Supervised and unsupervised music classification

Fred Foos and Sebastian Waz

Psych 186B

The problem

Supervised: how can we train a neural network to categorize songs according to guidelines provided by the user?

Unsupervised: how can we train a neural network to classify songs without knowledge of genre and to distinguish new, emergent genres?

Why is it interesting?

- The classification of genres by people is often contentious in and of itself. Perhaps a data driven taxonomy is more effective
- Using a machine to process songs beforehand will allow listeners to more easily find songs which they will probably enjoy
- Music is awesome

What has been done?

---

supervised techniques

- Sage
  - Multiple Linear/Backprop networks using FFT data to classify EDM
  - High Accuracy (80-95%)
  - Used different songs/parameters as us

unsupervised techniques

- Clark, Park, and Guerard
  - Growing Neural Gas and Backprop Neural Net techniques to classify genres like rap, reggae, country, and rock
  - Medium Accuracy (60-85%)
  - Used supervised techniques with unsupervised to be most effective
  - Used similar songs/parameters as us, with similar results

Million Song Database

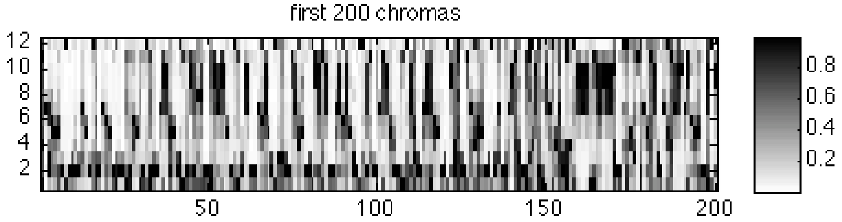
Database online with data about >1 million songs

We obtained a subset of 10000 and imported the data into MATLAB

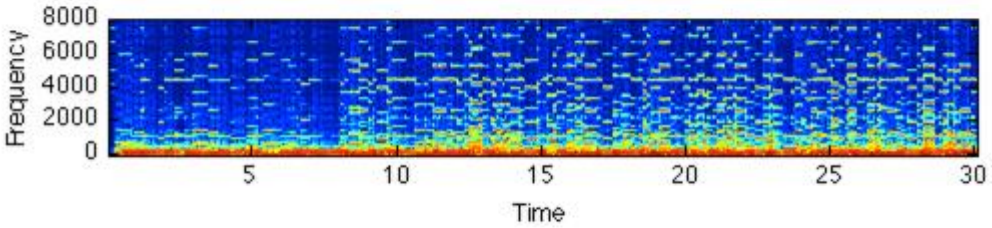
Each song structure has data about a variety of things, like tempo, artist, danceability, beat-onset times, timbre, etc, all easy to access via built in get methods

Chroma and FFT

Chromas are basically time synced compositional information
Segment synced
Easily accessible from db



FFT (Fast Fourier Transform) is a graph of freq over time
Time synced, visualization of timbre



Input Data

A combination of:

- Chroma Raw (1200)
- Chroma Average (12)
- FFT Average (257) - most difficult to extract
- Tempo, Duration, Start of fade out, Loudness, Mean difference between bars, Key, Mode, Time Signature (all 1), Mean Loudness between segments
- Electronic, Rap, Country, and Jazz

All numbers in parentheses are input vector size. Only quantitative data used (no subjective data, like artist tags)

Output: 1 hot, genre synced encoding. Genre obtained online through separate source that corresponded with the database

Why these inputs?

