Using Machine Learning and Statistical Analysis to Make Musical Discoveries

Cory McKay

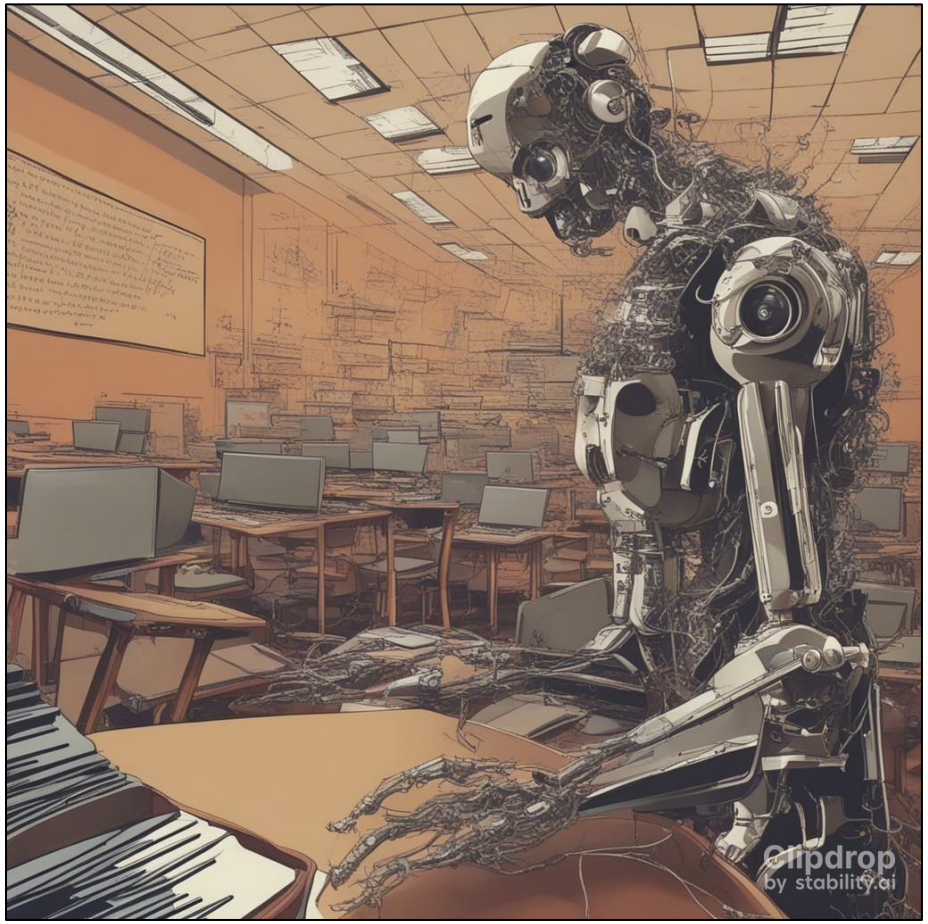
Marianopolis College, Canada

CIRMMT, Canada

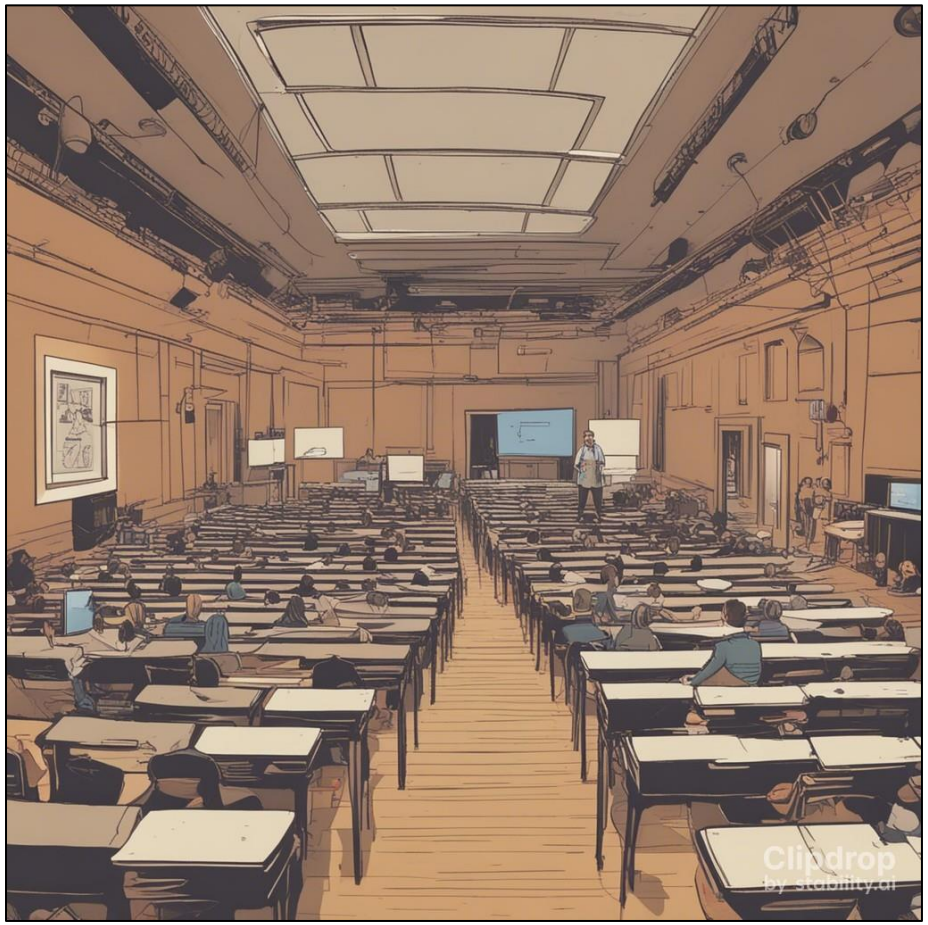
Topics

- Whirlwind tour of machine learning
- Statistically characterizing music
- Highlights from my own research
- Other musical applications of machine learning

“A brilliant speech on machine learning and music in a college”

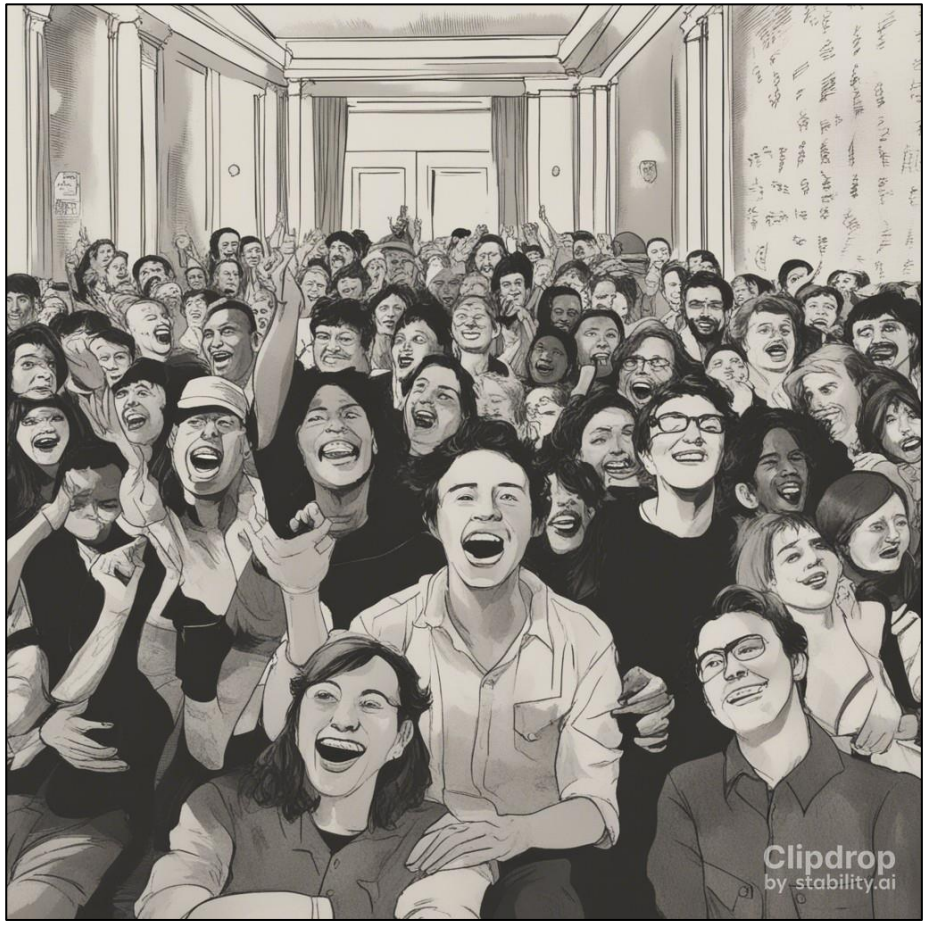


“A brilliant speech on machine learning and music in a college”



“A rapturous lecture audience after being gifted with erudition”

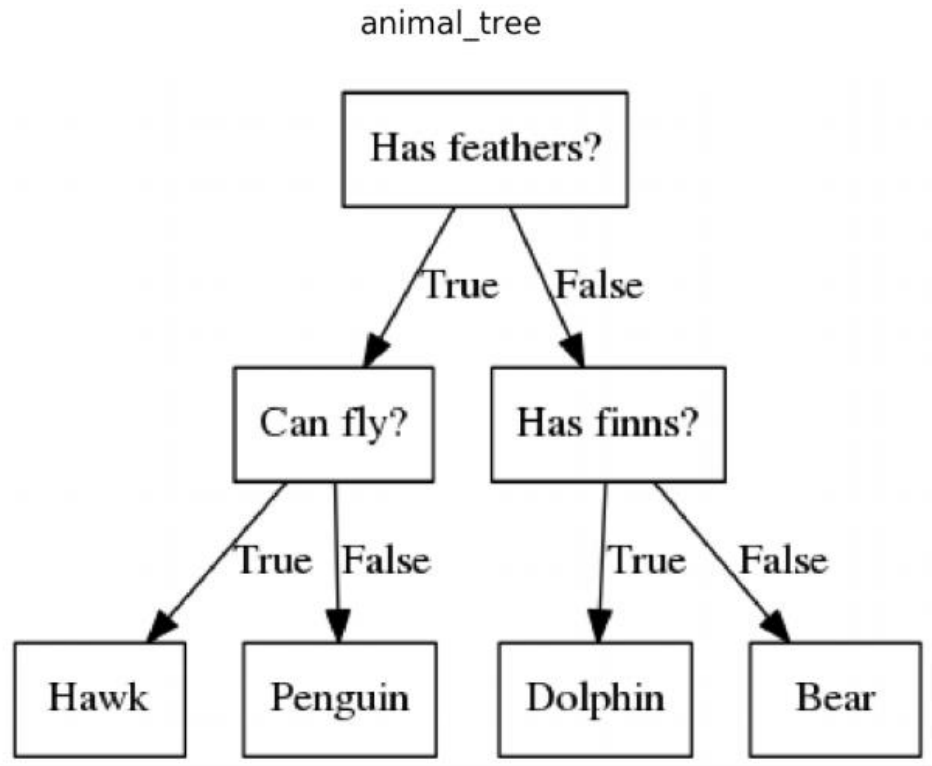
“An ecstatic college audience after attending the greatest music technology lecture of their lives”



Whirlwind tour of machine learning

Early AI: Expert systems

- Early AI focused on manually encoding sets of explicit rules ("heuristics") that a computer could follow
  - Often represented as "decision trees" (right)
- Early specialized programming languages like "Prolog" were used to specify logical rules and relationships



Early AI: Expert systems

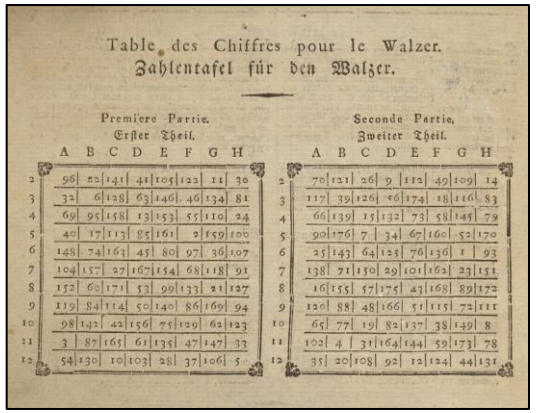
- "Expert systems" based on these kinds of approaches require:
  - "Knowledge representation" for encoding, connecting and structuring information in machine-interpretable ways
  - An "inference engine" for processing inputs in the context of embedded knowledge to produce outputs
    - Usually based on logical reasoning via explicitly defined rules
    - Sometimes involving some probabilistic reasoning (e.g. "fuzzy logic")

Early AI: Expert systems

- Tree search techniques like "branch and bound" can be used to explore as many possible outcomes of a decision as can be computed
  - Unpromising (or probably unpromising) outcomes can be eliminated when exhaustive computing is not possible
  - Often used in early game-playing AI (e.g. checkers)

Sample historical expert systems

- My own graduating undergraduate physics thesis project (1998, right) on using magnetic fields to help robots navigate unknown environments
- INTERNIST-I or MYCIN, prototype systems from the 1970's for performing clinical diagnoses
- Musical dice games (popular in the 18th century) can be seen as early generative expert systems
  - Music is composed by series of dice rolls that lead to lookups on pre-constructed tables
  - Such games have been attributed to composers like C.P.E. Bach, Mozart and others

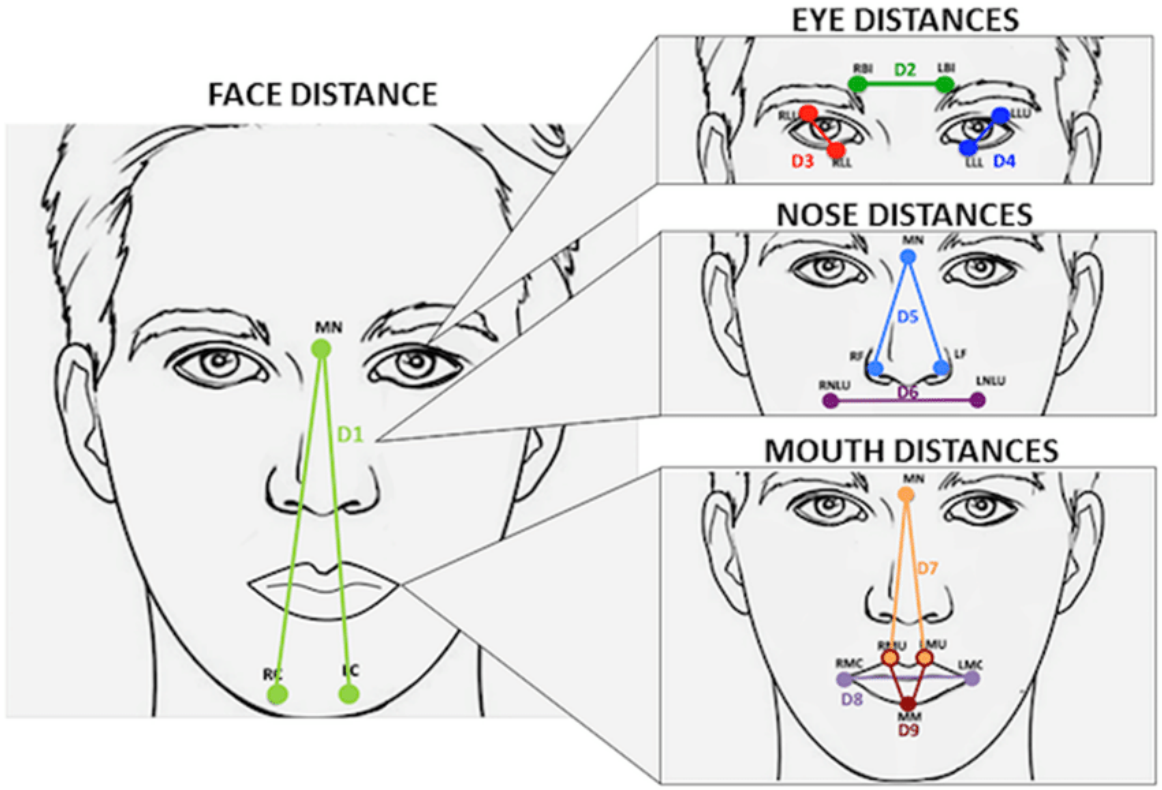


Machine learning

- Manually specifying heuristics can be onerous or impossible for complex problems
  - And is intrinsically flawed when expert domain knowledge is imperfect
- "Machine learning" bypasses this problem by having AIs automatically learn their own heuristics (or other ways of making decisions)
  - Training, testing and validation data is needed to train a "model" that can then be used to make decisions
    - "Generalization" beyond the training data must be verified
  - Traditionally (but much less so now), bespoke engineered "features" (characteristic measurements) expected to be useful were pre-calculated from the data and used as the percepts of the models

Machine learning

- Sample application: Facial expression recognition
  - In the past, pre-computed feature values rather than raw images were used to train models
    - e.g. distance between eyes
    - e.g. width of mouth
    - etc.



