Towards a Machine Learning DJ: First Experiments

Gary Holness Kimberly Martin

Computer Science Department
140 Governor's Drive
University of Massachusetts
Amherst, MA 01003-4601

Technical Report TR-04-01
January 2004

1 Introduction

Classification techniques have been applied to real world problems such as fish classification and email sorting. In this work, we introduce a new application called ANIMAL. ANIMAL is a Machine Learning Disc Jockey. A model for beat (from music) and bop (from head motion) is proposed. Using this model, we treat a listener's musical enjoyment as a classification problem. We define a beat/bop similarity metric based on harmonic matching over frequencies in the Fourier domain across the raw inputs (windowed proportionally to heterogeneous sampling rates, uncovered empirically). From our similarity metric, we define features which we use in a number of classification methods. We use both generative and discriminative methods such as Logistic Regression, Naive Bayes, Stochastic Gradient Linear Regression and Support Vector Machines (SVMs).We have results for a test set comprised of a 33% set asside from our corpus of data. We compare the performance of these methods among different sets of features extracted from the harmonic match. Our results show that the Naive Bayes Classifier outperforms the aforementioned classification techniques.

2 Learning Problem

A Disc Jockey (DJ) is responsible for musical selection at social gatherings. A key skill for a DJ is the ability to access the listening audience's satisfaction through visual queues. This knowledge is employed in the control decision of selecting the next song. We are given a camera trained on a subject and musical source (figure 1). The goal of this system is to extract visual queues from the camera and make a determination as to the subject's satisfaction with the current musical selection. In this experiment, we perform motion segmentation and track the subject's head and

compute the magnitude head bop velocity. Head bop velocity is used in our determination of listener satisfaction. This is a two class problem, where our classes are *Enjoy* and *Not-Enjoy*.

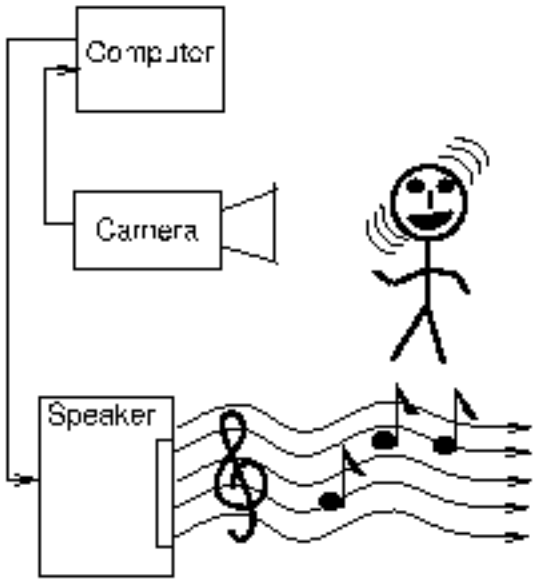

Figure 1: System Overview: Machine Learning DJ

More formally, we are given an input audio stream (a), for which we define a model $f(a)$ which characterizes the music's beat. Associated with beat is an underlying noise process ϵ_a. We define our full beat model as a family $A = \{f_i(a) + \epsilon_a,\ i = 0, 1, ..., m\}$. Similarly, given an input visual stream (v), we define a model $g(v)$ which characterizes the motion queues for the listener. Associated with $g(v)$ is a noise process ϵ_v. We define the full bop model as the family $V = \{g_i(v) + \epsilon_v,\ i = 0, 1, ..., n\}$.

We define a third process, H, which which relates A and V. Using features computed from this process, we learn the class label of each audio-visual pair.

3 Feature Representation

A Fast Fourier Transform (FFT) and power spectrum are computed over a 1 second frame of audio and head bop signals. We call a single such power spectrum for audio 1 beat. Likewise, for head bop, we call a single power spectrum 1 bop. Borrowing from the bag-of-words model in Information Retrieval, we model a song as a bag of beats and the listener's head bop sequence during the song as bag of bops. For each beat/bop pair associated with a song, we compute a distance metric, the *harmonic match* between a beat and its corresponding bop. Features in our model are defined in terms of the sequence of harmonic match numbers.