Chroma and FFT give good, large, objective descriptions

Includes timbre and instrumentation

Have given good results in past

The additional aspects vary between songs as well, sometimes good indicators (variable time signature for jazz, consistent tempo for electronic sub genres, etc)

Avoided subjective/string data

Supervised Learning Results

Results of linear and backpropogation models

Network ID	Method	# Songs	# Genres	Data used	Input size	% Accuracy	Chance
1	Linear	300	2	Timbre	1200	69	50
2	BackP	400	2	Timbre	1200	67	50
3	Linear	300	2	Av. Timbre, 4 aspects	16	72	50
5	BackP	400	2	Av. Timbre, 9 aspects	21	82	50
7	BackP	680	4	Av. Timbre, 9 aspects	21	65	25
8	BackP	680	4	Timbre	1200	25	25
9	BackP	400	2	Aver FFT	257	62	50
10	BackP	680	4	Aver FFT	257	25	25
11	BackP	680	4	Av Timbre, 9 apects (z scored)	21	70	25
12	BackP	680	4	9 apects (z scored)	9	47	25
13	BackP	680	4	Av Chroma, Av timbre, 9 apects (z scored)	33	70	25
15	BackP	680	4	Av Chroma, 9 aspects	21	56	25

What we think

More data is likely needed to be effective

- Deep learning for computer vision takes 1000s-millions of inputs to be effective, and chroma/fft can be considered like images

Avoided overfitting, but data fitted to very specific cases - difficult to interpolate chroma/fft data

Variability in the songs made it difficult, as opposed to consistency on different Electronic Genres (see Sage exp)

Not terrible results overall, but shoulda just used EDM

Unsupervised Learning Technique: Self-Organizing Map

Uses a neural structure called a "map," composed of a sheet of nodes

Used to reduce dimensionality

Each node is a vector with dimensionality that matches the map's inputs

Initial values on these vectors are typically random, but may be a gradient

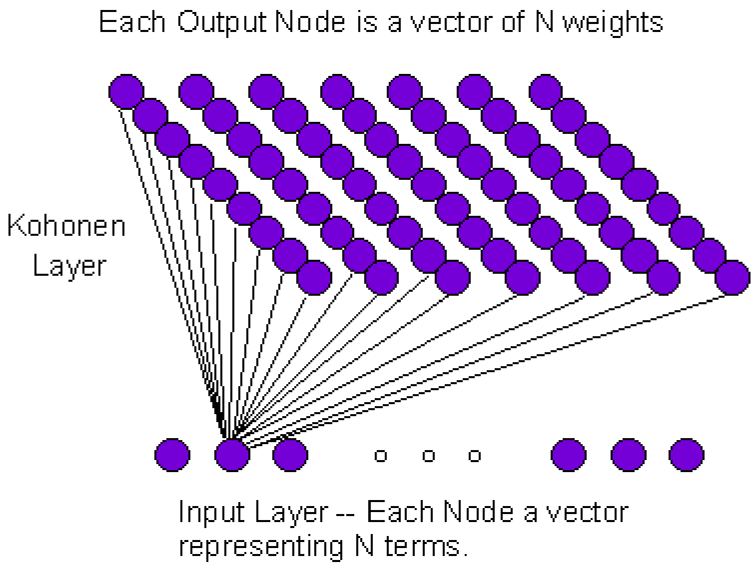


image: http://arizona.openrepository.com/arizona/html/10150/106141/A_Scalable-98.htm

How it works

Every input is compared for similarity to every node in the map (we used cos Θ) to find the best-matching unit (BMU)

A radius around the BMU is calculated to determine the BMU “neighborhood”

The difference between each node and BMU is multiplied by a Gaussian decay and added to each node in the neighborhood