Machine learning

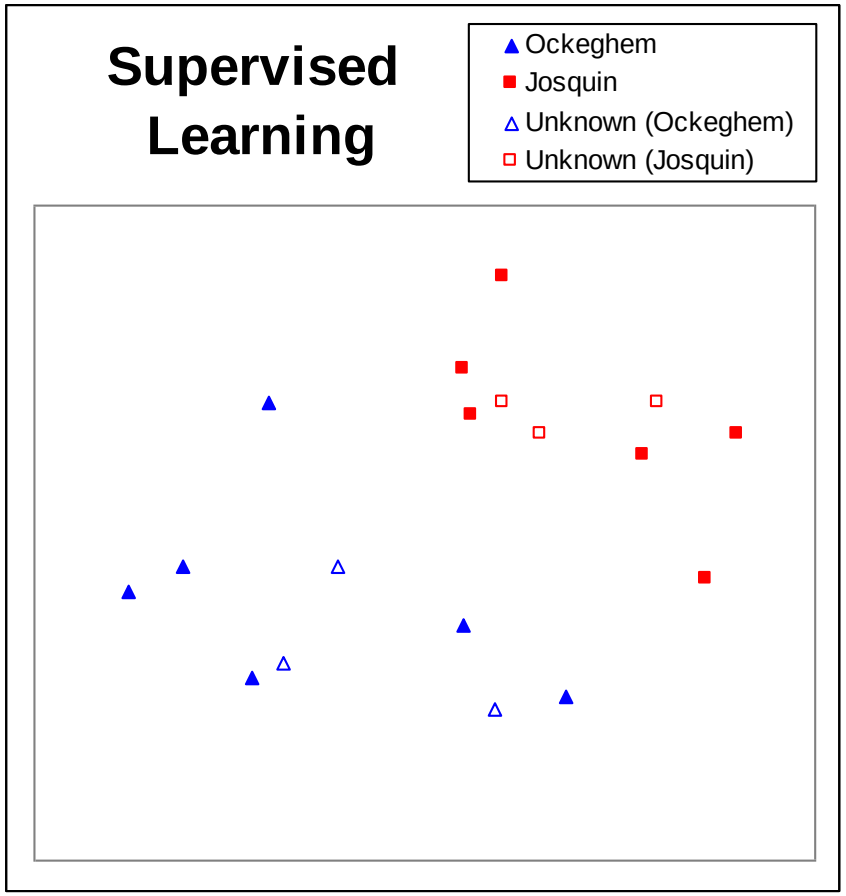
- Not having enough features, or not having a sufficiently complex model, risks "underfitting" complex problems
  - i.e. the model could be intrinsically incapable of representing a sufficiently sophisticated mapping, either because the available features are missing essential information or because the model is too simplistic
- Conversely, too many features or too complex a model risks "overfitting" training data
  - i.e. capturing patterns in the training data that are not characteristic of the overall population from which it is drawn
  - This is related to the "curse of dimensionality"
- More training data can counterbalance the curse of dimensionality
  - But more data can be expensive or impossible to obtain
- "Dimensionality reduction" methods can also help, by automatically pre-shrinking feature spaces to (hopefully) be more tractable and directly salient
  - e.g. PCA, forward-backward selection, genetic algorithms, etc.

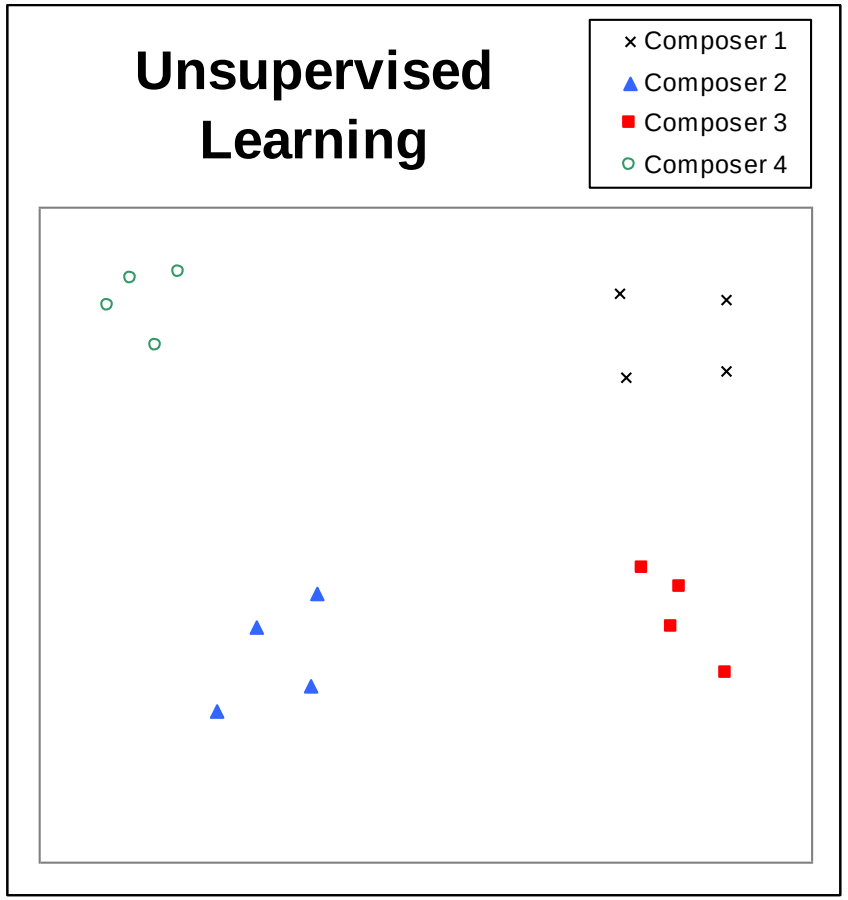
Machine learning

- Three particularly important overall types of machine learning:
  - Supervised learning: training instances are labelled with "classes", and during training models learn to map patterns in input instances (or their extracted feature values) to appropriate class name(s)
  - Unsupervised learning / clustering: training instances are not labelled, so during training models look for emergent patterns in the training instances (or their extracted feature values)
  - Reinforcement learning: an AI agent interacts with an environment and learns from feedback it receives from the environment
    - e.g., if a Roomba consistently bumps into an object in a certain place, it learns not to go there

Machine learning

- An illustration of supervised vs. unsupervised learning from a (2-feature) comparison from my own research on composer identification



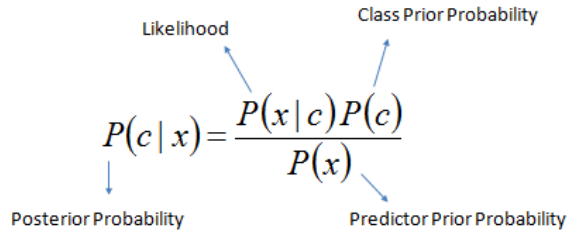


Classic machine learning algorithms

- There is a great variety of algorithm types that have been developed to train models, a small sampling of which is outlined in the following slides

Classic machine learning algorithms

- Naive Bayesian: uses Bayes' law, a basic principle in probability theory, to build models based on the observed relative frequencies of feature patterns



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- Mixture models: form mappings via weighted sums of specific statistical feature probability distributions (e.g. Gaussians), and learns by iteratively estimating the parameters (e.g. mean vectors, covariance matrixes) and relative weights of these distributions

Classic machine learning algorithms

- k-NN (k-Nearest Neighbor): a geometric approach that represents an input instance as a point in an n-dimensional feature space, and labels it based on the labels of the k closest memorized training instances

- Linear discriminant models: separate classes in n-dimensional feature space using hyperplane "discriminants" (decision boundaries separating classes)
- Support vector machines (SVMs): define discriminants using "support vectors"
  - Can use "kernels" to deal with data that is not linearly separable by projecting it to higher-dimensional spaces
  - Especially good (relatively speaking) at dealing with problems with many features and relatively little training data

Classic machine learning algorithms

- Decision tree learners: decision trees are inferred directly from feature data

- Ensemble learning: multiple models are trained and used together to arrive at final outputs
  - Individual base learners are often quite simple (e.g. decision tree “stubs”)
  - “Bagging” or “boosting” are common arrangements for combining base learners
  - “Adaboost” has been a particularly successful approach

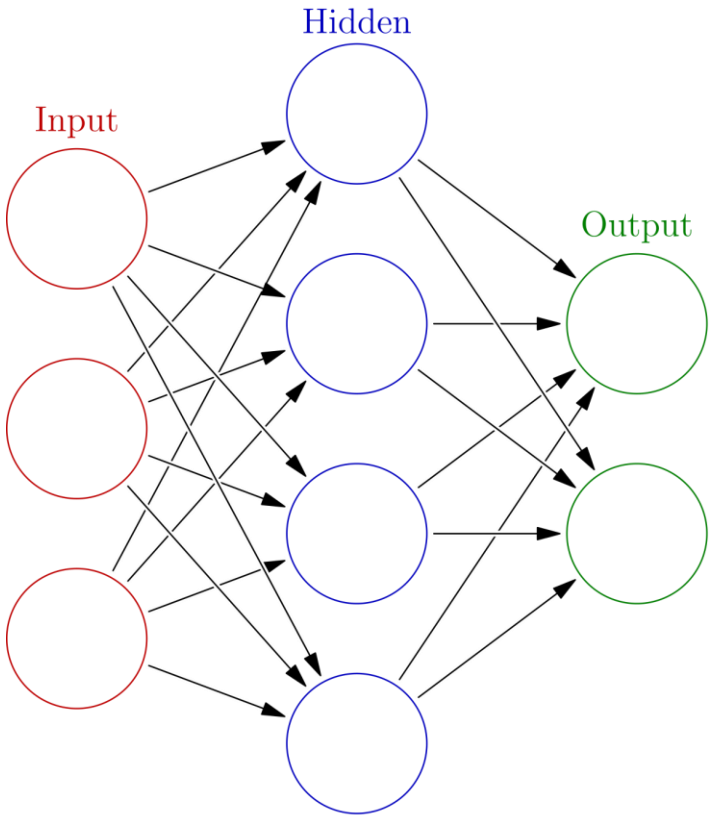
Classic machine learning algorithms

- Sequential learning: instead of treating instances of data independently, the sequence in which they occur is modelled in meaningful ways
  - e.g. processing melodies or chord progressions
  - Hidden Markov models (HMMs) and conditional random field classifiers are two well-known approaches

- Genetic algorithms: learning is performed by “evolving” sets of solutions to be optimized based on some measure of “fitness”

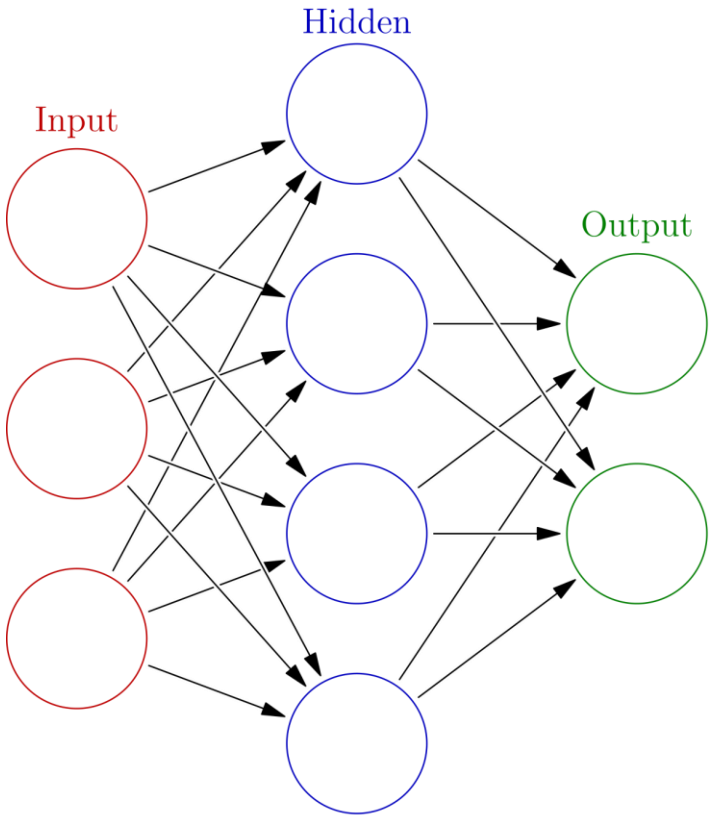
Artificial neural networks

- "Artificial neural networks" are a machine learning approach (very) loosely based on animal brains
- Sets of "nodes" are connected by "edges" and arranged in "layers" going left to right
  - The leftmost nodes receive input
    - Raw data, or features pre-extracted from it
  - The rightmost nodes specify output
    - e.g. classifications predicted by the network
  - Information can be iteratively passed through "hidden layers" of nodes separating the input and output nodes



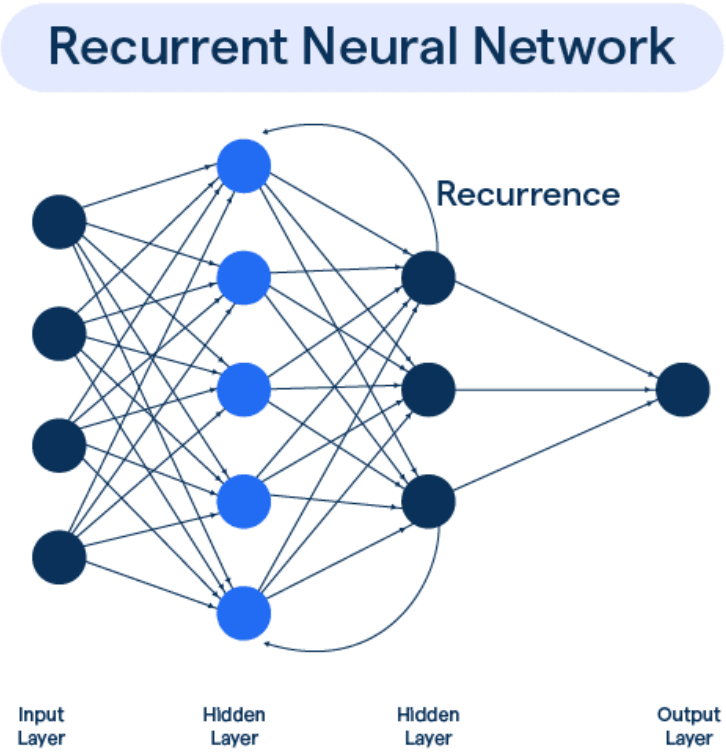
Artificial neural networks

- Each edge has a "weight" used to multiply values passed through it from one node to another
- The (weighted) sum of all inputs to any node (and its own constant "bias" value) is processed by an "activation function"
  - The result is then sent to nodes in the next layer of the network, via connecting edges
- Training happens by incrementally adjusting the weights of the edges and (sometimes) the bias values



Artificial neural networks

- Many variants of this basic neural net architecture exist
  - Some quite sophisticated
- For example, “recurrent” nets have feedback edges connecting nodes later in a network back to nodes in earlier layers
  - This architecture can model sequences, like melodies and chord progressions



Artificial neural networks

- Neural nets have waxed and waned in popularity over the decades
- Early neural nets ("perceptrons") lacked hidden layers, and researchers lost interest in them in 1969 when it was shown that they could not model certain essential functions, like the Boolean XOR operator
- Interest was rekindled in the 1980s when a training process called "backpropagation" permitted the incorporation of hidden layers, which in turn permitted essential functions like XOR to be modelled
- Although there were some initial success, limited training data and prohibitively long training times for even moderately complex models led neural nets to be neglected in favor of other approaches for many years

Deep learning

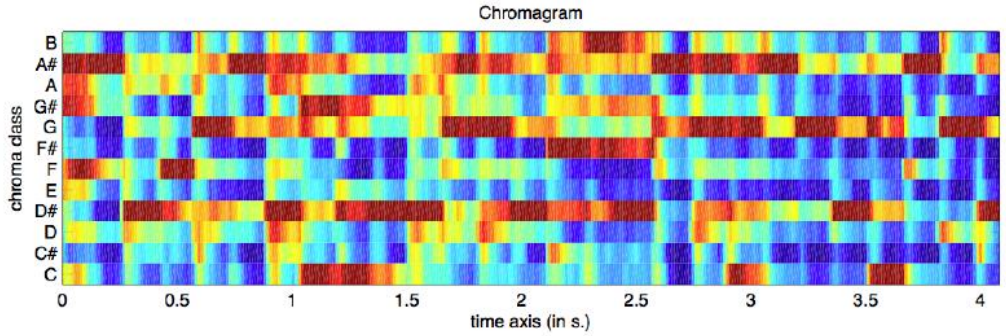
- Eventually, by the late 2000's and early 2010's, developments like the increased availability of massive amounts of training data ("big data") and efficient hardware processing of neural nets using video card GPUs made extremely large and complex neural nets viable
- This led to the era of "deep learning," which has shown that (very!) large and complex neural nets can be deployed very successful in many application areas

Deep learning

- Many different network architectures, training techniques and activation functions continue to be experimented with
- The "transformer" neural network architecture is the basis for many of the (enormous!) "large language models (LLMs)" like ChatGPT
  - GPT-4 has 1.76 trillion learnable parameters!
- Huge models like this are powerful enough that unsupervised approaches can be quite effective
  - Training data with only limited labels, or no labels at all, is often viable
  - Even low-quality, noisy data can still be useful
  - All this makes training data much cheaper

Deep learning

- Deep learning is also powerful enough that hand-engineered, pre-computed features are often no longer bothered with
  - Early layers of a network can in effect learn to find the features they need directly from (relatively) raw data like image pixel maps or audio "chromagrams" (below)



- Other kinds of pre-processing that were necessary for earlier types of AI are also typically no longer necessary either
  - e.g. edge detection in images or note segmentation in music audio

Deep learning

- The quick rise, power and relative opaqueness of deep learning has led to important ethical concerns, including:
  - Environmental impact
    - e.g. enormous amounts of electricity are needed to train models and (in the case of LLMs in particular) process prompts
  - Intellectual property rights
    - e.g. is it right that an AI music composer be trained on existing music without the permission or recompense of those who wrote that music?
  - Fraud
    - e.g. misrepresentation of machine generated content as one's own
  - Bias and fairness
    - e.g. building human biases or prejudices implicit in training data into models
  - Privacy
    - e.g. video or audio surveillance of public spaces by AIs
  - False output due to "hallucinations"

Alternatives to deep learning?

