Hugh Zabriskie

Dylan Nagler’s music for CS people layout:

* pitch classes
* chords (collections of pitch classes)
* scales (collections of pitch classes)
* chord structures (triads, tonic, dominant, subdominant)
* chord progressions (voice leading)
* modulations (*with analogies to computer science*)
* models for Schubertian harmonic analysis (Schenkerian and *Ursatz*, Riemannian and *tonnetz*)

What can we assume computer scientists have a knowledge of?

What needs to be taught:

* Fundamentals of tonality
* Harmony (one-voice and multi-voice) and counterpoint
* Roman numeral analysis
* Chorales

What will make my introduction special:

* Introduce a new interesting theorists who have studied the chorales
* A discussion of the chorale and its structure
  + Divided into phrases by fermatas
  + We’re strictly looking at the 4-part ones
  + A history of the chorales (derived from where? Hymns?)
  + The soprano is the focus, supported by the other moving lines (describe counterpoint)
  + Why did I choose the chorales?
* **Why does this problem matter?**
* **Why the chorales? Why this data set?**

**OUTLINE**

1. The fundamentals of hierarchy in tonality
   1. Pitches
      1. 12 pitches
      2. An ordered series (stream) of pitches is a melody
      3. Define notes as pitches with duration
   2. Chords and scales
      1. Collections of pitches
      2. Interchangeable – one implies the other
   3. Harmony and counterpoint
      1. Counterpoint as the interaction between series of pitches

**I want to focus specifically on the relationships between melody and harmony.**

**I am trying to predict harmony based on melody. What I want to provoke first is how people have traditionally thought about the process – as predicting a *progression*. Examining the harmony we have at a certain point, what features explain it?**

Laitz’s Format

* Pitches and Pitch Classes (5)
  + 12 fundamental pitch classes
  + Pitch is pitch class + octave
  + Enharmonically equivalent pitches 🡪 same in sound, not function
* Scales (6)
  + “When pitches are hierarchically arranged in a diatonic scale and one pitch is stronger and more stable than all the others, this effect is referred to as a tonality.”
* Scale Degrees (7)
  + Scale – a series of half and whole step intervals (*my definition*)
    - Half steps (H, “semitones”) and whole steps (W, “tones”)
  + Tonic, dominant, subdominant, mediant, leading tone
  + Major, minor
    - Harmonic minor, Melodic minor
* Keys(8)
  + “Key signatures convey the pitch classes of major and natural minor scales” (8)
  + Defined by the half/whole series and a tonic pitch (my definition)
* Intervals (13)
  + Def: “distance between two pitches”
  + Size (unison – 7)
  + Quality (major, minor, augmented, diminished)
* Inversions (14)
  + Can move in either direction
  + Distance spanned by inversionally-related intervals is always 12 half-steps (the total number of pitch classes)
  + Enharmonic intervals, consonant and dissonant intervals (15)
  + Me: intervals represent the most basic form of counterpoint – one voice against another
* Metrical realm (19)
  + Time is divided into beats, or pulses.

Music to CS:

Focus on a few examples:

* find a canonical paper
* talk through a basic example (Dylan’s fermata prediction)
  + Define a distribution (i.e. multinomial)
  + Maximum likelihood criterion

Neural networks give us a way to understand deeper interactions

Recurrent networks – a section of the introduction, “strong empirical evidence” in the fields of NLP that they’re able to catch *long-term* phenomenon, which seems like a particularly good fit for music

Talk about RNN-music paper:

* What did they do well, what they didn’t do well
* Get mathematical

A primer on neural networks:

In machine learning, a subfield of artificial intelligence, we are concerned with the concept of learning from a dataset. Machine learning can be broadly divided into two categories, supervised and unsupervised learning, and the two have unique goals in learning. In supervised learning, a set of observations, referred to as the inputs, is correlated with a set of matching outcomes, or the outputs. Consider the task of learning to predict the price of a car given a set of the car’s features. In a “learn-by-example” manner, a machine is shown several cars and their known prices, and it eventually learn how to predict what a car’s price might be. If a linear regression is made between the model of the car and its price, then the

The maximum likelihood estimate is so called because it is the choice of parameter values which gives the highest probability to the training corpus. (198)

\begin{enumerate}

\item Talk in terms of statistics first (distribution)

\item We don't know these distributions, we want to learn to estimate them

\item Trying to learn a probability model

\item Every machine learning algorithm gives you a distribution

\item Neural network is a powerful way to do this

\item Use digit recognition as an example

\item RNNs: a vector (define that) that represents in low-dimensional space that represents the past history of decisions, $h\_i$ is the hidden state, $p\_i$ is the prediction, a logsoftmax to give you a distribution

\item TIKZ for figures (Sasha does it, that's just LaTeX), Omnigraffle

\item newcite, use "citet" to do things like "Menzel et. al. (1992)"

\item Take the max, estimate uncertainty, k max, etc.

\end{enumerate}