3.1 Spectral Information

The head-bop velocity and audio data describe time varying signals. Fourier analysis tells us that such signals can be approximated by a weighted sum of sinusoidal basis vectors. The Fourier transform gives us this estimate.

Given a time varying continuous signal $f(t)$ the Fourier transform is...

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$$

For the discrete case, we have a window of N samples taken at a rate f_s from a time varying function $x(t)$ and compute the Fast Fourier Transform (FFT)...

$$X(k) = \sum_{n=1}^{N} x(n)e^{-j(\frac{2\pi}{N})nkf_s}$$

This is used in a related statistical approach. The cross-correlation between a signals $x(t)$ and $y(t)$ at different times is...

$$R_{xy}(m) = E\{x_{n+m}y_n\}$$

for $-\infty < n < \infty$ and the covariance between $x(t)$ and $y(t)$ is...

$$C_{xy}(m) = E\{(x_{n+m} - \mu_x)(y_n - \mu_y)\}$$

When $x = y$ we get autocorrelation and variance respectively. Given a discrete sample window, we compute an estimator for R as...

$$\hat{R}_{xy}(m) = \sum_{n=1}^{N-m-1} x_{n+m}y_n$$

With spectral analysis, we describe how energy in a time varying periodic signal is distributed across sinusoidal basis vectors. The power spectrum is essentially the FFT of the cross-correlation sequence...

$$S_{xy}(\omega) = \sum_{k=-\infty}^{\infty} R_{xy}(m)e^{-j\omega m}$$

Given that $\omega = 2\pi\frac{f}{f_s}$ we write...

$S_{xy}(f) = \sum_{k=-\infty}^{\infty} R_{xy}(m) e^{-2\pi \frac{jfm}{f_s}}$

Averaging over f_s we get...

$P_{xy}(f) = \frac{S_{xy}(f)}{f_s}$

Which is the expression for power spectral density of a signal x when $x = y$ and cross power spectral density when $x \neq y$. We can estimate P using an FFT over a window of samples of length N. By this approach, the energy density at frequency f is approximated by the magnitude squared energy at f divided by the normalized sampling rate...

$\hat{P}_{xx}(f) = \frac{|X_N(f)|^2}{f_s N}$

Note here that $X_L(f) = \sum_{n=1}^{N} x(n) e^{-2\pi j \frac{fn}{f_s}}$ is the FFT of $x(t)$ over a window of length N. Thus, the power spectrum density estimator is approximated by the normalized squared energy at frequency f. Given two different signals $x(t)$ and $y(t)$, we measure the correlation between two signals at a frequency f. This quantity is called coherence...

$C_{xy}(f) = \frac{|S_{xy}(f)|^2}{S_{xx}(f) S_{yy}(f)}$

3.2 Harmonic Match

For our learning problem, we wanted a feature space which adequately presented aspects of how the listener's head bop coincides with a piece of music's rhythm. Some observations we made were...

- head bop motion is limited to relatively low frequencies
- head bop motion may change during the course of a song
- a musical selection contains frequencies across the audible spectrum
- the low notes (bass, drums) typically guide the beat in modern popular western music
- these low notes are found at lower frequencies

Our main thesis is that frequencies containing energy within a bop can be found in a beat either in their natural form or as some harmonic. We designed a function which computes a number representing the degree to which head bop motion coincides with musical beat. This function rewards for low frequency coincidence between head bop and musical beat. Our function also considers harmonics of head bop contained within the musical signal. In [Durand] they employ a two way matching scheme between a predicted and measured harmonic series. We independently designed a technique with many similarities. We employ a one-way matching scheme where the PSD estimate for head bop signal frame is matched to the PSD estimate for an audio signal frame. Our matching

scheme accounts for minor local variations by admitting matches within an ϵ. In addition to matching natural frequencies in the head bop PSD, we also match the first 100 harmonics. We define harmonic match for a synchronized time step t within the head bop $x(t)$ and audio $y(t)$ signals as...

$$\mathcal{H}_t = \sum_{i=1}^{s} \sum_{h=1}^{k} \frac{1}{h}[\hat{P}_{xx}(f_{i,t})\hat{P}_{yy}(\mathcal{F}_h(f_{i,t}))]C_{xy}(f_{i,t})$$

Where $\frac{1}{h}$ is the h-th element in the harmonic sum $H_n = \sum_{h=1}^{\infty} \frac{1}{h}$ and $\mathcal{F}_h(f_{i,t})$ is the h-th harmonic of $f_{i,t}$ frequency i and time step t. The following are example outputs of PSD for head bop, audio signal, coherence, and harmonic match for the song *One Love* by Bob Marley.