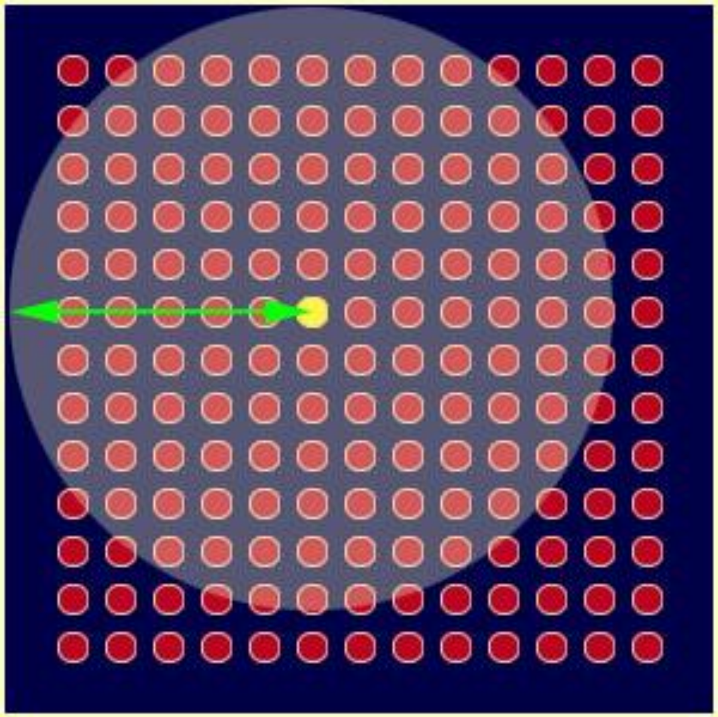


image: http://www.ai-junkie.com/ann/som/som3.html

How it works (continued)

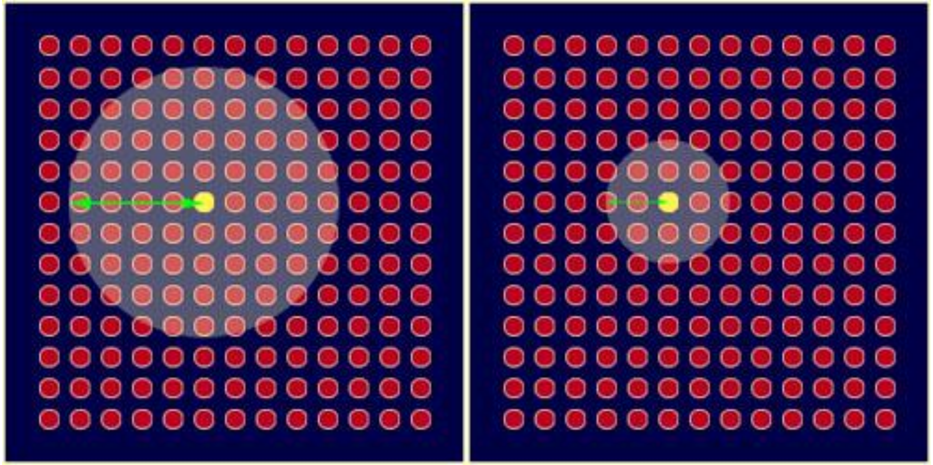


image: http://www.ai-junkie.com/ann/som/som3.html

This process is repeated for each input over several epochs

The neighborhood of the radius decreases each epoch at a specified learning rate. We used:

$$r(t) = r_0 \exp\left(-\frac{t}{l}\right)$$

t is the epoch number
l is a time constant = # epochs * log(map radius)

Proof of concept

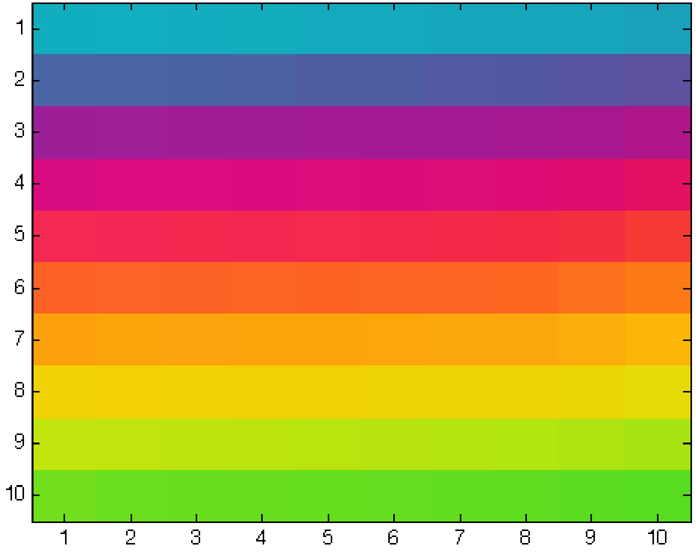
Before applying the map technique to music, we tested our function on RGB inputs