- What if you're worried by such ethical issues?
- What if only very limited training data is available?
  - e.g. Renaissance music
  - Deep learning techniques like "data augmentation" and "transfer learning" can only help so much
- What if you care not just about "what," but also "why?"
  - e.g. if I'm modelling a musician's style, I might want to know not only if it is distinct from another musician's style, but in what specific ways
  - The reasons for predictions output using deep learning are (usually) opaque

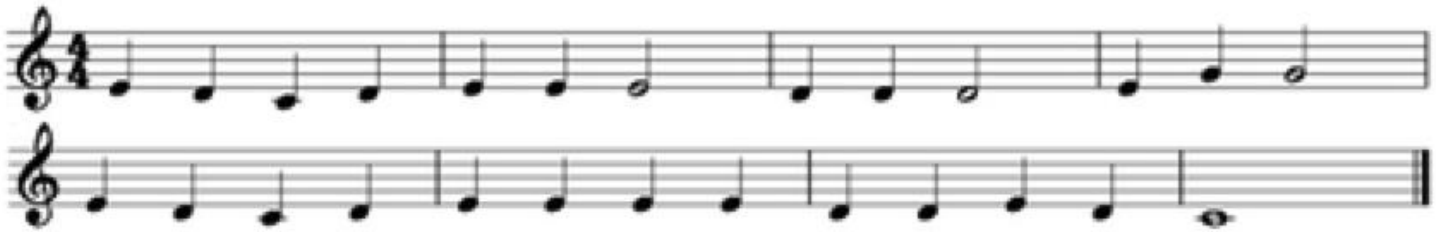
Statistically characterizing music

What is a musical "feature"?

- A piece of information that measures a characteristic of something (e.g. a piece of music) in a simple, consistent and precisely-defined way
- Represented using numbers
  - Can be a single value, or can be a set of related values (e.g. a histogram)
- Provides a summary description of the characteristic being measured
  - Usually macro, rather than local
- Usually extracted from pieces in their entirety
  - But can also be extracted from segments of pieces

Example: A simple (one-dimensional) feature

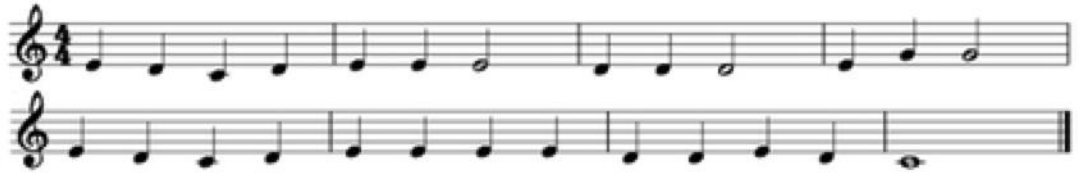
- Range (1-D): Difference in semitones between the lowest and highest pitches

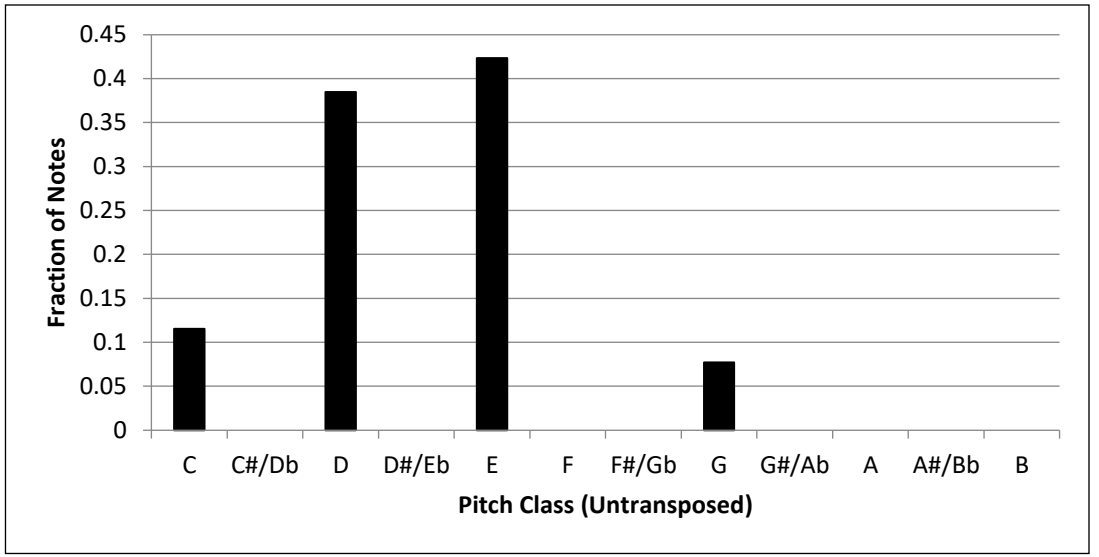


- Value of this feature for the above example: 7
  - G - C = 7 semitones

Example: A histogram feature

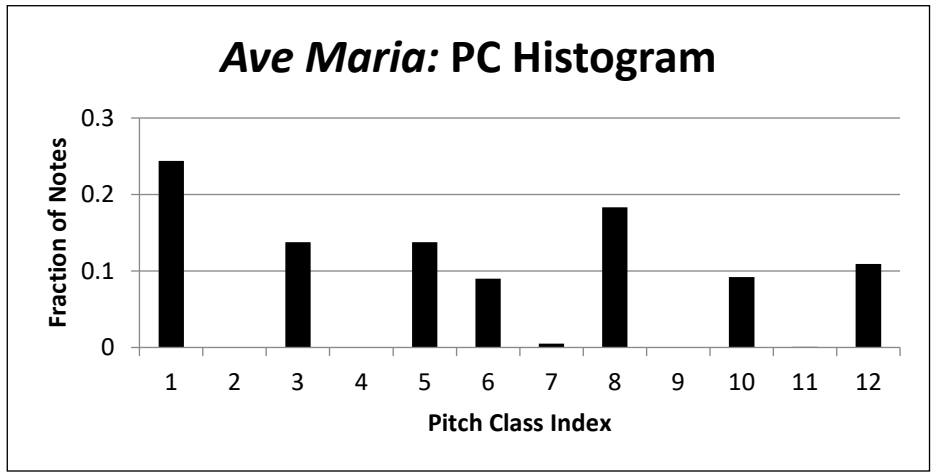
- Pitch Class Histogram: Consists of 12 values, each representing the fraction of all notes belonging to an enharmonic pitch class
- Histogram graph on right shows feature values for this melody
- Pitch class counts:
  - C: 3, D: 10, E: 11, G: 2
- Most common note is E:
  - 11/26 notes
  - Corresponds to a feature value of 0.423 for E

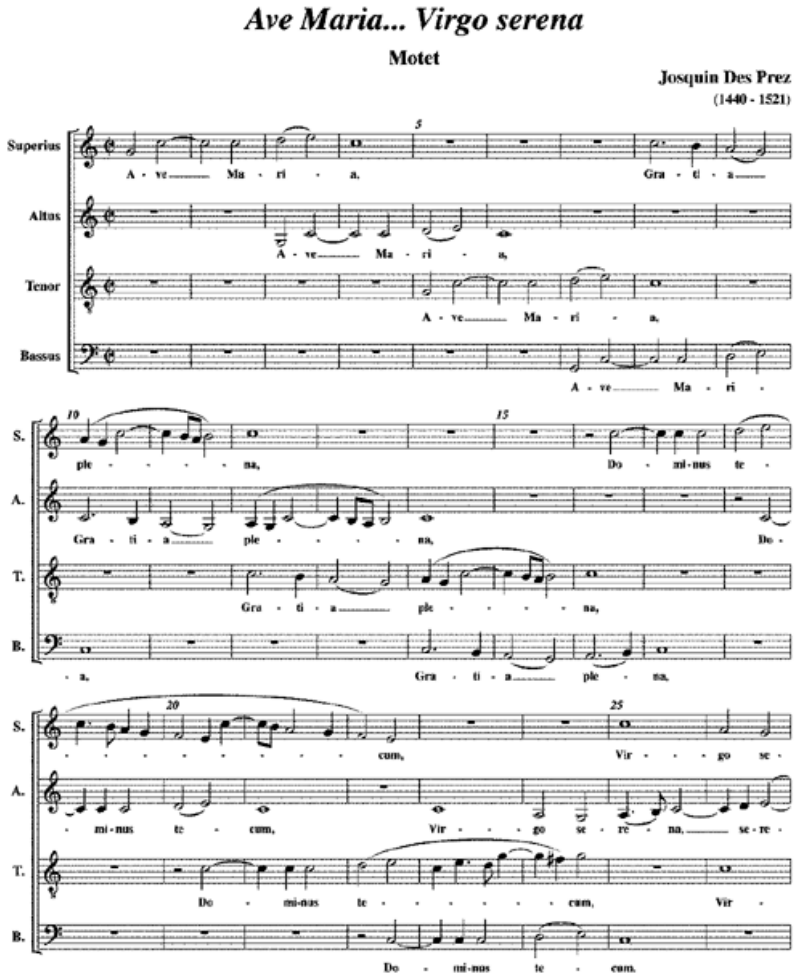




Josquin's *Ave Maria . . . Virgo serena*

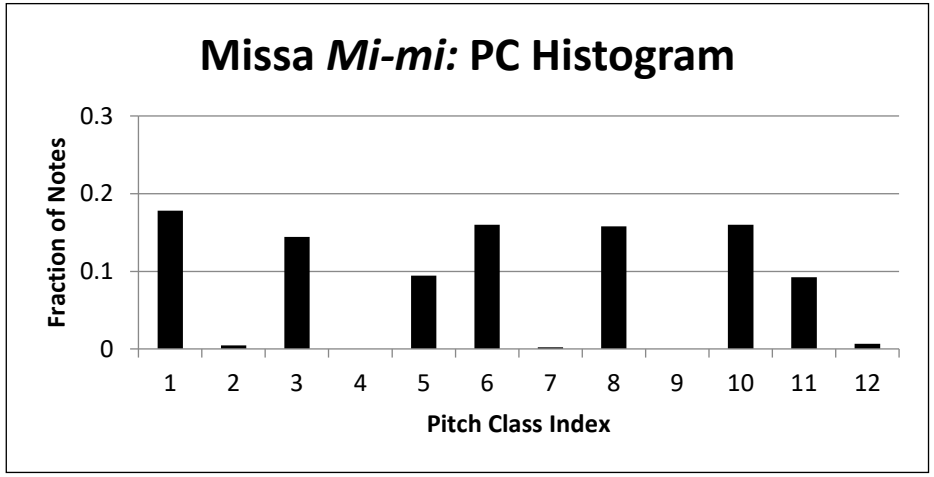
- Range: 34 (semitones)
- Repeated notes: 0.181 (18.1%)
- Vertical perfect 4^{ths}: 0.070 (7.0%)
- Rhythmic variability: 0.032
- Parallel motion: 0.039 (3.9%)

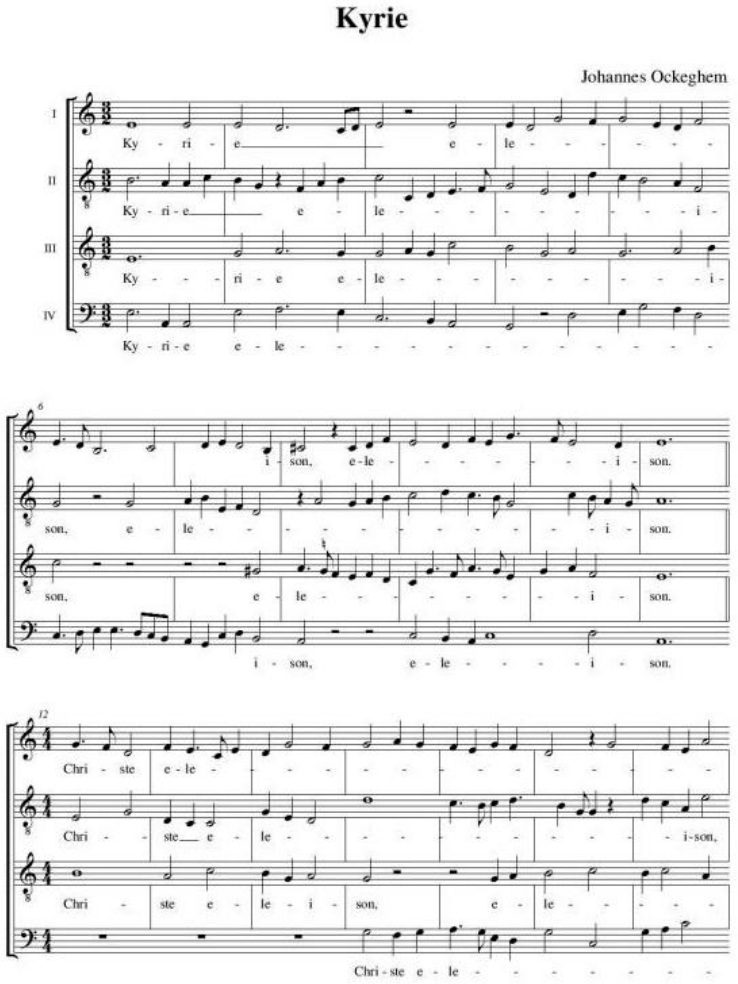




Ockeghem's Missa *Mi-mi* (Kyrie)

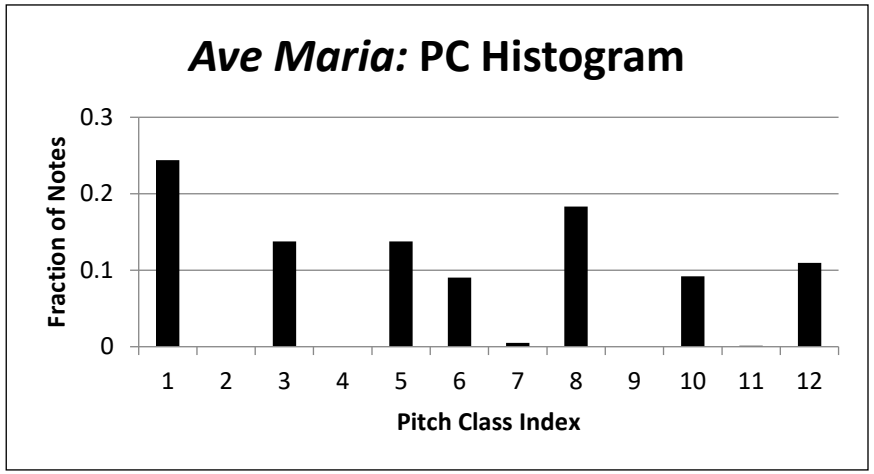
- Range: 26 (semitones)
- Repeated notes: 0.084 (8.4%)
- Vertical perfect 4^{ths}: 0.109 (10.9%)
- Rhythmic variability: 0.042
- Parallel motion: 0.076 (7.6%)

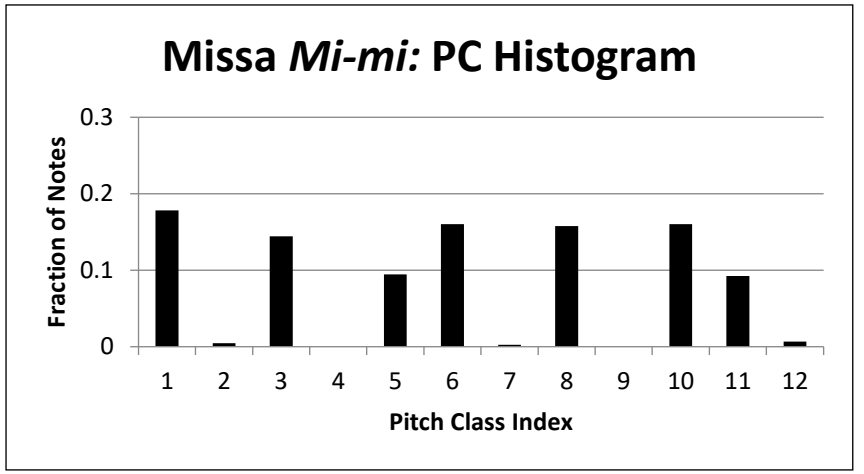




Feature value comparison

Feature	*Ave Maria*	Missa *Mi-mi*
Range	34	26
Repeated notes	0.181	0.084
Vertical perfect 4^{ths}	0.070	0.109
Rhythmic variability	0.032	0.042
Parallel motion	0.039	0.076





Comparing features

- Comparing pairs of pieces like this in terms of features can be revealing
  - Especially when that comparison involves hundreds or thousands of features, not just six
- Things get even more interesting when comparisons are made between hundreds or thousands of pieces, not just two
  - Especially when the music is divided into groups of interest, whose can then be collectively contrasted with one another
  - e.g. comparing the styles of composers, genres, regions, time periods, etc.