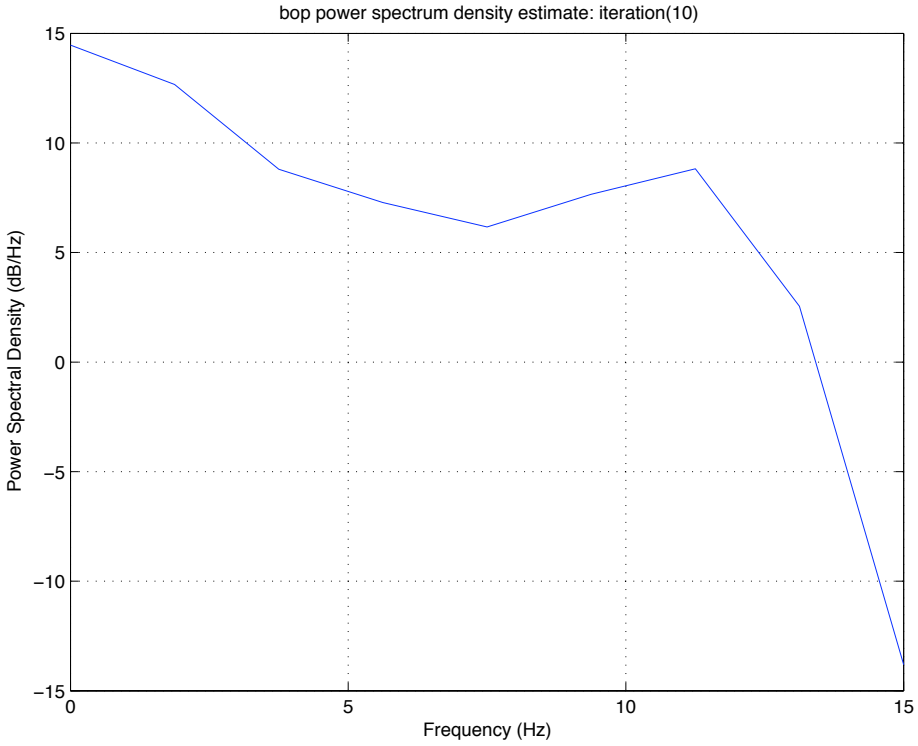

Figure 2: PSD for headbop motion

3.3 Feature Selection

We divided the audio signal into units of 2^{15} samples and the head bop signal into units of 10 samples. For each window, we performed a match. This resulted in a sequence of match iterations $\mathcal{H}_1, \ldots, \mathcal{H}_{45}$. Our feature space is infinite as it is the set of all possible statistics over the 45 element match sequence. The features we chose are...