Want to represent 3 dimensional relationships in a 2 dimensional figure:

- easy to demonstrate clustering because digital color is 3D (RGB)

- each row is a class, but the SOM doesn’t know that

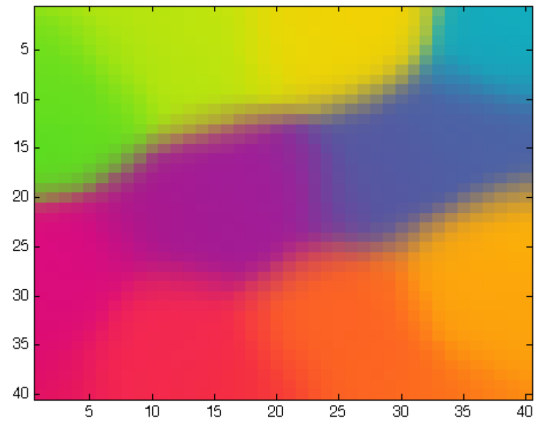
Image of our 100 RGB inputs, presented randomly each epoch:



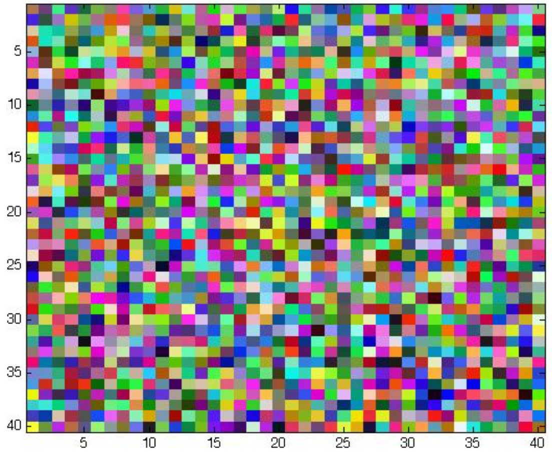
Getting the SOM

Used a 40 x 40 map with randomly initialized 3D nodes, and applied the SOM learning procedure for 50 epochs

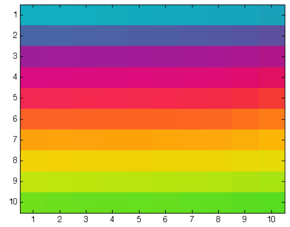
Note how map self-organized into areas which preserve 3 dimensional adjacency and match input colors



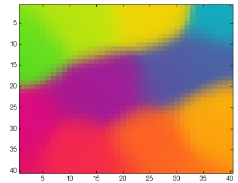
after 50 epochs



inputs:



