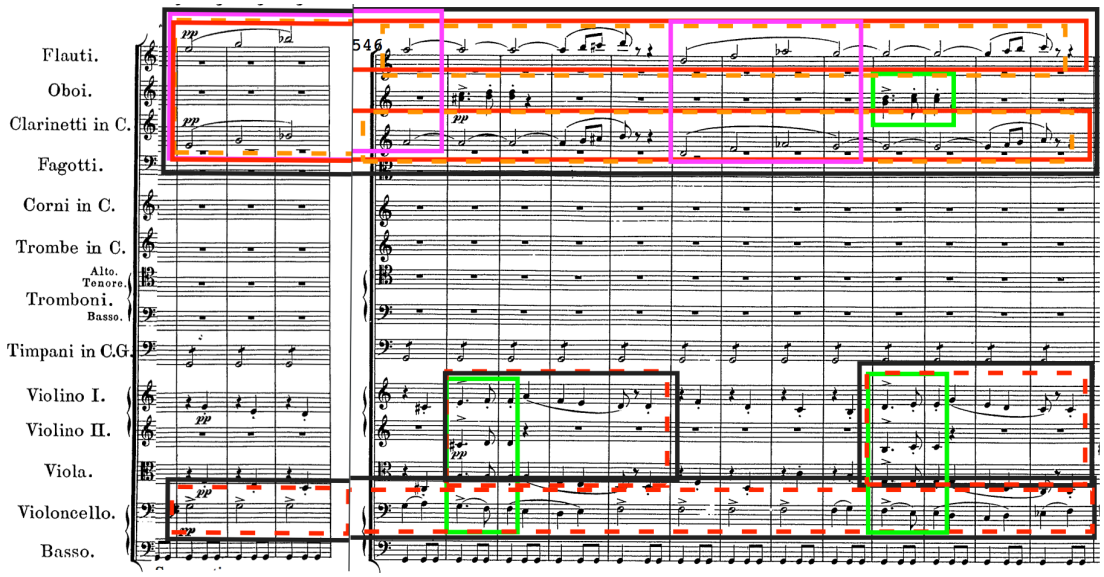

Figure 1: Schubert's Symphony No. 9 (D 944), IV, mm. 543-558

3 Requisite Morphisms

In the problem above we are trying to discover a black box that takes audio as input and returns a rating. Let us now take a step back and look at the transformations that will lead to a tractable problem that can be solved using SVM classification.

The raw audio data is in the form of the multitrack stems from a synthesized orchestral excerpt. All the audio is of the same length in samples, the ensemble of voices having essentially the form of a matrix $\mathbf{R}$ with dimensions V voices and L samples. One of these voices is the click track, a metronome indicating the downbeat of each measure. It is a very simple synthesized pulse and is perfect to use as a clock and to divide the audio, form a data structure the author has termed the *raw score*

$$\mathbf{S} = \begin{bmatrix} u_{1,1}, & u_{1,2} & \dots & u_{1,M} \\ u_{2,1}, & u_{2,2} & \dots & u_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ u_{V,1}, & u_{V,2} & \dots & u_{V,M} \end{bmatrix}.$$

Each cell in $\mathbf{S}$ is itself an audio vector for voice v that starts on the indicated downbeat m and ends at the next downbeat $m+1$.

3.1 Audio Feature Extraction

Classifying raw audio is a notoriously difficult task in that it requires an enormous number of examples and a huge amount of computing power. While reserving time on powerful computers is not an issue, there are only around ∼50 excerpts. As a first step towards our eventual goal we introduce an extraction stage

$$f : \text{audio} \rightarrow \text{singular features}$$

that operates audio to extract relevant musical features. These include some which are symbolic in nature and some which capture signal or perceptual aspects of the audio. The symbolic fea-

tures include note onset times and note pitch. Of the other variety, spectral envelope, amplitude envelope, harmonic (partial) harmonicity and vibrato control signal are a few.

Using this feature extraction process a new data structure may be formed, the *feature score*

$$\mathbf{F} = \begin{bmatrix} f(u_{1,1}), & f(u_{1,2}) & \dots & f(u_{1,M}) \\ f(u_{2,1}), & f(u_{2,2}) & \dots & f(u_{2,M}) \\ \vdots & \vdots & \ddots & \vdots \\ f(u_{V,1}), & f(u_{V,2}) & \dots & f(u_{V,M}) \end{bmatrix}.$$

In each cell is the ensemble of features describing the behavior of voice v over measure m.

3.2 Pairs of Voices

From our annotations we know that we are not classifying audio per se but pairs of voices over the duration of one measure. We are thus not looking to classify vectors in our audio feature space but rather looking to classify vectors in a feature space derived from the pair-wise combination of feature vectors.

First we must form the pairs of concurrent voices. These pairs have number

$$P = \sum_{n=1}^{V-1} n.$$

This permits us to form the *pair score* with dimension P pairs $\times$ M measures.

$$\mathbf{P} = \begin{bmatrix} (f(u_{1,1}), f(u_{2,1})) & (f(u_{1,2}), f(u_{2,2})) & \dots & (f(u_{1,M}), f(u_{2,M})) \\ (f(u_{2,1}), f(u_{3,1})) & (f(u_{2,2}), f(u_{3,2})) & \dots & (f(u_{2,M}), f(u_{3,M})) \\ \vdots & \vdots & \ddots & \vdots \\ (f(u_{V-1,1}), f(u_{V,1})) & (f(u_{V-1,2}), f(u_{V,2})) & \dots & (f(u_{V-1,M}), f(u_{V,M})) \end{bmatrix}$$

As an example we can consider a string quartet. The pair matrix will consist of column vectors for each measure of the six pairs: (violin 1, violin 2), (violin 1, viola), (violin 1, cello), (violin 2, viola), (violin 2, cello), (viola, cello).