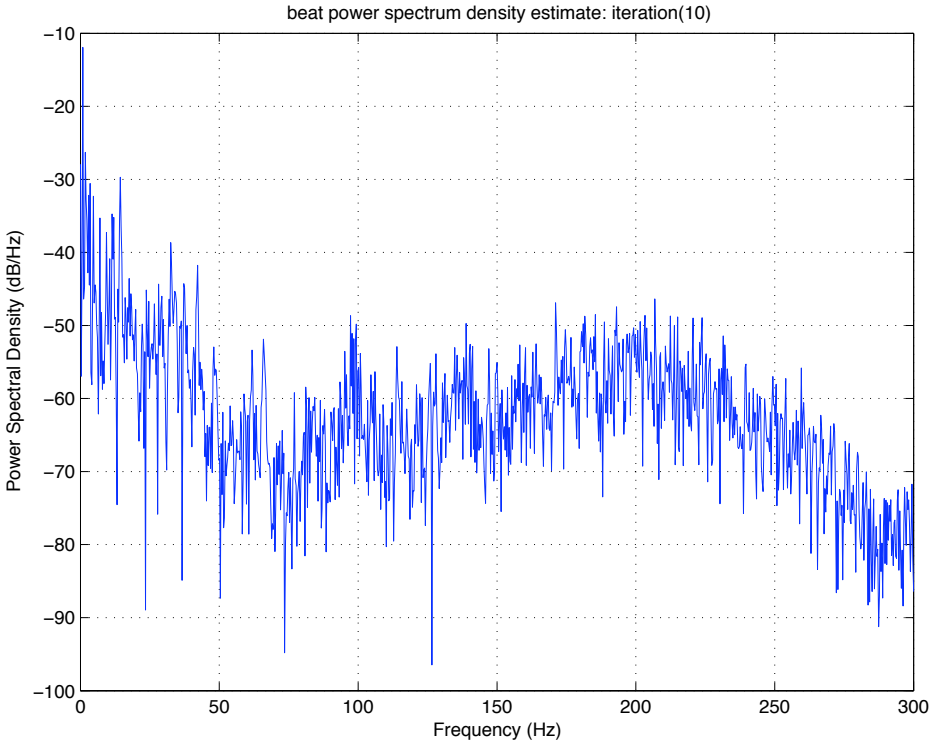

Figure 3: PSD for audio signal

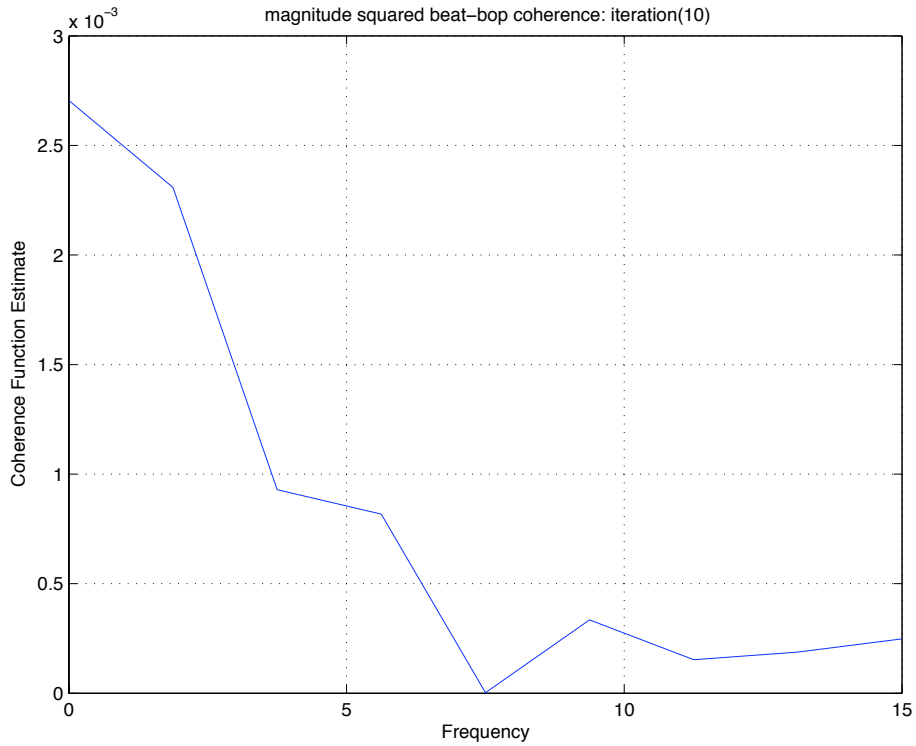

Figure 4: Coherence for beat-bop

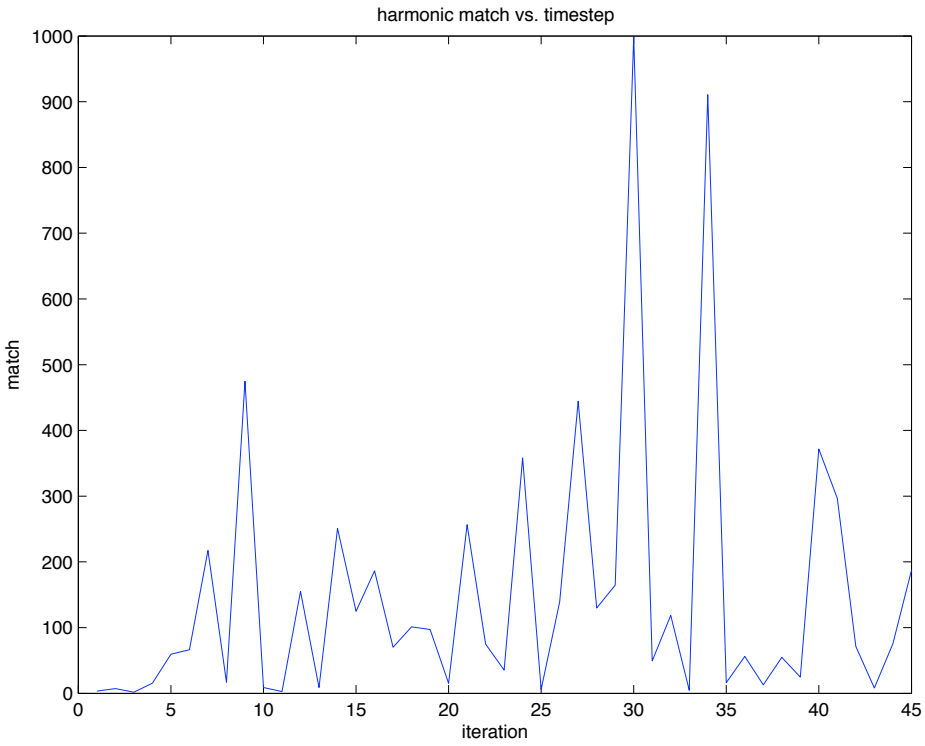

Figure 5: match $\mathcal{H}_t$

- the order statistics $X_{i:N}$
- the median
- the min
- the max
- the range
- the 1st moment: mean
- the 2nd moment: variance
- the 3rd moment: skew
- the 4th moment: kurtosis
- the number of positive velocities (single step slope)
- the number of positive accelerations (single step velocity slope)
- the number of peaks (inflection points)

There were many more statistics over the match numbers we could have considered. The 56 we chose we felt would be features which captured relevant information about harmonic match.

4 Results

The following table summarizes our results using 62 and 99 samples from our data set.

	62 Samples	**99 Samples**
Method	**Average Accuracy**	**Average Accuracy**
Naive Bayes	72.0930%	62.79%
Binomial GLM (link=logit)	54.762%	48.485%
Binomial GLM (link=probit)	59.524%	51.515%
Gaussian GLM	52.381%	46.967%
Stochastic Gradient	52.6%	42.424%

For Stochastic Gradient, initially we used all 56 features in our model using using a cooling schedule for α, the learning rate, of $\alpha_t = \frac{1}{t}$ where t is the time-step which describes the update iteration. An iteration is an update of all the positions in the coefficient weight matrix. After $4.5 million$ iterations and 1.5 days approaching convergence, we