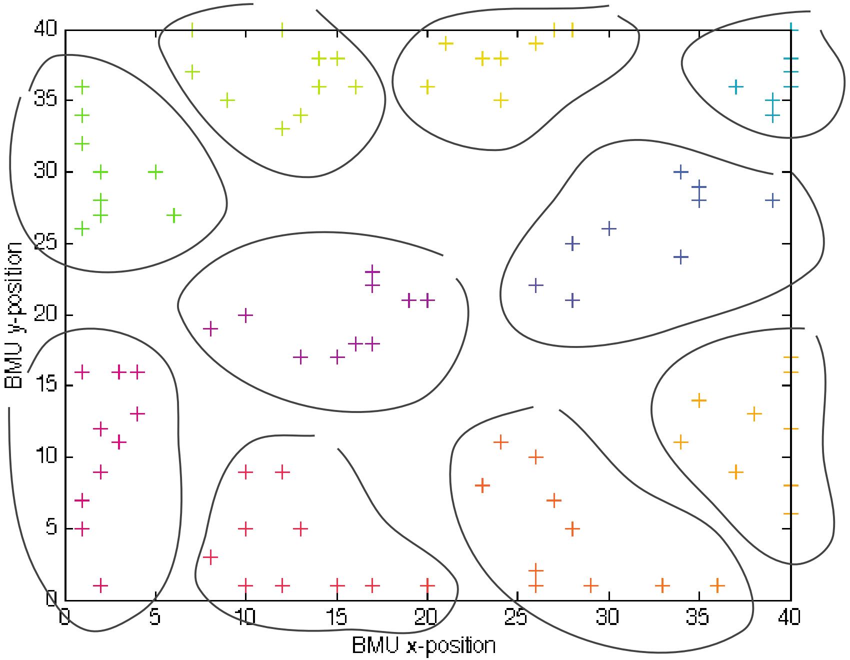
Using SOM to determine classifications



We found a BMU for each input and plotted the BMU's location using the input's color

used k-means algorithm to define clusters (outlines indicate membership)

Used purity to measure goodness of classification

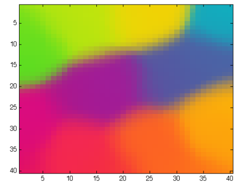


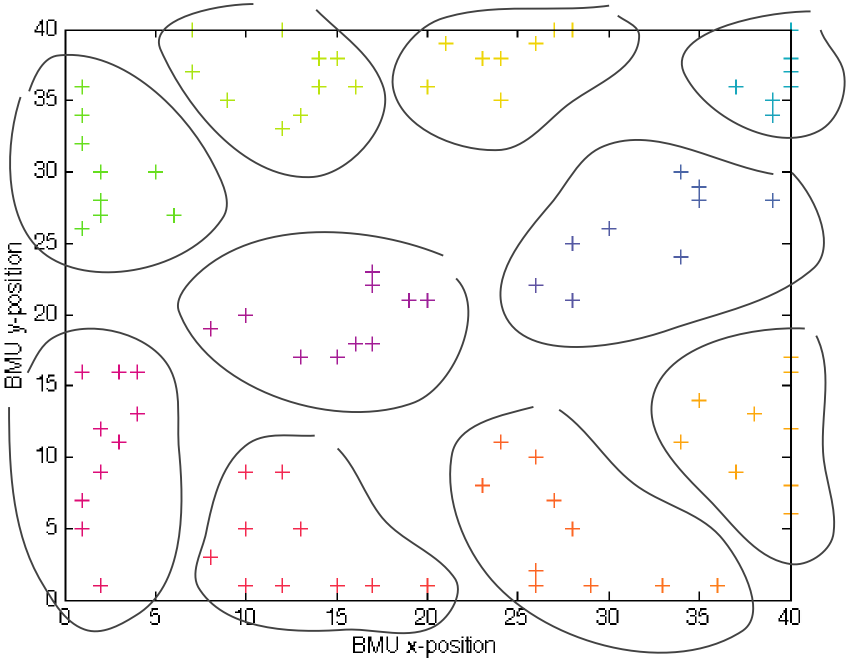
Purity

Purity is one of the easiest measures for quantifying unsupervised learning performance

Equal to number of elements belonging to most frequent input class in cluster, over total number of elements in cluster

Achieved 100% purity on trivial proof of concept





In practice

Music is often described with high-dimensionality, given its many features, which makes it harder to visualize

We applied the same technique using 2 input genres (electronic and rap) to see if the SOM would learn to separate them on its own

Used the same input vector organization as used in the supervised portion of the experiment (mostly used between input z-scores, by element, because this catered best to our definition of similarity for SOM)