Ways of comparing feature values

- Manually:
  - Text editors
  - Spreadsheets
- With automatic assistance:
  - Statistical analysis software
    - e.g. SPSS, SAS, etc.
  - Machine learning and data mining software
    - e.g. Weka, Orange, etc.
- Many of these tools can produce helpful visualizations

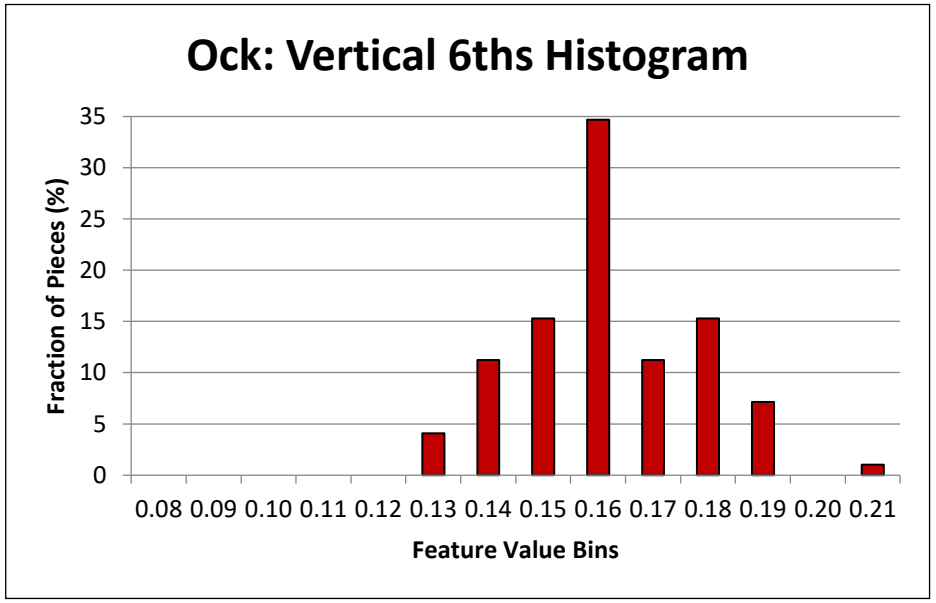
Feature visualization: Histograms

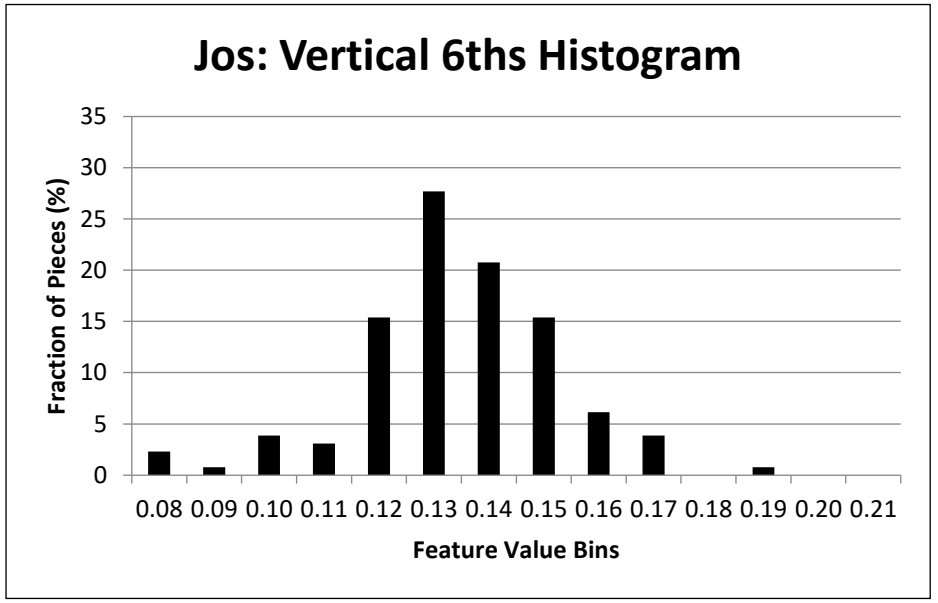
- Histograms offer a good way of visualizing how the values of a feature are distributed across a corpus of music as a whole
  - As opposed to focusing on individual pieces
- The x-axis corresponds to a series of bins, with each corresponding to a range of values for a given feature
  - e.g., the first bin could correspond to Parallel Motion feature values between 0 and 0.1, the next bin to Parallel Motion values between 0.1 and 0.2, etc.
- The y-axis indicates the fraction of pieces that have a feature value within the range of each given bin
  - e.g., if 30% of pieces in the corpus have Parallel Motion values between 0.1 and 0.2, then this bin (0.1 to 0.2) will have a y-coordinate of 30% (or, equivalently, 0.3)

Feature visualization: Histograms

- In other words: Each vertical bar on such a histogram represents the fraction of pieces in a corpus with a feature value falling in that bar's range of feature values

Feature visualization: Histograms

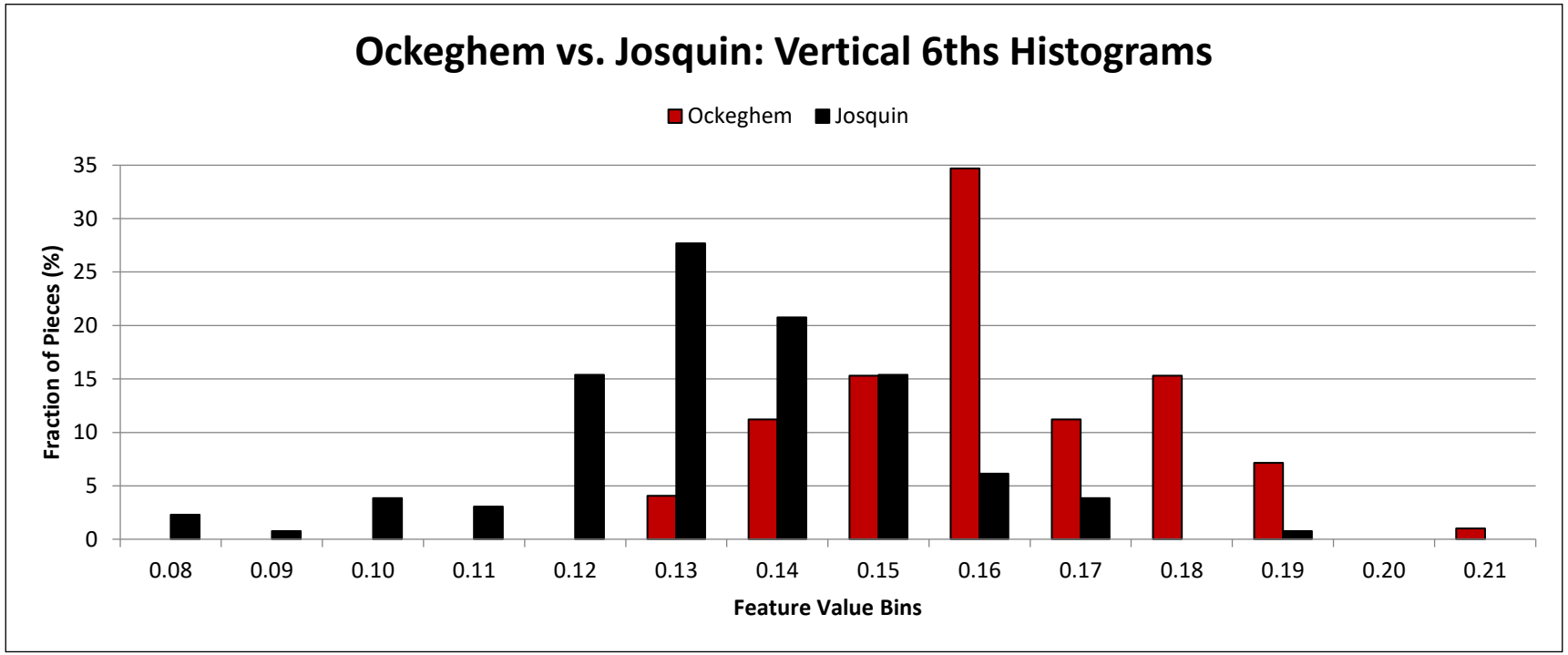




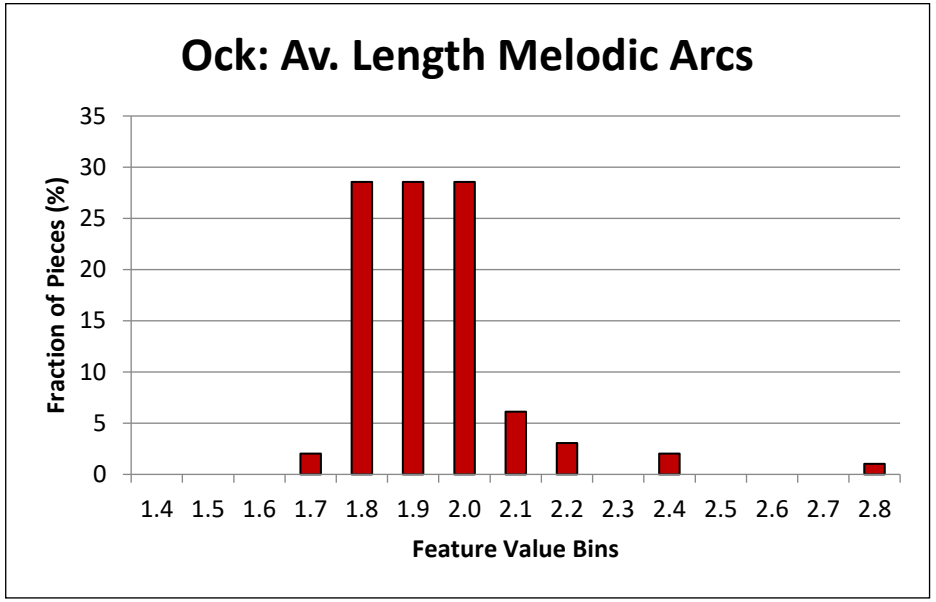
- These histograms show that Ockeghem tends to have more vertical 6^{ths} (between all pairs of voices) than Josquin
  - Ockeghem peaks in the 0.16 to 0.17 bin, at nearly 35%
  - Josquin peaks in the 0.13 to 0.14 bin, at about 28%
- Of course, there are also clearly many exceptions
  - This feature is helpful, but is limited if not considered in conjunction with other features

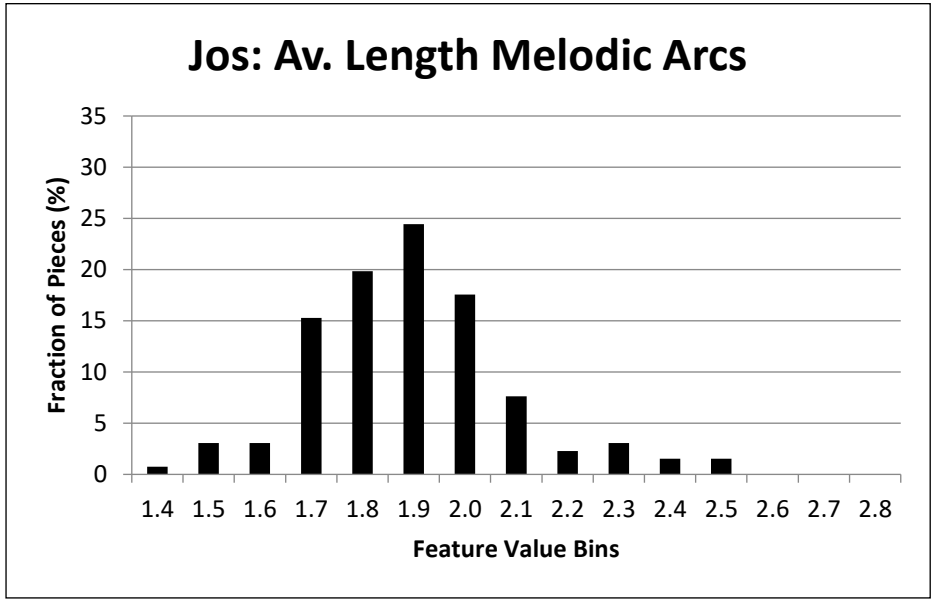
Feature visualization: Histograms

- The histograms for both composers can be superimposed onto a single chart:



Feature visualization: Histograms

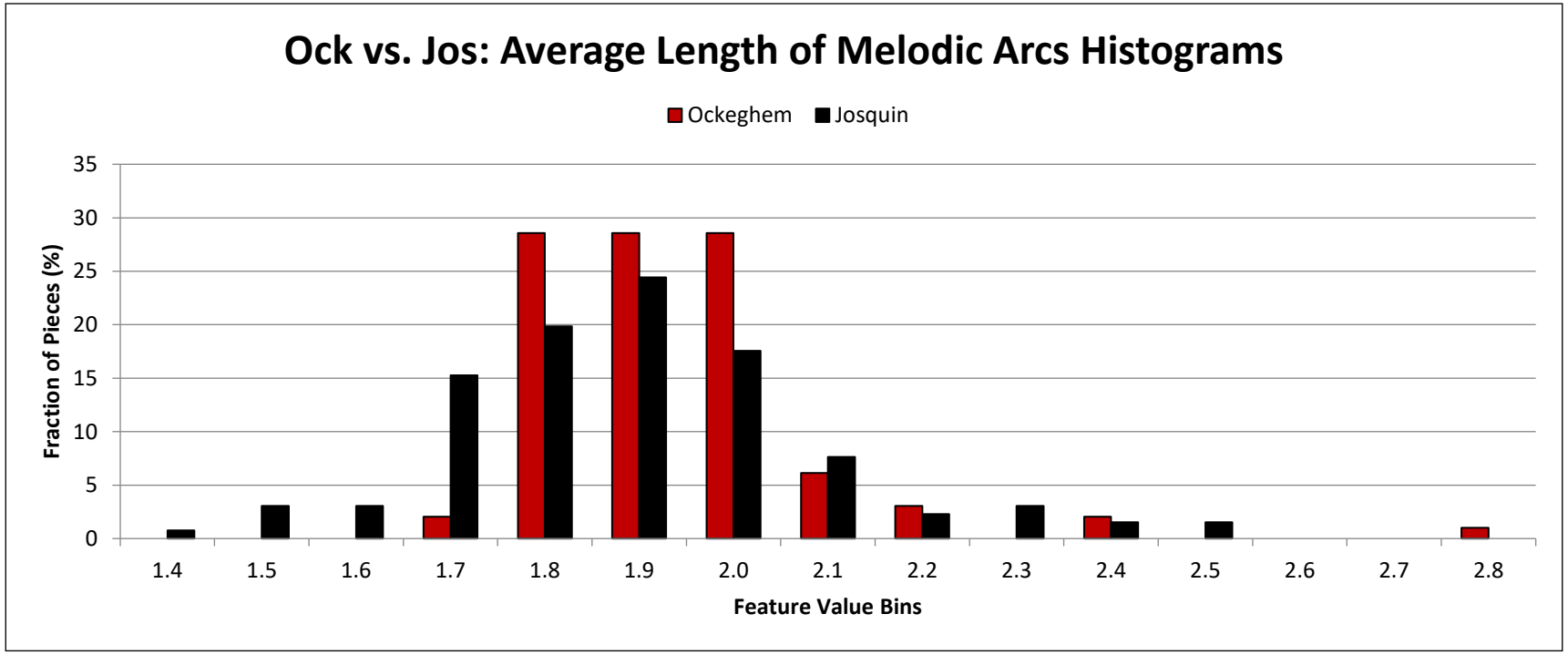




- These histograms show that Ockeghem tends to have longer melodic arcs (average number of notes separating melodic peaks & troughs)
  - Both peak in the 1.9 to 2.0 bin
  - However, Josquin's histogram is more skewed to the left
- Of course, there are once again clearly many exceptions
  - This feature is also helpful, but also limited if only considered alone