halted our stochastic gradient computations and decided to change some parameters to get faster convergence. In addition to the cooling schedule for α, another parameter is the convergence threshold. This parameter controls the stopping criteria. In Stochastic gradient, in a single iteration, we update all the weights with $\beta_j = \beta_j + \alpha_t(y^i - P(x|\beta))x_j^i$. That is, the update for coefficient β_j is some proportion, α_t of the error between the model and the predicted output modulated by the j-th feature's value for the current i-th sample. The convergence threshold is the value which maximum update among all the β_j's must not exceed in order for Stochastic Gradient to be considered converged.

For the 62 sample result for Stochastic Gradient, we adjusted the cooling schedule to $\alpha_t = \frac{2}{t}$. For the 99 sample result for Stochastic Gradient the new cooling schedule did not speed convergence. After 2 days and 6.5million iterations, we halted this computation. After a number of experiments, we found that removing the feature for variance over the set of harmonic numbers $\mathcal{H}_1, \ldots, \mathcal{H}_{45}$, keeping $\alpha_t = \frac{1}{t}$, and increasing the convergence threshold from $\delta = 0.1$ to $\delta = 0.5$ reduced convergence to 335105 iterations. The variance feature was removed because it's β update was very far away from convergence after 6.5 million iterations while all the other β updates had converged.