3.3 Relative Features

At this juncture we again run into the issue of feature space that is too large to use inference based learning methods, as our excerpt set is small. At this stage we construct a process that aims at extracting the meaningful interaction between voices. The ensemble $f(u_{v,m})$ contains features describing the musical behavior of a single voice. It is necessary to introduce a process

$$p\text{: singular features} \times \text{singular features} \to \text{relative features}$$

that is able to extract metrics of the following: harmonic (tonal) relationships between voices, the presence of parallel motion, vibrato synchrony, harmonic (partial) overlap, and more as they are deemed productive to the task.

This process is used to modify the pair score to generate the *pair feature score*

$$\tilde{\mathbf{P}} = \begin{bmatrix} p(f(u_{1,1}), f(u_{2,1})) & p(f(u_{1,2}), f(u_{2,2})) & \dots & p(f(u_{1,M}), f(u_{2,M})) \\ p(f(u_{2,1}), f(u_{3,1})) & p(f(u_{2,2}), f(u_{3,2})) & \dots & p(f(u_{2,M}), f(u_{3,M})) \\ \vdots & \vdots & \ddots & \vdots \\ p(f(u_{V-1,1}), f(u_{V,1})) & p(f(u_{V-1,2}), f(u_{V,2})) & \dots & p(f(u_{V-1,M}), f(u_{V,M})) \end{bmatrix}$$

This is the final form used to train the SVM classifier.

3.3.1 Pair-wise Pitch Harmonicity

As an example of one subprocess in the pair-wise process p, the ranking of pitch (tonal) harmonicity is described. The *MIR Toolbox* is a useful collection of tools for musical feature extraction [Lartillot et al., 2008]. Using the `mirframe()` decomposition object and `mirpitch()`, one or more pitch values can be assigned to each hop of the short time decomposition. Performing this on two concurrent monophonic voices results in two pitch tracking vectors of the same length v_1 and v_2. The total number of frames is the integer value T. The values of these vectors are rescaled and quantized to the nearest MIDI pitch.

The harmonic relationship between pitches is found in their ratio: the octave in powers of 2, the fifth in powers of 3/2. Because the MIDI scale is a logarithmic, the difference between pitches encodes their ratio:

$$r = \texttt{freq2midi}(v_1) - \texttt{freq2midi}(v_2).$$

The vector $r \in \mathbf{Z}^T$ will contain integer values that describe the intervals for each frame. What the author is proposing to do is to allocate elements of the relative feature set to holding the percentage of an interval relationship relative to the total duration. For instance, if

$$v_1 = \texttt{[440, 440, 587, 587, 880]}$$
$$v_2 = \texttt{[880, 880, 987, 1046, 1174]}$$

Then after converting to MIDI, the relation vector r takes the value

$$r = \texttt{[-12, -12, -9, -7, -5]}$$

Let us allocate for each pair 25 values, from an octave below to an octave above including unison. Here $T = 5$, and the harmonicity (tonal) relative feature is

$$p_{\text{tonal}} = \texttt{[.4, 0, 0, .2, 0, .2, 0, .2, 0, ..., 0]}.$$

We may find that classification will make use the dimensions representing perfect intervals to indicate blend and the rest to indicate the lack thereof.

3.4 Ground Truth

The ground truth data must be transformed from the Excel documents into a matrix $\mathbf{G}$ that matches the dimensions of $\mathbf{P}$. Each cell in $\mathbf{G}$ will take either the value 1 if the voice pair and measure are blending or -1 in the case of a non-blend pair. Blend as can be observed from the annotations is sparse in the space of pairs, as most pairs of voices do not blend.

3.5 Overview

These transformations of data begins with raw audio multitracks in the matrix $\mathbf{R}$. This is then divided into measures forming the raw score $\mathbf{S}$ that is still just audio but subdivided. The features are extracted from each cell within $\mathbf{S}$ to generate the singular feature matrix $\mathbf{F}$. Pairs of features form $\mathbf{P}$ and their features populate the final form $\tilde{\mathbf{P}}$.

$$\mathbf{R} \to \mathbf{S} \to \mathbf{F} \to \mathbf{P} \to \tilde{\mathbf{P}}$$

Ground truth is held in the matrix $\mathbf{G} \in \{1, -1\}^{P \times M}$.

4 Classification

These transformations outlined above allow us to leverage the power of statistical learning to classify pairs of measures into the categories *blend* or *nonblend*. To do this we vectorize the cells of the matrix $\tilde{\mathbf{P}}$ and $\mathbf{G}$ similarly and use this to train an SVM model. In MATLAB we will use `fitcsvm(...)`. The use of a necessary kernel, linear, polynomial or Gaussian, will warp the relative feature space forcing the training data to linear separation.

References

Lartillot, O., Toiviainen, P., and Eerola, T. (2008). A Matlab Toolbox for Music Information Retrieval. In Preisach, C., Burkhardt, H., Schmidt-Thieme, L., and Decker, R., editors, *Data Analysis, Machine Learning and Applications*, pages 261–268. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Studies in Classification, Data Analysis, and Knowledge Organization.