Feature visualization: Histograms

- Once again, the histograms for both composers can be superimposed onto a single chart:

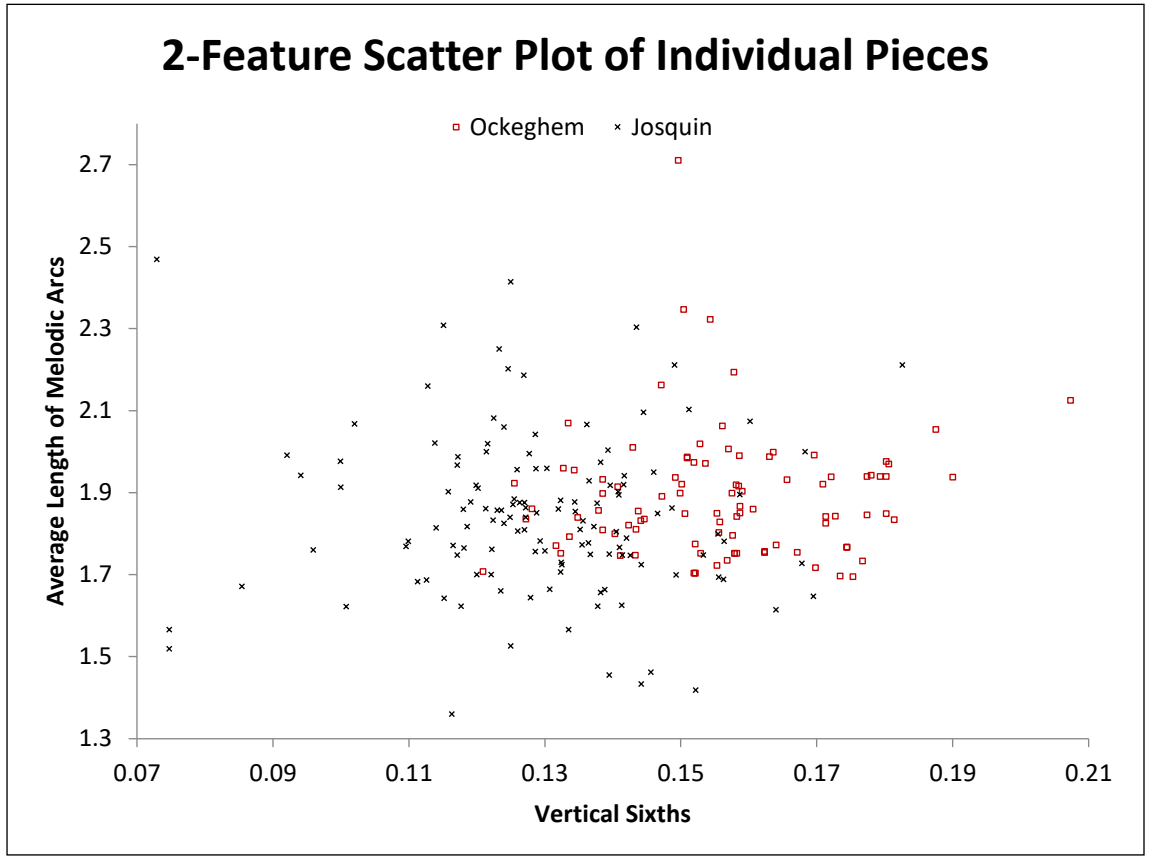


Feature visualization: Scatter plots

- Scatter plots are another nice way of visualizing feature data
  - The x-axis represents one feature
  - The y-axis represents a different feature
  - Each point represents the values of these two features for a single piece
- Scatter plots let you see pieces individually, rather than aggregating them into bins (as histograms do)
  - Scatter plots also let you see more clearly how features jointly separate the different categories of interest
- To make them easier to read, scatter plots typically have just 2 dimensions
  - Computer classifiers, in contrast, work with much larger n-dimensional scatterplots (one dimension per feature)

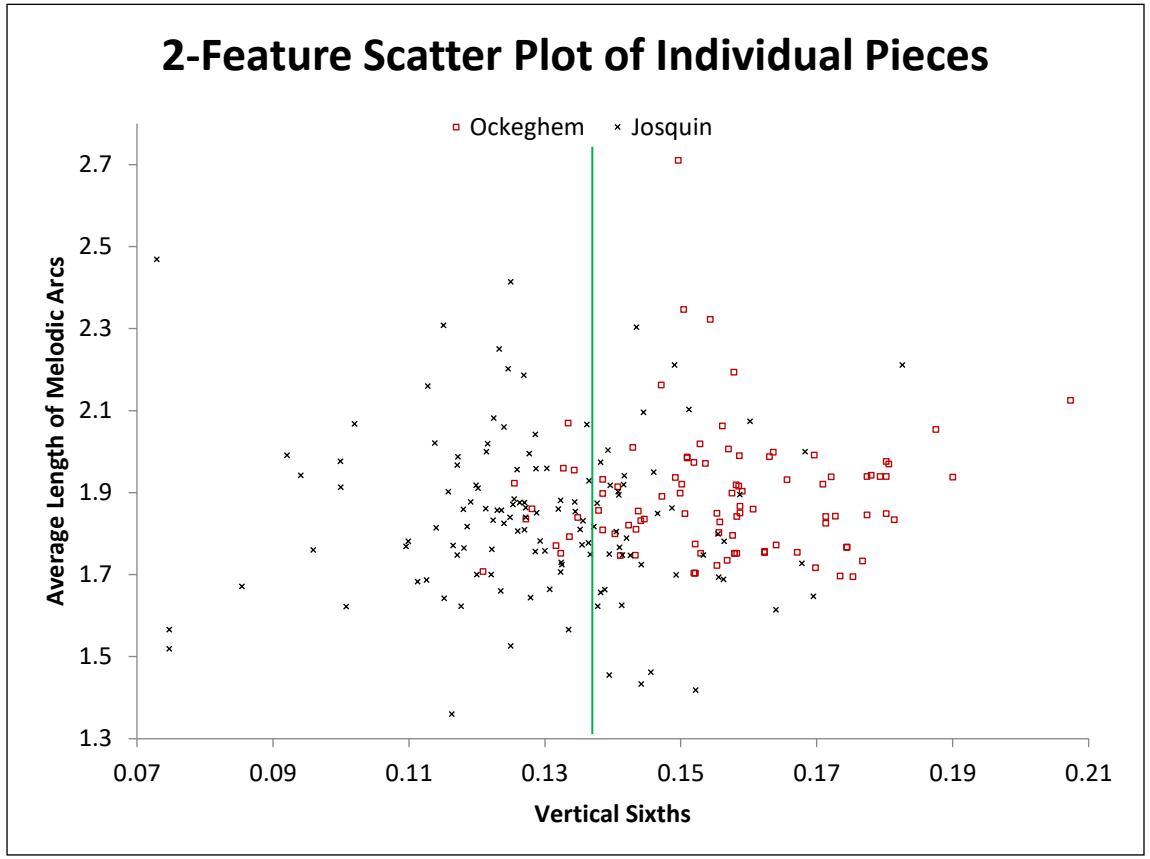
Feature visualization: Scatter plots

- Josquin pieces tend to be left and low on this graph



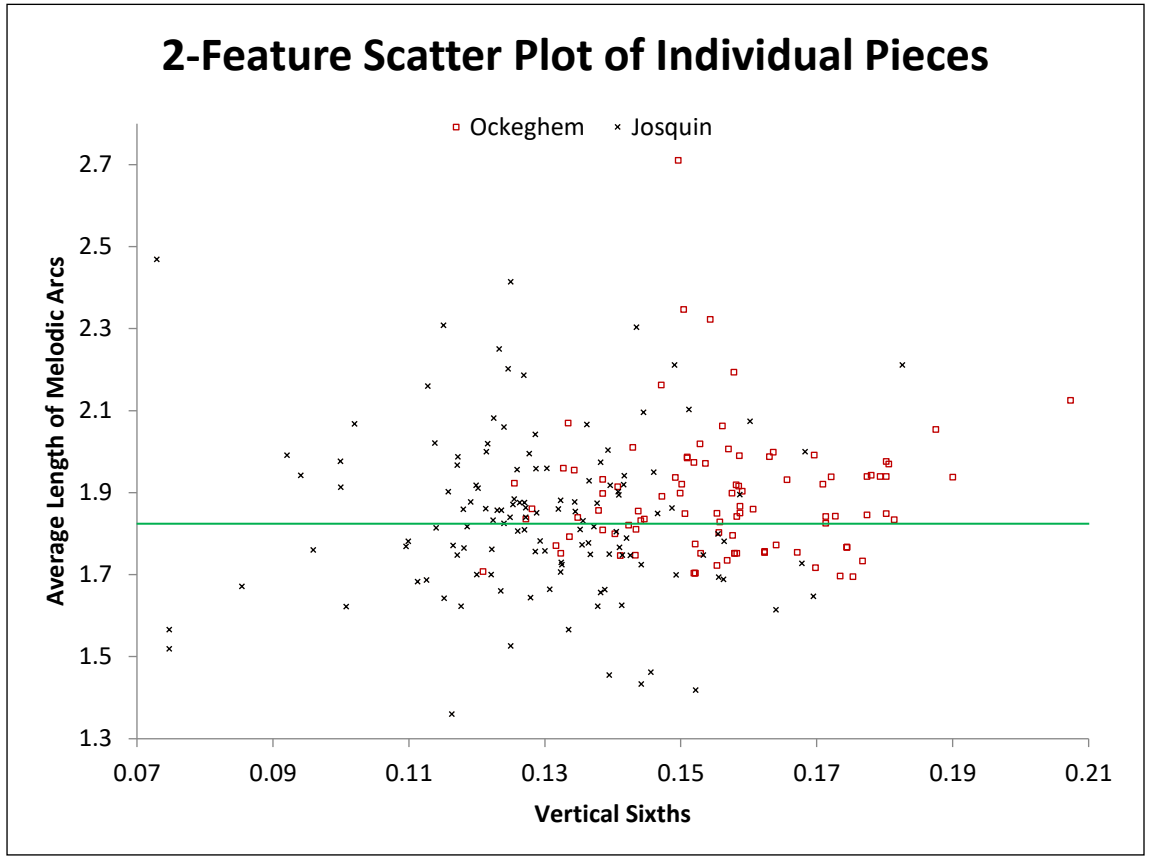
Feature visualization: Scatter plots

- Simply drawing a single 1-D vertical dividing line ("discriminant") results in a not entirely terrible classifier based only on Vertical Sixths
  - Can get 62% classification accuracy using an SVM classifier and just this one feature
  - But many pieces would still be misclassified



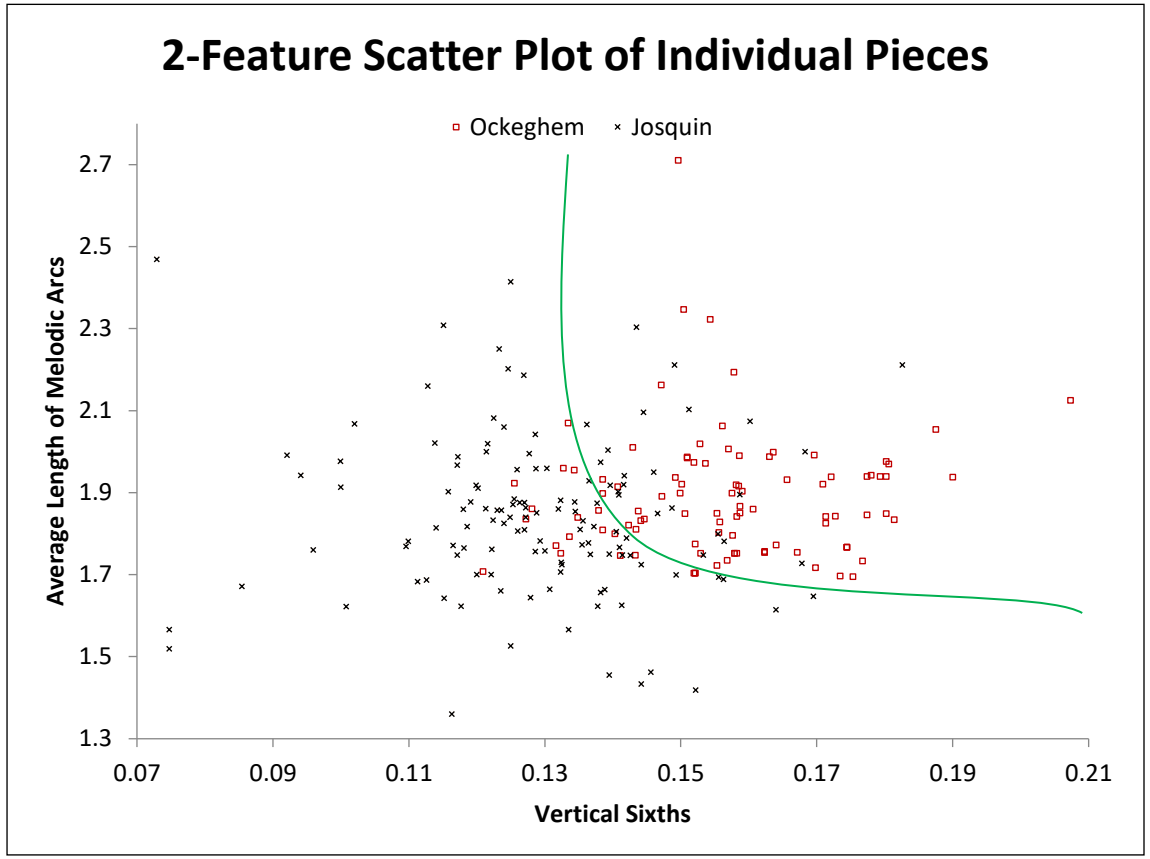
Feature visualization: Scatter plots

- Could alternatively draw a 1-D discriminant dividing the pieces based only on the Average Length of Melodic Arcs
  - Can get 57% classification accuracy using an SVM classifier and just this one feature
  - Not as good as the Vertical Sixths discriminant (62%), but still better than chance (50%)



Feature visualization: Scatter plots

- Drawing a curve (another kind of discriminant) divides the composers still better than either of the previous (linear) discriminants
  - Can get 80% accuracy using an SVM classifier looking at just these 2 features!
- More than 2 features are needed to improve performance further



Choosing features to implement

- What particular features should we investigate?
  - Yes, probably the ones we already know or suspect are important to the kinds of music under consideration
  - But also ideally ones that are important, but we do not know or suspect it yet
- To do this, we may need a lot of diverse features
  - So we encourage unexpected but important surprises
  - So we are less likely to miss important insights
  - So we can apply them to many types of music
  - So we can address the interests of many different researchers
- The same can be said for data
  - The more music and the more varied it is the better (typically)

Calculating features

- So how can music researchers get access to all these diverse and wonderful features?

Highlights from my own research

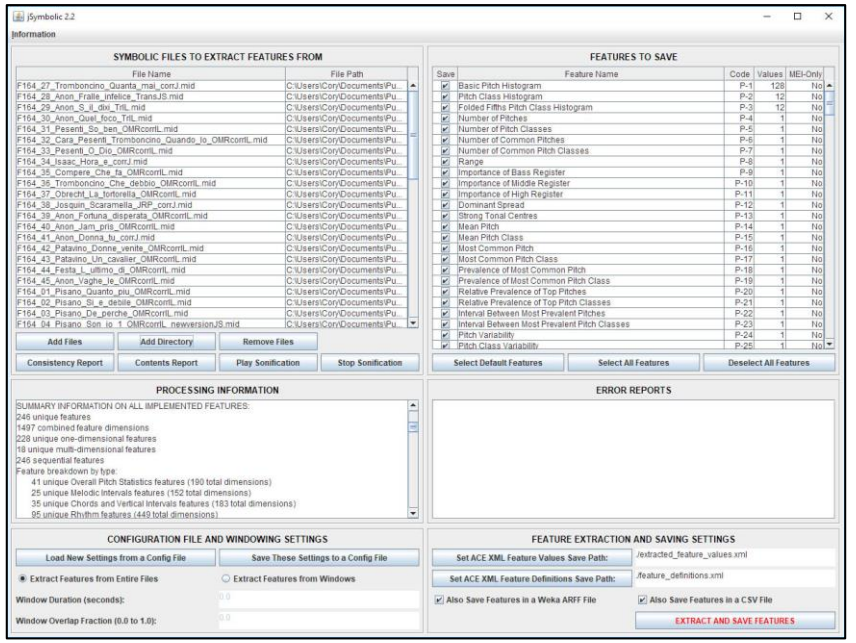
jSymbolic: Introduction

- jSymbolic is a software platform I have been developing since 2004 for automatically extracting features from symbolic musical representations (digital musical scores) such as MIDI
  - Tristano Tenaglia and Rian Adamian, two research assistants, also made important contributions between 2015 to 2020
- Free and open-source
  - https://jmir.sourceforge.net + https://github.com/DDMAL/jSymbolic2

- McKay, C., J. Cumming, and I. Fujinaga. 2018. jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research. *Proceedings of the International Society for Music Information Retrieval Conference*. 348–354.
- McKay, C., T. Tenaglia, and I. Fujinaga. 2016. jSymbolic2: Extracting features from symbolic music representations. *Extended Abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference*.
- McKay, C., and I. Fujinaga. 2006. jSymbolic: A feature extractor for MIDI files. *Proceedings of the International Computer Music Conference*. 302–305.