5 Conclusions

This first experiment has yielded satisfying results, but there are many areas where the system can be improved.

There were many factors contributing to variance in our data. There is noise in the sensor elements in the camera as well as in the resulting image processing. This noise contributes to variability in the head bop data. Additionally, listener fatigue while recording the head bop data meant that we had variability in head bop data. Likewise, habituation contributed to variability in head bop data. We chose music belonging to 3 genre's: techno, slow-music, hip-hop. By habituation, we mean that a listener gets bored after listening to the same kind of music for hours. This also contributes to variability in head bop data. A subjects level of musical training also introduces variability in the

head bop data. A listener who was musically trained or played an instrument, is more likely to follow a standard set of measures (e.g. $\frac{4}{4}$th) when bopping their head. This would tend to bias our head bop data to one of the standard musical measures.

The audio data was digitally recorded music in WAV format and, thus, variability in audio was not much of an issue. But, in a deployed system we would use microphones to provide the audio signal. This sensory mode would introduce variability in the audio data.

We performed versions of our experiment using a harmonic match function that considered the first 10,15,and 100 harmonics. We found that the version what considered the first 100 harmonics produced the most stable results which we present in this report.

Based on our results, we conclude that the amount of noise in the data overwhelm any covariance between features which might be exploited in our models. By making the independence assumption between samples, Naive Bayes essentially throws away the covariance matrix. Naive Bayes outperforms the other classifiers because it assumes that the covariance matrix is identity. Also, from the results you can see that the SVMs performed miserably having found no support vectors. This because the variance in data for the size of the corpus we created was very high. SVM was unable to find any support vectors onto which to project test instances in our model for the DJ task. We attempted SVM using a linear kernel and polynomial kernels of degree two and three. For future experiments we plan on using SVM with a radial basis kernel.

In making our bag-of-beats and bag-of-bops assumption, we essentially ignore possible sequencing in the audio and head motion data. Incorporating syntactic information into our model would improve performance of models which make use of covariance matrix between features. A method such as PCA or LDA as a means of performing feature selection on the harmonic match numbers may have reduced our model complexity significantly.

6 Future Work

- Improvement to System - Create a more robust system for detecting head bop and a different method of processing the audio signal
- Create more data - Data creation is a very time consuming process, on the order of 4 - 6 minutes per sample including time to process both audio and video data. A sufficient corpus we anticipate would be roughly 1000 songs (10 genres, 100 per musical genre).
- Song Selection (DJ) - Once the user is classified as enjoying the music, once the music ends, select a song within the same genre, assuming the user likes songs that are similar. If the user is classified as not enjoying the music, select the next song from a different genre.
- Mixing (DJ) - Mix the sound so that the transition between songs is seamless.
- Real-Time Performance - Right now the system is made for pre-processing the audio signal, but once the accuracy rate is improved, this can be extended to real-time audio processing.
- Multiple Subjects - Currently the system is designed to detect one person bopping their head, but should be extended to multiple person detection in a party scene or other.

- Learn Preferences : Generate New Music - If we can classify features of the music that a person enjoys, then are we able to create music that this same person would enjoy?

References

Durand DURAND and GOMEZ, Periodicity Analysis using a Harmonic Matching Method and Bandwise Processing *Music Technology Group, Pompeu Fabra University*

Gibson JERRY GIBSON, Principles of Digital and Analog Communications *Macmillan Publishing Company, New York*, 1989

Bain1991 BAIN and ENGELHARDT, Introduction to Probability and Mathematical Statistics, *Duxbury Classic Series*, 1991.

Barber2002 BARBER, DAVID, Learning from Data 1: Logistic Regression *http://anc.ed.ac.uk/ dbarber/lfd1/lfd1.html*, 2002

Cristianini CRISTIANINI and SHAWE-TAYLOR, Support Vector Machines and other kernel-based learning methods *Cambridge University Press*, 2000

Broersen BROERSEN and WAELE, Windowed Periodograms and Moving Average Models *Department of Applied Physics, Delft University of Technology, Delft, The Netherlands*

Lang LANG, KEN, NewsWeeder: Learning to filter netnews *Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufman Publishers, San Mateo, California*, 1995

MATLAB MATHWORKS Matlab 13 *http://www.mathworks.com*