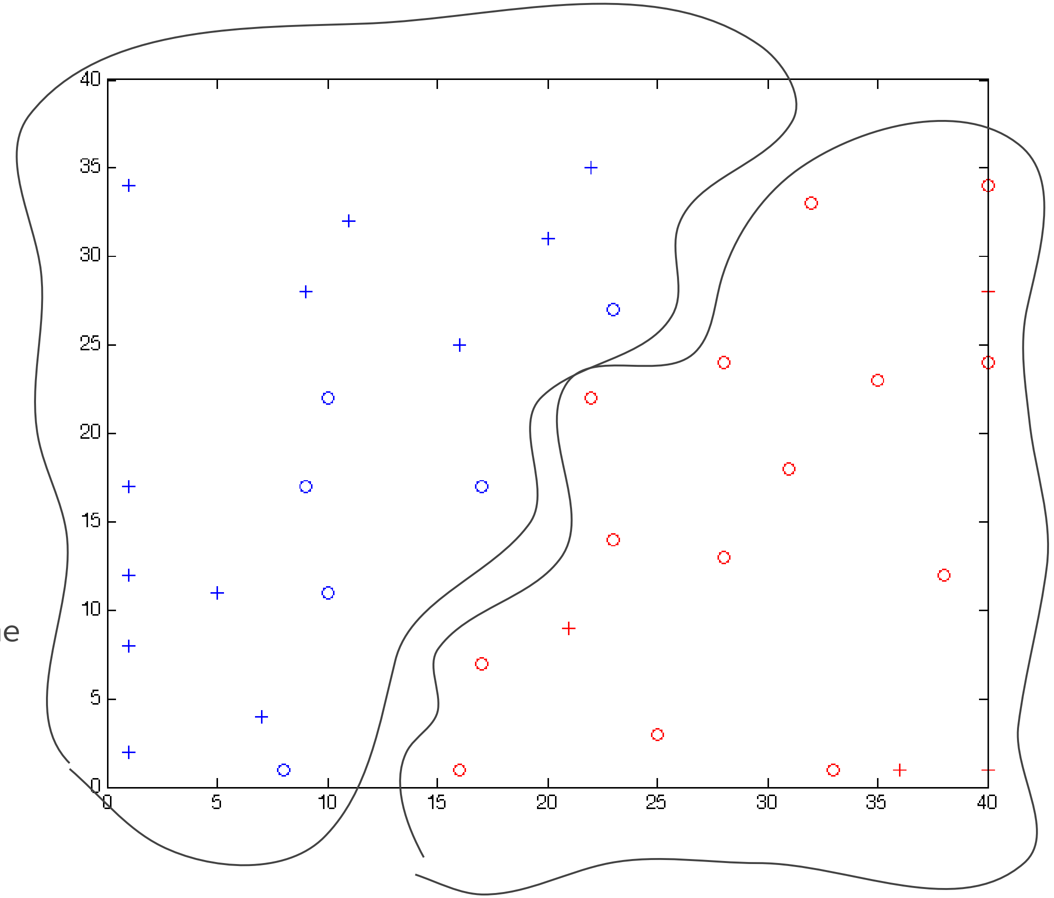
Reduced inputs

First attempt used 40 randomly chosen input songs from the 400 used for supervised learning

Used subset because of time complexity

blue = cluster 1 (16/22, 0.727)
red = cluster 2 (14/18, 0.778)
+ for rap
o for electronic

Note that there is no obvious clustering. Even though SOM put same genres near one another, they were all almost equidistant: therefore clusters may not be correct

