Separating Advertisements and DJ Chatter from Music

Bryan Klingner

Problem:

Radio Advertisements are Annoying

Broadcast radio offers an appealing way to enjoy music. Devices used to access FM radio are ubiquitous in modern life: cars, portable music players, and home stereos almost universally contain the hardware necessary to tune in. Usually, there are at least a few and sometimes dozens of stations available, offering different genres of music and talk.

Because of the funding model of radio, advertisements appear frequently between periods of music. Listeners often find themselves scanning through channels looking for one that's actually playing music. Sometimes, this effort is undertaken while driving and can be dangerous. **What if this job could be performed by the radio instead?**

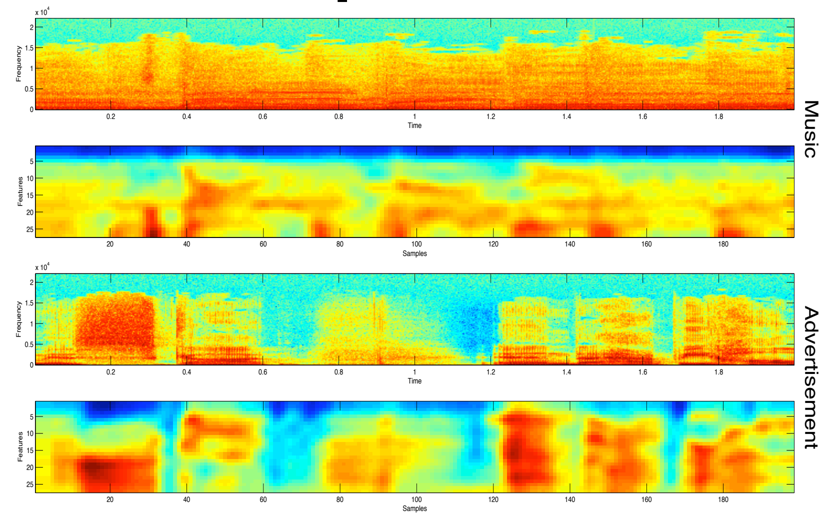
Goal:

Automatically Recognize Ads and DJ Chatter

We attempt automatically separate audio from radio into two classes: **music** and **non-music**. Non-music audio includes advertisements and DJ chatter. With a robust method for differentiating these two classes of audio, one could imagine a completely automatic system built into the radio of a car that would always keep the radio tuned to a station that was playing music.
We perform this classification of audio in two steps. First, we **extract salient features** from clips of music. Then, we train a **support vector machine** to classify novel audio clips as it receives them.

Approach

Step 1: Feature Extraction

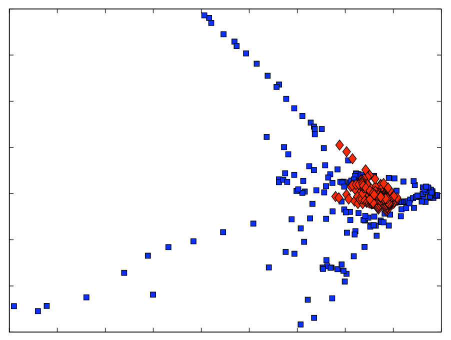


Top: the **spectral intensity** and **extracted features** for a two-second clip of music. Bottom: the same data from a two-second clip of a radio advertisement.

A listing of example audio clips. The first set of audio clips (ending in 1) were used for training, while the second (ending in 2) were used for testing.

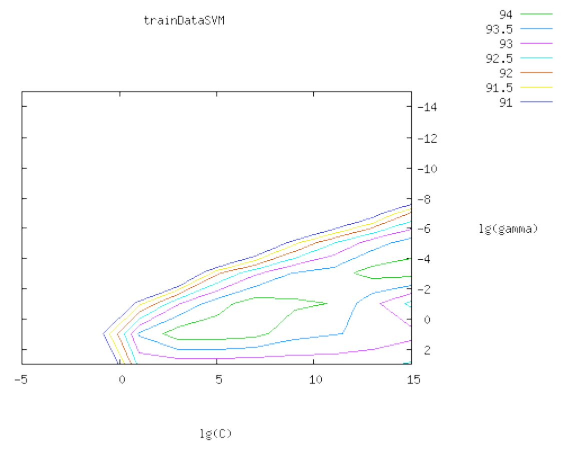
Name	Class	Length	Description
deathcabforcutie[1,2].wav	music	2 sec	Male vocals, electric.
decemberists[1,2].wav	music	2 sec	Mixed vocals, aucoustic.
franzferdinand[1,2].wav	music	2 sec	Male vocals, guitars, drums.
loscampesinos[1,2].wav	music	2 sec	Punk with male vocals.
annuals[1,2].wav	music	2 sec	Electronic and rock, mixed vocals.
whitestripes[1,2].wav	music	2 sec	Male vocals, electric.
yolatengo[1,2].wav	music	2 sec	Male vocals, electric.
adairamerica[1,2].wav	non-music	2 sec	Ad with background music.
adbkgnd[1,2].wav	non-music	2 sec	Ad with loud background music.
adpeacecorps[1,2].wav	non-music	4 sec	Ad with background music.
adnpr[1,2].wav	non-music	2 sec	Ad with background music.
djlowvoice[1,2].wav	non-music	2 sec	DJ chatter with music.
djnobackground[1,2].wav	non-music	2 sec	DJ chatter, no music.
nprnews[1,2].wav	non-music	4 sec	News report, no music.

Step 2: SVM Classification



We performed a **principle component analysis** to find the two most significant dimensions of variation in the data. We then projected the data down onto this reduced basis in to determine that we should use a **radial basis function kernel SVM**

We used a grid search to determine **optimal SVM kernel parameters**. A plot of isocontours of cross-validation accuracy for different values of the RBF kernel parameters C and gamma is to the right.



Results

Classification Accuracy: 87%

Overall, our classifier was quite successful. For the training and testing sets used, it correctly classified music and non-music data **88.6%** of the time. The optimization concluded within about 11,000 iterations and included 1365 total support vectors. It correctly predicted the class of **3690 of 4172** examples.

Running times were quite manageable. On a four-processor Power Mac G5 2.5GHz, the grid search for **parameter optimization** broadly dominated the running time at about **10.2** minutes. Once parameters were selected, the **training of the SVM took 2.90 seconds**, and **classification of the 4172 examples took just 1.04 seconds**.

Future Work

-Vastly increase the size of the sample set.

-Use audio features tailored to music, or at least more general than those designed for speech recognition.