Chord Recognition

Jonathan Ward

The task of this assignment was to improve the accuracy of a Hidden Markov Model chord recognition program. With a starting accuracy of 48.2% on chord identification in a selection of songs, there was much room for improvement. I was aiming for 85+% accuracy, but was only able to achieve a maximum of 76.4% after trying a number of different techniques.

The biggest improvement in performance came from compressing the chroma features. Compression puts less emphasis on large chroma values, which are probably outliers, and can reduce the program's tendency towards erroneous matches. Raising each chroma value to the .15 power seemed to have the best results, around 75% accuracy. A normalization pass after compression improves the accuracy by a further 1%.

I also tried hard clipping the highest chroma values, but this didn't work nearly as well. For hard clipping, I tried various methods. One of the most promising algorithms involved clipping all the chromas that were a few standard deviations away from the mean. For example, if the chroma value was greater than two standard deviations away, I would change it to be just two standard deviations away. The rationale behind this method was that it would be easier to eliminate outliers if I used the standard deviation rather than straight compression.

```
AvgVals = mean(Chroma);
StdDev = std(Chroma);

for i = 1:size(Chroma,2)
     for j = 1:size(Chroma,1)
          if Chroma(j,i) > AvgVals(i) + 2 * StdDev(j)
               Chroma(j,i) = AvgVals(i) + 2 * StdDev(j);
          end
     end
end
```

Unfortunately, this did not turn out to be the case. Compression still produced better results. A possible reason for this is that hard clipping produces more distortion in the chroma features than compression. Clipping sets a hard limit for chroma values based on the total spread of chroma values. This assumes that high chroma values are outliers and are not significant. Since this clipping technique did not pan out as envisioned, this is probably a poor assumption. In contrast, compression reduces the high valued chromas, but their high value still retains some meaning. It just puts more emphasis on lower-valued chroma.

I also tried altering the transition matrix without much success. Tweaking the transition matrix modifies the way the Hidden Markov Model identifies chords. The transition matrix enumerates the probability of what the next chord will be, given that a certain chord is played. It is built during the learning stage. If these probabilities are altered, it may be possible to force the Hidden Markov Model to arrive at a better solution. Even after attempting to hand tune the transition matrix, no improvement was seen over just straight compression, even used in concert with compression.

I wanted to achieve greater than 85% accuracy, but was unable to attain that goal. Various

techniques were tried and ultimately 76.4% was the best I could squeeze out of the program. It is surprising that the best results came from chroma compression, the simplest technique. With better understanding of the Hidden Markov Model implementation, higher accuracy should be achievable, but I am sure that there is still some low-hanging fruit that I have yet to discover.