What does jSymbolic do?

- (Version 2.2, the current release version) extracts 246 unique features
  - Some of these are multi-dimensional histograms
  - In all, (version 2.2) extracts a total of 1497 separate values



jSymbolic 2.2: Feature types (1/2)

- Pitch Statistics:
  - What are the occurrence rates of different pitches and pitch classes?
  - How tonal is the piece?
  - How much variety in pitch is there?
- Melody / horizontal intervals:
  - What kinds of melodic intervals are present?
  - How much melodic variation is there?
  - What kinds of melodic contours are used?
- Chords / vertical intervals:
  - What vertical intervals are present?
  - What types of chords do they combine to make?
  - How much harmonic movement is there?

jSymbolic 2.2: Feature types (2/2)

- Rhythm:
  - Rhythmic values of notes
  - Intervals between the attacks of different notes
  - Use of rests
  - What kinds of meter are used?
  - Rubato?
- Texture:
  - How many independent voices are there and how do they interact (e.g. moving in parallel, crossing voices, etc.)?
- Instrumentation:
  - What types of instruments are present and which are given particular importance relative to others?
- Dynamics:
  - How loud are notes and what kinds of dynamic variations occur?

jSymbolic: Extensibility

- jSymbolic is specifically designed such that music researchers can design their own features and easily add these features to the jSymbolic infrastructure as plug-ins

To come in jSymbolic 3

- Many new features
  - 533 unique features and 2040 feature values
  - Including features base on n-grams
    - A useful way of statistically exploring patterns in melodic, harmonic or rhythmic sequences

- McKay, C. 2023. From jSymbolic 2 to 3: More musical features. *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*. 752–755.
- McKay, C., J. Cumming, and I. Fujinaga. 2023. Rhythmic, melodic and vertical n-gram features as a means of studying symbolic music computationally. Presented at the *Digital Humanities Conference*.
- McKay, C., R. Adamian, J. Cumming, and I. Fujinaga. 2020. Exploring Renaissance music using n-gram aggregates to summarize local musical content. Presented at the *Medieval and Renaissance Music Conference*.

Musicological research based on jSymbolic's features

Musical genre (popular and art music)

- I showed that (just) jSymbolic features and machine learning could be used to automatically identify the genre of a piece of music
  - e.g. 93% accuracy among 5 genres
  - e.g. 78% accuracy among 10 genres
  - e.g. 64% accuracy among 38 genres
- Features based on instrumentation were especially effective

- McKay, C., J. Cumming, and I. Fujinaga. 2018. jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research. *Proceedings of the International Society for Music Information Retrieval Conference*. 348–354.
- McKay, C. 2010. Automatic music classification with jMIR. *Ph.D. Dissertation*. McGill University, Canada.
- McKay, C., and I. Fujinaga. 2007. Style-independent computer-assisted exploratory analysis of large music collections. *Journal of Interdisciplinary Music Studies* 1 (1): 63–85.
- McKay, C., and I. Fujinaga. 2006. Musical genre classification: Is it worth pursuing and how can it be improved?. *Proceedings of the International Conference on Music Information Retrieval*. 101–106.
- McKay, C., and I. Fujinaga. 2005. Automatic music classification and the importance of instrument identification. *Proceedings of the Conference on Interdisciplinary Musicology*. CD-ROM.
- McKay, C., and I. Fujinaga. 2006. Style-independent computer-assisted exploratory analysis of large music collections. Presented at the *Joint Meeting of the American Musicological Society and the Society for Music Theory*.
- McKay, C., and I. Fujinaga. 2004. Automatic genre classification as a study of the viability of high-level features for music classification. *Proceedings of the International Computer Music Conference*. 367–370.
- McKay, C. and I. Fujinaga. 2004. Automatic genre classification using large high-level musical feature sets. *Proceedings of the International Conference on Music Information Retrieval*. 525–530.
- McKay, C. 2004. Automatic genre classification of MIDI recordings. *M.A. Thesis*. McGill University, Canada.

Renaissance musicology: Origins of the madrigal

- The madrigal (a type of polyphonic vocal music from the 17th to 18th centuries) is typically said in the literature to be derived primarily from motets and chansons
- Our feature-based analysis found that early madrigals are actually statistically fairly dissimilar to both motets and chansons, and are musically much closer to other Italian-texted forms, especially the vilotta

- Cumming, J., and C. McKay. 2021. Using corpus studies to find the origins of the madrigal. *Proceedings of the Future Directions of Music Cognition International Conference*. 38–42.
- Cumming, J., and C. McKay. 2018. Revisiting the origins of the Italian madrigal using machine learning. Presented at the *Medieval and Renaissance Music Conference*.

Renaissance musicology: Composer style

- jSymbolic features and machine learning can also be used to automatically recognize compositional style
  - e.g. able to identify the composer of securely attributed works 92% of the time (among 7 composers)
  - Could even distinguish very stylistically similar composers
    - e.g. 86% accuracy for Josquin and La Rue
- Particularly useful for Renaissance music, much of which is unattributed or insecurely attributed

- McKay, C., J. Cumming, and I. Fujinaga. 2018. jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research. *Proceedings of the International Society for Music Information Retrieval Conference*. 348–354.
- McKay, C., J. Cumming, and I. Fujinaga. 2017. Characterizing composers using jSymbolic2 features. *Extended Abstracts for the Late-Breaking Demo Session of the 18th International Society for Music Information Retrieval Conference*.
- McKay, C., T. Tenaglia, J. Cumming, and I. Fujinaga. 2017. Using statistical feature extraction to distinguish the styles of different composers. Presented at the *Medieval and Renaissance Music Conference*.

Renaissance musicology: Composer style

- Also used jSymbolic feature values to identify specific differences in the styles of different composers
- I began by asking an eminent specialist musicologist and an eminent specialist theorist to identify what they thought differentiated the styles of Josquin and Ockeghem, and then checked their expectations using calculated features . . .

Renaissance musicology: Composer style

- Josquin vs. Ockeghem *a priori* expert expectations: Ockeghem has . . .
  - Slightly more large leaps (larger than a 5th)
  - Less stepwise motion in some voices
  - More notes at the bottom of the range
  - Slightly more chords (or simultaneities) without a third
  - Slightly more dissonance
  - A lot more triple meter
  - More varied rhythmic note values
  - More 3-voice music
  - Less music for more than 4 voices

Renaissance musicology: Composer style

- Josquin vs. Ockeghem empirical reality: Ockeghem has . . .
  - OPPOSITE: Slightly more large leaps (larger than a 5^{th})
  - SAME: Less stepwise motion in some voices
  - SAME: More notes at the bottom of the range
  - SAME: Slightly more chords (or simultaneities) without a third
  - OPPOSITE: Slightly more dissonance
  - YES: A lot more triple meter
  - SAME: More varied rhythmic note values
  - YES: More 3-voice music
  - YES: Less music for more than 4 voices

Renaissance musicology: Composer style

- Additional unexpected insights revealed from a purely feature-based analysis:
  - Rhythm-related features are particularly important
    - e.g. Josquin tends to have greater rhythmic variety
      - Especially in terms of both especially short and especially long notes
    - It turns out that rhythmic features in general tend to have much more predictive power than most expert Renaissance scholars expect
      - Much of the literature largely ignores rhythm
  - Ockeghem tends to have more diminished triads
  - Ockeghem tends to have more vertical sixths
  - Ockeghems tends to have longer melodic arcs

Renaissance musicology: Composer style

- In another study, I used jSymbolic features to validate a benchmark ranking by musicologist Jesse Rodin of the attribution certainty of all pieces associated with Josquin
  - Rodin's rankings are based purely on historical evidence, not on the music itself
  - My analysis was based only on the music
- Happily, the two approaches produced largely coinciding results
- Updated version of this work to be published in a 2025 book chapter

Renaissance musicology: Other published research with jSymbolic using jSymbolic features

- Attribution of anonymous and doubtfully attributed works:
  - Masses transcribed by Siro Cisilino
  - Coimbra manuscripts
  - *Ave verum corpus* and *O decus virgineum*
  - *Ave festiva ferculis*
  - Gaffurius Codices
- Regional style in Iberian Renaissance music:
  - Musical influences of Pedro Fernández Buch
  - Musical Influences of Cristóbal de Morales and Francisco Guerrero

Renaissance musicology: Other published work using jSymbolic features

- Cuenca, M. E., and C. McKay. 2023. The stylistic origin of the anonymous 16th century masses transcribed by Siro Cisilino (1903-1987) at the Fondazione Cini: A statistical and machine learning approach. Presented at the *Medieval and Renaissance Music Conference*.
- McKay, C., and M. E. Cuenca. 2022. Influencias musicales en las misas y motetes de Cristóbal de Morales y Francisco Guerrero: Una aproximación estadística. In *Musicología en transición*, eds. J. Marín-López, A. Mazuela-Anguita and J. J. Pastor-Comín, 1031–1052. Madrid, Spain: Sociedad Española de Musicología.
- Cuenca, M. E., and C. McKay. 2022. Musical influences on the masses of Pedro Fernández Buch (c. 1574-1648): A stylistic comparison using statistical analysis. Presented at the *Medieval and Renaissance Music Conference*.
- Cuenca, M. E., and C. McKay. 2021. Exploring musical style in the anonymous and doubtfully attributed mass movements of the Coimbra manuscripts: A statistical and machine learning approach. *Journal of New Music Research* 50 (3): 199–219.
- Cuenca, M. E., and C. McKay. 2021. Influencias musicales en las misas y motetes de Cristóbal de Morales y Francisco Guerrero: Una aproximación estadística. Presented at the *Congreso de la Sociedad Española de Musicología*.
- McKay, C. 2021. Exploring composer attribution in motet cycles using machine learning. *Gaffurius Codices Online*, Schola Cantorum Basiliensis.
- Rodríguez-García, E., and C. McKay. 2021. Composer attribution of Renaissance motets: A case study using statistical features and machine learning. In *The Anatomy of Iberian Polyphony Around 1500*, eds. E. Rodríguez-García and J. P. d'Alvarenga, 401–38. Kassel, Germany: Edition Reichenberger.
- McKay, C., and M. E. Cuenca. 2021. Musical influences on the masses and motets of Cristóbal de Morales and Francisco Guerrero: A statistical approach. Presented at the *Medieval and Renaissance Music Conference*.
- Rodriguez-Garcia, E., and C. McKay. 2021. *Ave festiva ferculis:* Exploring attribution by combining manual and computational analysis. Presented at the *Medieval and Renaissance Music Conference*.
- Cuenca, M. E., and C. McKay. 2019. Análisis estadístico de misas ibéricas renacentistas a través del software jSymbolic. Presented at the *El análisis musical actual: Marco teórico e interdisciplinariedad conference*.
- Cuenca, M. E., and C. McKay. 2019. Exploring musical style in the anonymous and doubtfully attributed mass movements of the Coimbra manuscripts: A statistical approach. Presented at the *Medieval and Renaissance Music Conference*.

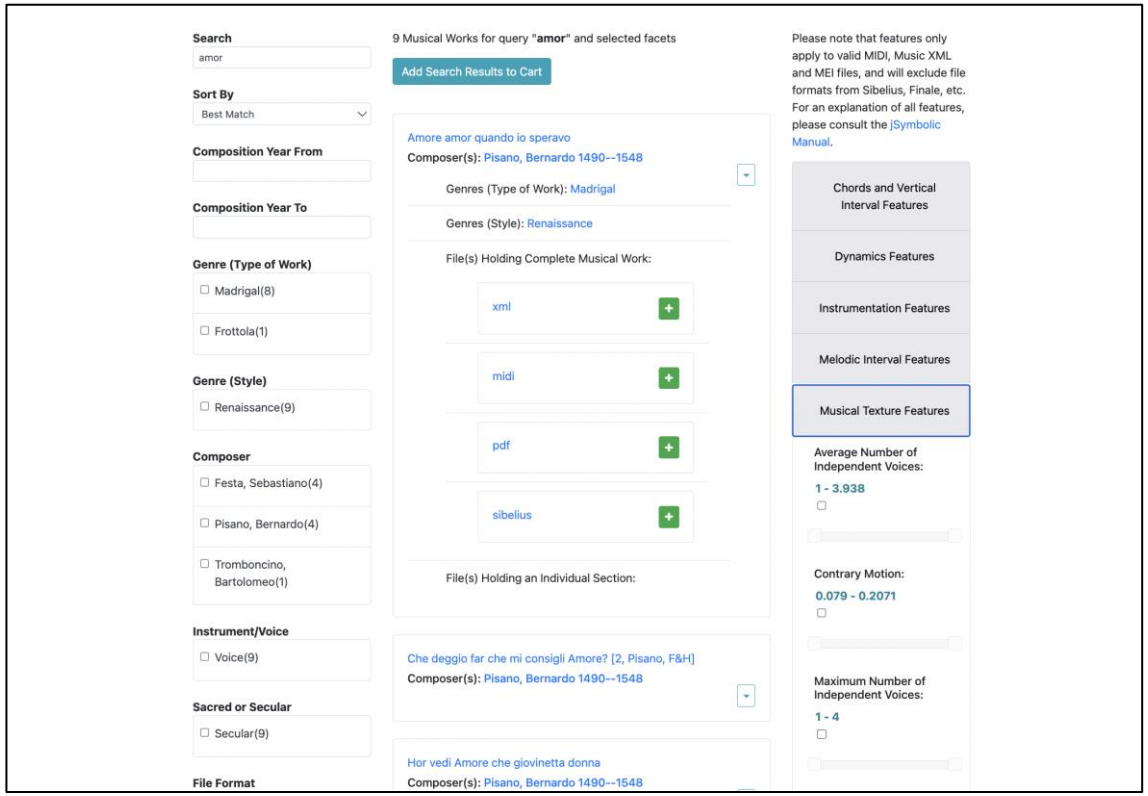
Chord identification in Baroque music

- I co-supervised Yaolong Ju's PhD research on using machine learning to identify chords in Baroque music
  - Did not use jSymbolic, did use neural networks
  - Sometimes using figured bass annotations
  - Interesting investigation of how to address intrinsic ambiguity

- Ju, Y., S. Margot, C. McKay, and I. Fujinaga. 2020. Automatic chord labeling: A figured bass approach. *Proceedings of DLfM 2020: The 7th International Conference on Digital Libraries for Musicology*. 27–31.
- Ju, Y., S. Margot, C. McKay, L. Dahn, and I. Fujinaga. 2020. Automatic figured bass annotation using the new Bach Chorales Figured Bass dataset. *Proceedings of the International Society for Music Information Retrieval Conference*. 640–646.
- Ju, Y., S. Margot, C. McKay, and I. Fujinaga. 2020. Figured bass encodings for Bach chorales in various symbolic formats: A case study. *Music Encoding Conference Proceedings*. 71–74.
- Ju, Y., S. Howes, C. McKay, N. Condit-Schultz, J. Calvo-Zaragoza, and I. Fujinaga. 2019. An interactive workflow for generating chord labels for homorhythmic music in symbolic formats. *Proceedings of the International Society for Music Information Retrieval Conference*. 862–869.

SIMSSA DB

- A prototype database of symbolic music files intended for research in computational musicology
  - And, eventually, associated images, audio recordings, texts, etc.
  - https://db.simssa.ca
- Searchable by:
  - Free text
  - Faceted metadata
  - Feature values
    - Auto-annotated by jSymbolic



SIMSSA DB

- To be used in combination with other music repositories as part of the long-term LinkedMusic project (2022-2029)
- I designed the data model, but all the actual implementation work was done by RAs I co-supervised:
  - Gustavo Polins Pedro, Yaolong Ju, Rebecca Mizrahi, Hong Van Pham

- McKay, C., and J. Cumming. 2022. Summary features as the basis for content-based queries of symbolic music repositories. Presented at the *Congress of the International Association of Music Libraries, Archives and Documentation Centres*.
- Hopkins, E., Y. Ju, G. Polins Pedro, C. McKay, J. Cumming, and I. Fujinaga. 2019. SIMSSA DB: Symbolic music discovery and search. Poster presentation at the *International Conference on Digital Libraries for Musicology*.
- Ju, Y., G. Polins Pedro, C. McKay, E. Hopkins, J. Cumming, and I. Fujinaga. 2019. Enabling music search and analysis: A database for symbolic music files. Presented at the *Music Encoding Conference*.
- McKay, C., E. Hopkins, G. Polins Pedro, Y. Ju, A. Kam, J. Cumming, and I. Fujinaga. 2019. A collaborative symbolic music database for computational research on music. Presented at the *Medieval and Renaissance Music Conference*.
- McKay, C., A. Hankinson, J. Cumming, and I. Fujinaga. 2017. A database model for computational music research. Poster presentation at the *International Workshop on Digital Libraries for Musicology*.

General data and metadata issues

- The availability, quality and selection of data and metadata can be an essential factors when conducting empirical studies

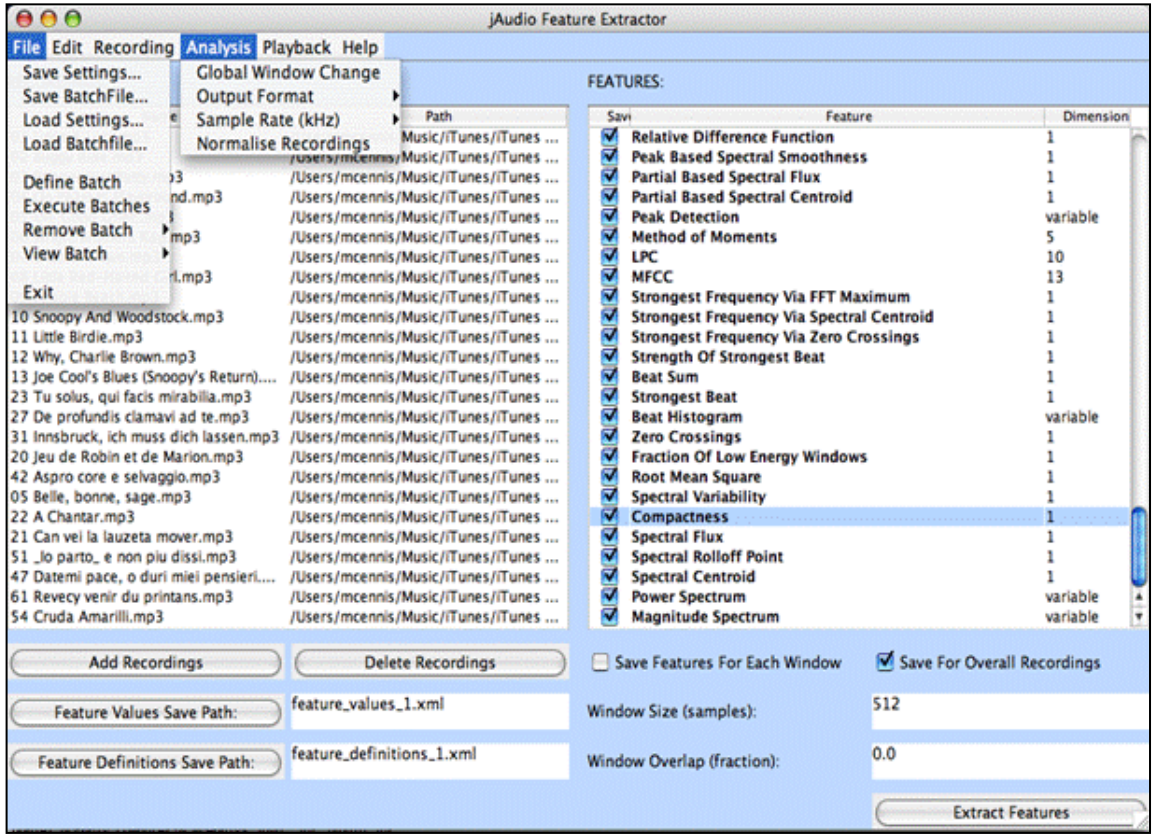
- McKay, C., J. Cumming, and I. Fujinaga. 2021. Lessons learned in a large-scale project to digitize and computationally analyze musical scores. *Digital Scholarship in the Humanities* 36 (s2): ii198–ii202.
- Cumming, J., C. McKay, J. Stuchbery, and I. Fujinaga. 2018. Methodologies for creating symbolic corpora of Western music before 1600. *Proceedings of the International Society for Music Information Retrieval Conference*. 491–498.
- McKay, C., and I. Fujinaga. 2015. Building an infrastructure for a 21st-century global music library. *Extended Abstracts for the Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference*.
- McKay, C., and I. Fujinaga. 2013. Expressing musical features, class labels, ontologies, and metadata using ACE XML 2.0. In *Structuring Music Through Markup Language: Designs and Architectures*, ed. J. Steyn, 48–79. Hershey, PA: IGI Global.
- McKay, C., and D. Bainbridge. 2011. A musical web mining and audio feature extraction extension to the Greenstone digital library software. *Proceedings of the International Society for Music Information Retrieval Conference*. 459–464.
- Angeles, B., C. McKay, and I. Fujinaga. 2010. Discovering metadata inconsistencies. *Proceedings of the International Society for Music Information Retrieval Conference*. 195–200.
- McKay, C., J. A. Burgoyne, J. Thompson, and I. Fujinaga. 2009. Using ACE XML 2.0 to store and share feature, instance and class data for musical classification. *Proceedings of the International Society for Music Information Retrieval Conference*. 303–308.
- McEnnis, D., C. McKay, and I. Fujinaga. 2006. Overview of OMEN. *Proceedings of the International Conference on Music Information Retrieval*. 7–12.
- McKay, C., D. McEnnis and I. Fujinaga. 2006. A large publicly accessible prototype audio database for music research. *Proceedings of the International Conference on Music Information Retrieval*. 160–163.

jMIR and multimodal research

- All the research I have presented so far was based on symbolic data (i.e. digital scores), like MIDI files
- However, I have also conducted research involving many other kinds of musical data (e.g. audio, images, video, text, etc.), both individually and in combination ("multimodal" analysis)
- In fact, jSymbolic is just one part of the larger open-source jMIR multimodal music research software suite that I wrote

jMIR's jAudio 2

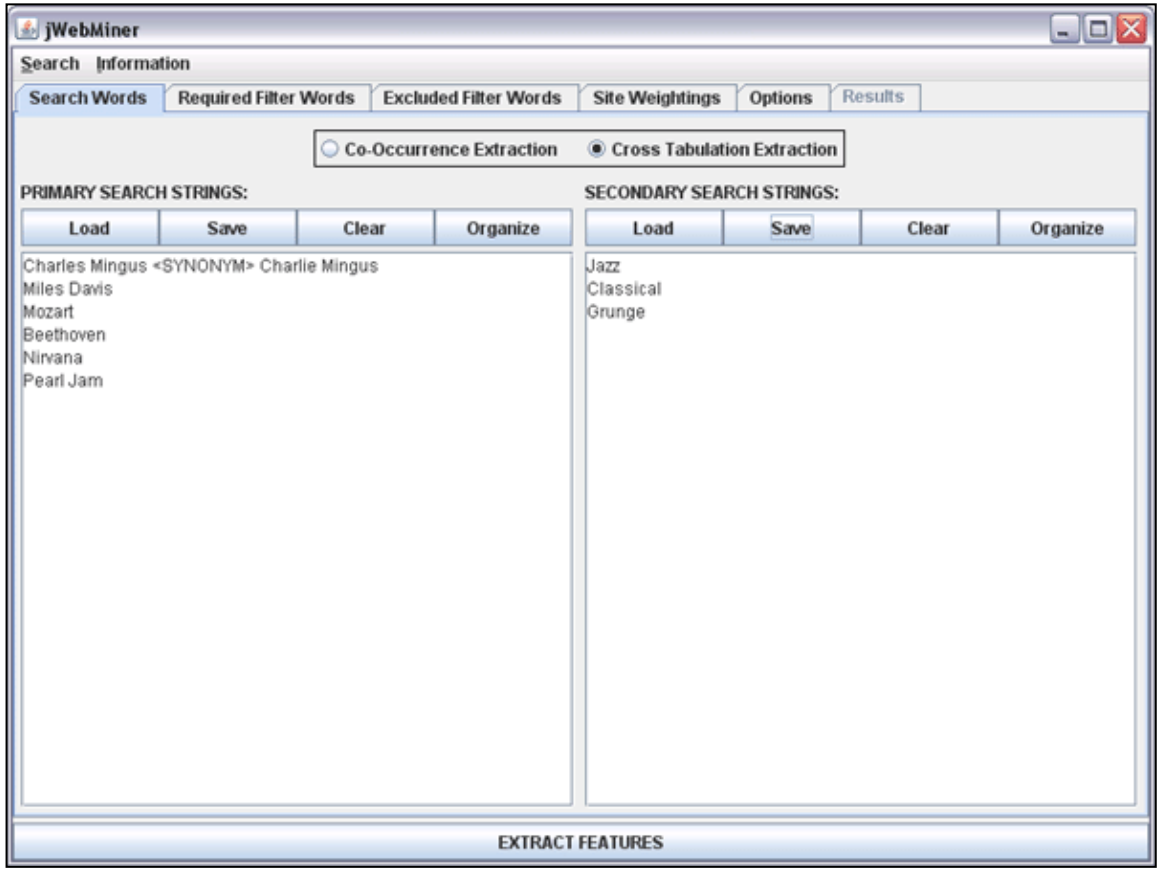
- Implemented jointly with Daniel McEnnis
- Extracts features from audio files
  - e.g. MP3, WAV, AIFF, AU, SND
- 28 bundled core features
  - Mainly low-level, some high-level



- McEnnis, D., C. McKay, and I. Fujinaga. 2006. jAudio: Additions and improvements. *Proceedings of the International Conference on Music Information Retrieval*. 385–386.
- McEnnis, D., C. McKay, I. Fujinaga, and P. Depalle. 2005. jAudio: A feature extraction library. *Proceedings of the International Conference on Music Information Retrieval*. 600–603.

jMIR's jWebMiner 2

- Extracts cultural features from the web based on automated search engine queries
  - Calculates how often particular strings co-occur on the same web pages
  - e.g. how often does "J. S. Bach" compared to "Prokofiev" co-occur on web pages with "Baroque"?
- Version 2 (implemented by Gabriel Vigliensoni) also extracts features from Last.fm social tags



- Vigliensoni, G., C. McKay, and I. Fujinaga. 2010. Using jWebMiner 2.0 to improve music classification performance by combining different types of features mined from the web. *Proceedings of the International Society for Music Information Retrieval Conference*. 607–612.
- McKay, C., and I. Fujinaga. 2007. jWebMiner: A web-based feature extractor. *Proceedings of the International Conference on Music Information Retrieval*. 113–114.

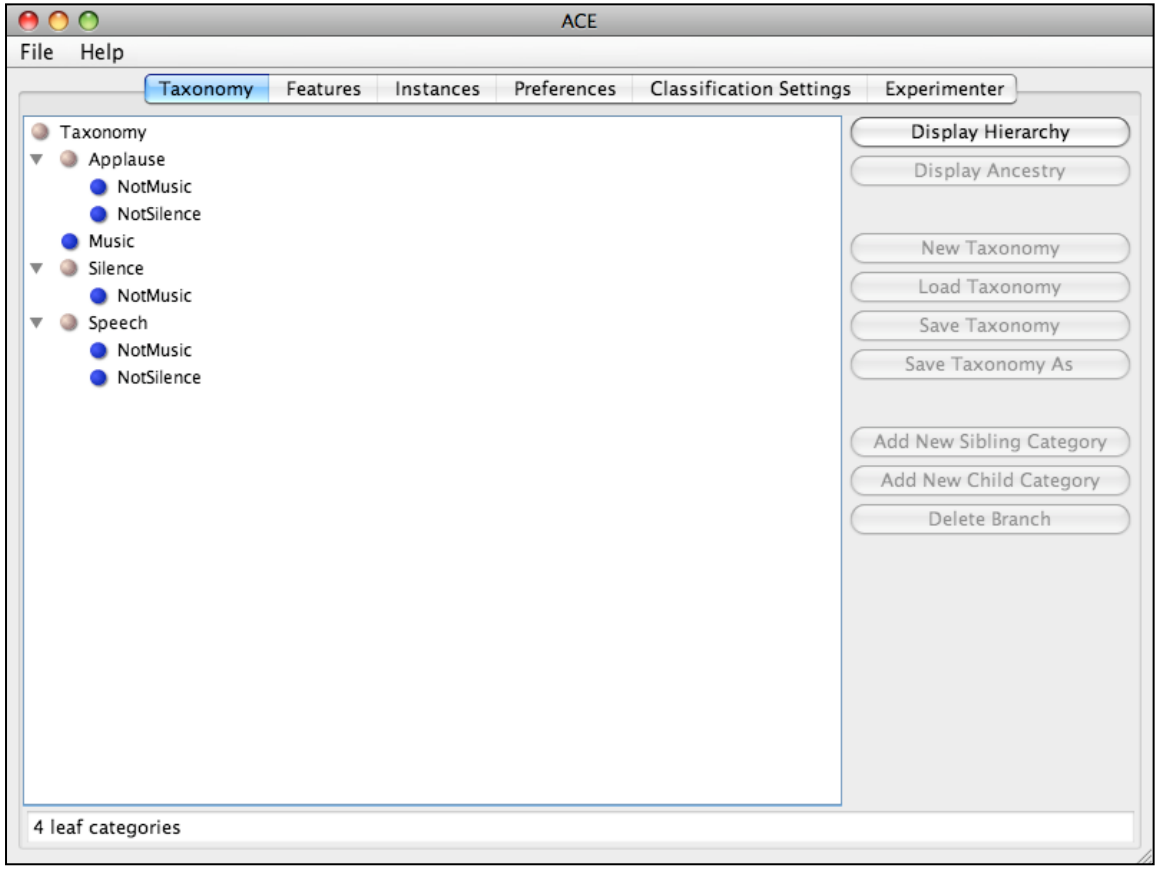
jMIR's jLyrics

- Extracts features from textual lyrics
- Can also automatically generate word frequency profiles for particular classes if training data is provided

- McKay, C., J. A. Burgoyne, J. Hockman, J. B. L. Smith, G. Vigliensoni, and I. Fujinaga. 2010. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. *Proceedings of the International Society for Music Information Retrieval Conference*. 213–218.

jMIR's ACE 2

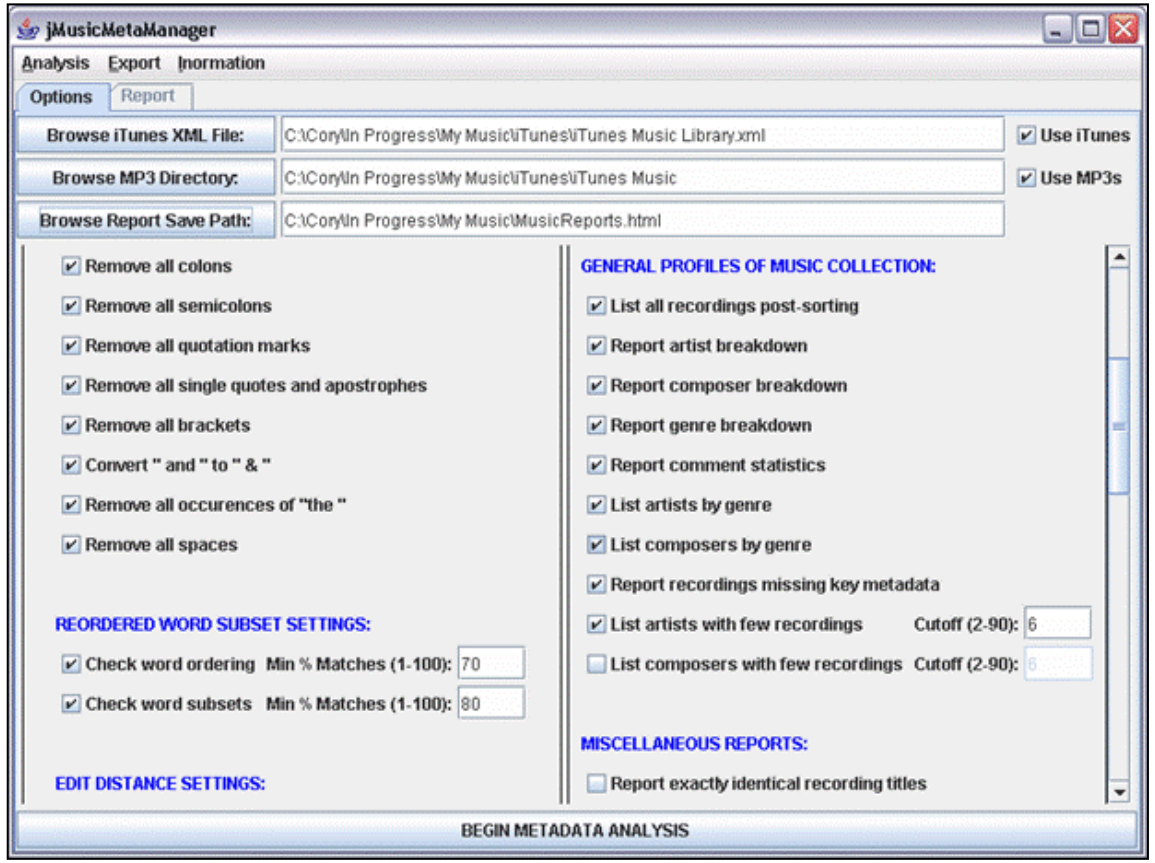
- Automatically uses meta-learning to evaluate the relative suitability of machine learning and dimensionality reduction algorithms for a given problem, in terms of:
  - Classification accuracy
  - Consistency
  - Time complexity
- Jessica Thompson improved the architecture, interface and functionality in version 2



- Thompson, J., C. McKay, J. A. Burgoyne, and I. Fujinaga. 2009. Additions and improvements to the ACE 2.0 music classifier. *Proceedings of the International Society for Music Information Retrieval Conference*. 435–440.
- McKay, C., R. Fiebrink, D. McEnnis, B. Li, and I. Fujinaga. 2005. ACE: A framework for optimizing music classification. *Proceedings of the International Conference on Music Information Retrieval*. 42–49.
- McKay, C., D. McEnnis, R. Fiebrink, and I. Fujinaga. 2005. ACE: A general-purpose classification ensemble optimization framework. *Proceedings of the International Computer Music Conference*. 161–164.

jMIR's jMusicMetaManager 2

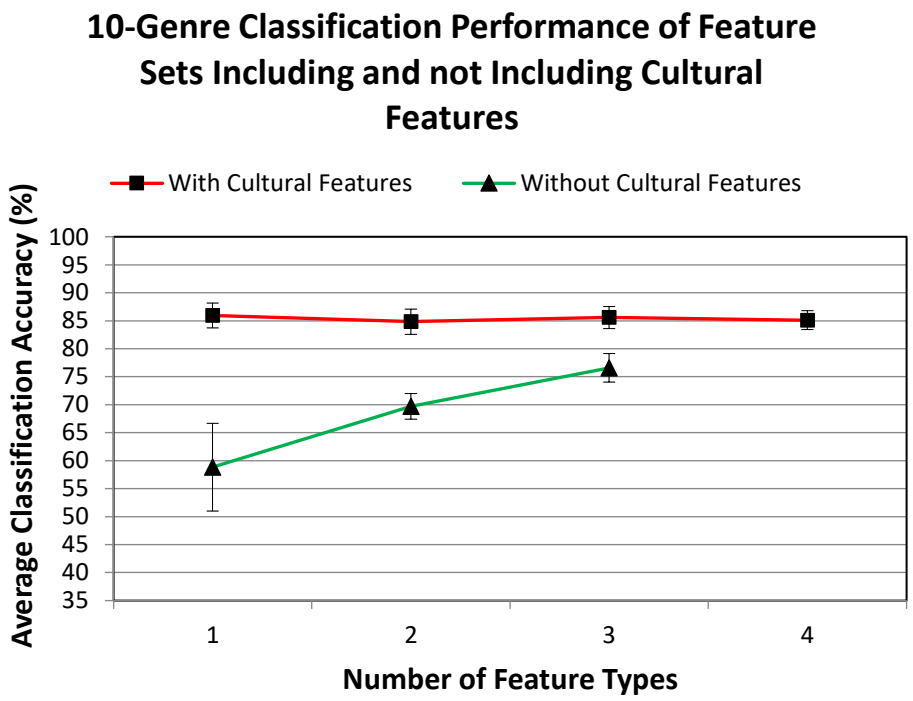
- Auto-detects metadata errors or inconsistencies and redundant copies of recordings
  - e.g. "Charlie Mingus" vs. "Mingus, Charles"
  - Using novel algorithms based on edit distance, replacement and word reordering
- Generates inventory and profile reports
  - 39 reports in all
- Bruno Angeles added fingerprinting (auto-identification of audio) functionality to version 2



- Angeles, B., C. McKay, and I. Fujinaga. 2010. Discovering metadata inconsistencies. *Proceedings of the International Society for Music Information Retrieval Conference*. 195–200.
- McKay, C., D. McEnnis and I. Fujinaga. 2006. A *large publicly accessible prototype audio database for music research. Proceedings of the International Conference on Music Information Retrieval*. 160–163.

Multi-modal analysis (1st approach)

- Extracted features using the four jMIR extractors and trained classifiers on all groupings of them
  - Symbolic, audio, cultural (jWebMiner) and lyrical data were all used
- For tasks like genre classification, adding more feature types improved performance
  - Except when cultural features were included: their predictive power was so strong that adding additional feature types had a negligible effect



- McKay, C., J. A. Burgoyne, J. Hockman, J. B. L. Smith, G. Vigliensoni, and I. Fujinaga. 2010. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. *Proceedings of the International Society for Music Information Retrieval Conference*. 213–218.
- McKay, C., and I. Fujinaga. 2010. Improving automatic music classification performance by extracting features from different types of data. *Proceedings of the ACM SIGMM International Conference on Multimedia Information Retrieval*. 257–266.
- McKay, C. 2010. Automatic music classification with jMIR. *Ph.D. Dissertation*. McGill University, Canada.
- McKay, C., and I. Fujinaga. 2008. Combining features extracted from audio, symbolic and cultural sources. *Proceedings of the International Conference on Music Information Retrieval*. 597–602.

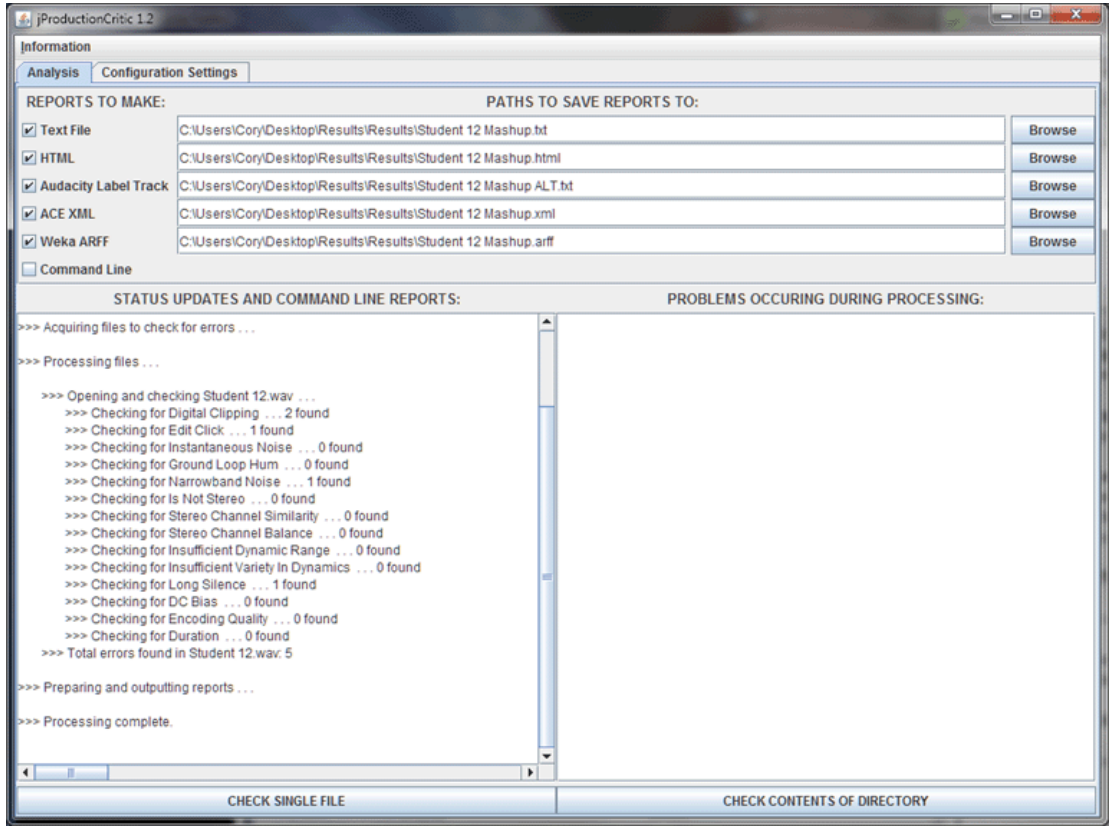
Multi-modal analysis (2nd approach)

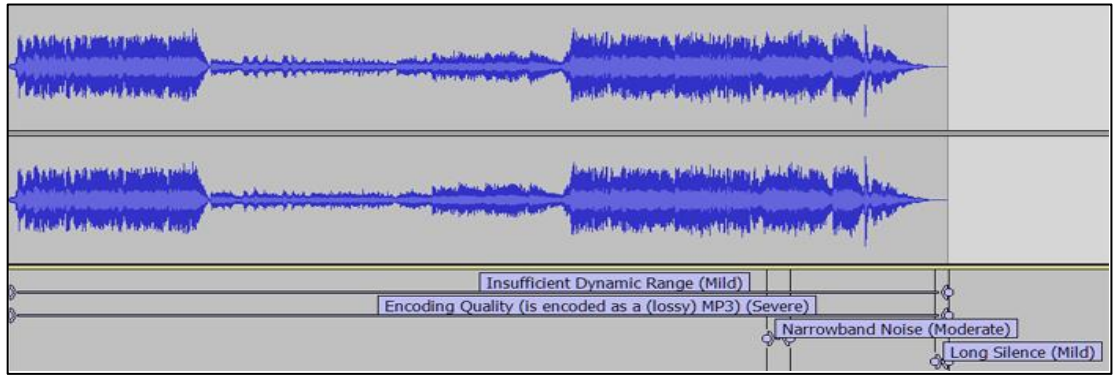
- Used genetic algorithms to explore different combinations of modalities and feature types
  - Features extracted from audio, lyric texts, symbolic scores, album cover images, semantic tags and playlist co-occurrences
  - Evaluated subsets based on "importance," "redundancy" and "stability"
- Found that combining data and feature types often does improve performance, but which ones in particular work best tends to depend greatly on the particular classification problem
  - In general, though, playlist features were often particularly predictive, and album cover image features tended to be the least predictive

- Vatolkin, I., and C. McKay. 2022. Multi-objective investigation of six feature source types for multi-modal music classification. *Transactions of the International Society for Music Information Retrieval* 5 (1): 1–19.
- Vatolkin, I., and C. McKay. 2022. Stability of symbolic feature group importance in the context of multi-modal music classification. *Proceedings of the International Society for Music Information Retrieval Conference*. 469–476.

jMIR's jProductionCritic

- Auto-detects technical sound recording errors, production errors and potential issues in audio files
  - e.g. clipping, edit clicks, background noise, etc.
- Labels audio tracks

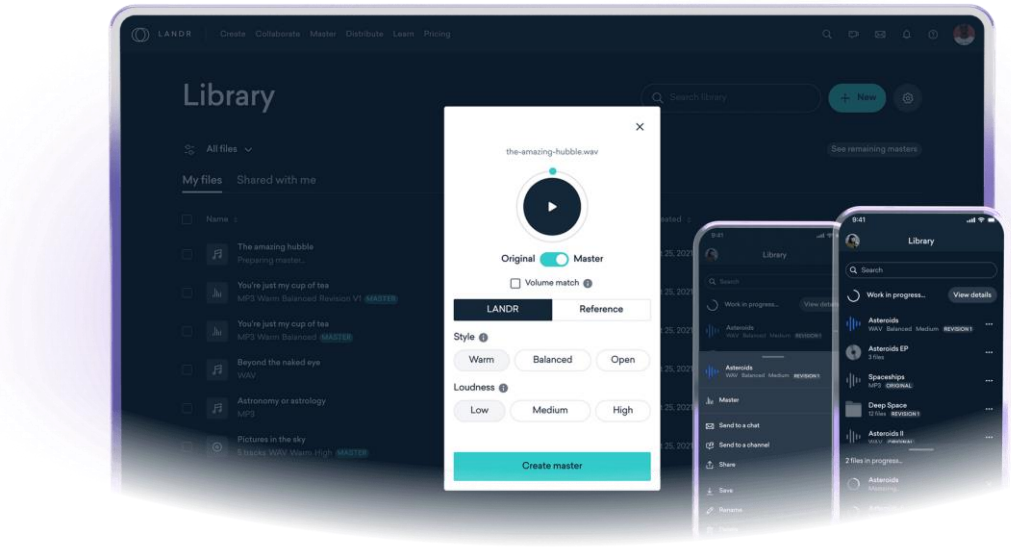




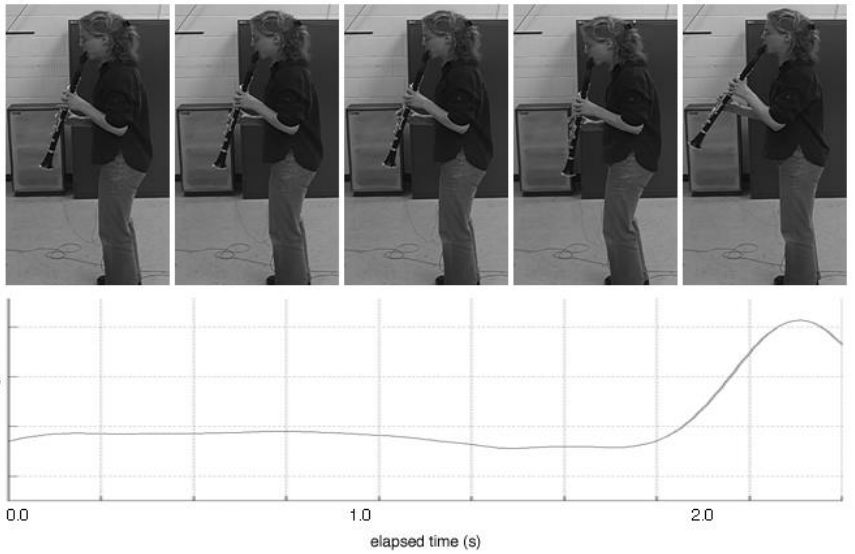
- McKay, C. 2013. jProductionCritic: An educational tool for detecting technical errors in audio mixes. *Proceedings of the International Society for Music Information Retrieval Conference*. 71–76.

LANDR

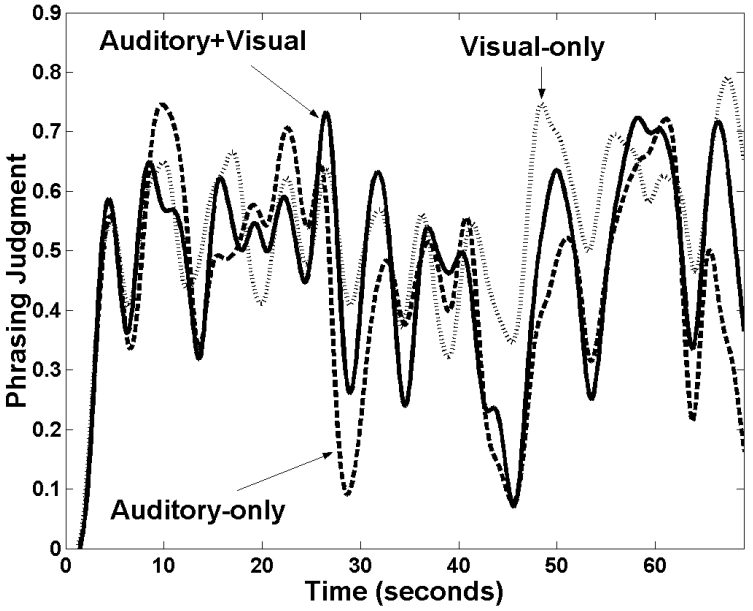
- I have also done commercial consulting work with a Montreal company called LANDR that auto-masters submitted audio
  - i.e. adjusts to aspects like EQ and dynamics to make the music sound better
  - Done in style-appropriate ways
- https://www.landr.com/online-audio-mastering/



Clarinet performance gesture



- Used a range of sensors, cameras and recordings to analyze the motions of clarinet performers to see how meaningful gestures were in communicating information to listeners / watchers
  - Phrasing, expressiveness, rhythm, etc.
- We just used statistical analysis, no AI



- Wanderley, M., B. Vines, N. Middleton, C. McKay, and W. Hatch. 2005. The musical significance of clarinetists' ancillary gestures: An exploration of the field. *Journal of New Music Research* 34 (1): 97–113.

Other musical applications of machine learning

- Music generation / composition
- Hit prediction
- Copyright analysis
- Music recommendation / playlist generation
- Noise removal
- Source separation
  - e.g. separating recorded piano from other instruments embedded in the same recording
- Automatic transcription
  - i.e. generating a symbolic score from an audio recording

- And many, many, many more

Thanks for your attention!

c.mckay@marianopolis.edu

Room B-416, Marianopolis College

Digital Research Alliance of Canada | Alliance de recherche numérique du Canada

Calcul Québec

MARIANOPOLIS COLLEGE

CIRMMT Centre for Interdisciplinary Research in Music Media and Technology

McGill Schulich School of Music / École de musique Schulich

Fonds de recherche Société et culture Québec

SSHRC CRSH Canada