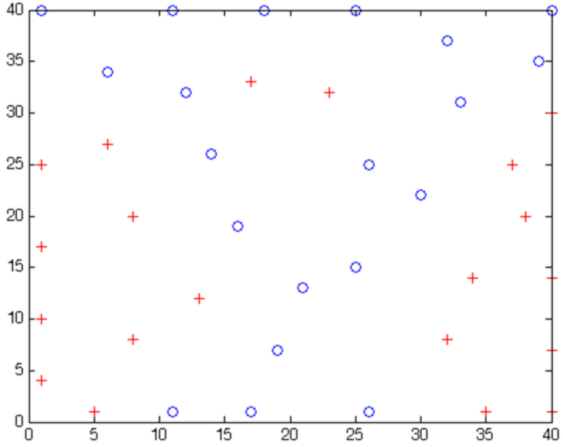


Attempts at improving clustering

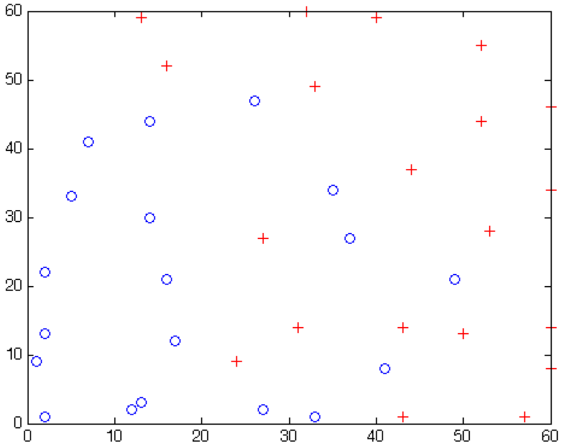
Red: electronic
Blue: rap

if we cannot reliably determine cluster membership, we cannot accurately measure purity

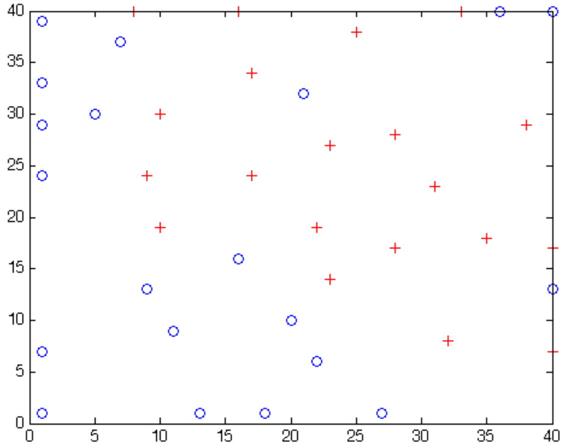
Remove chromas



Increase SOM size



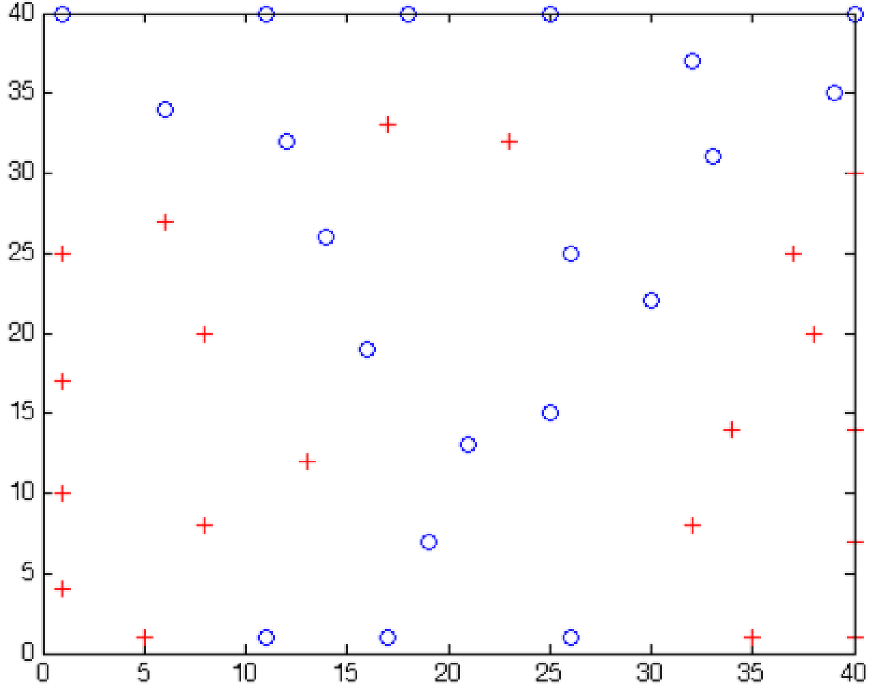
Raw values



What we think

We consistently find that songs from the same genre are placed near one another on the SOM, but never grouped together and away from the opposite genre

We suspect that features such as tempo, key, mode, and time signature keep class members together, but either chroma (compositional) information is so diverse as to prevent them from actually clustering, or size of SOM is too small

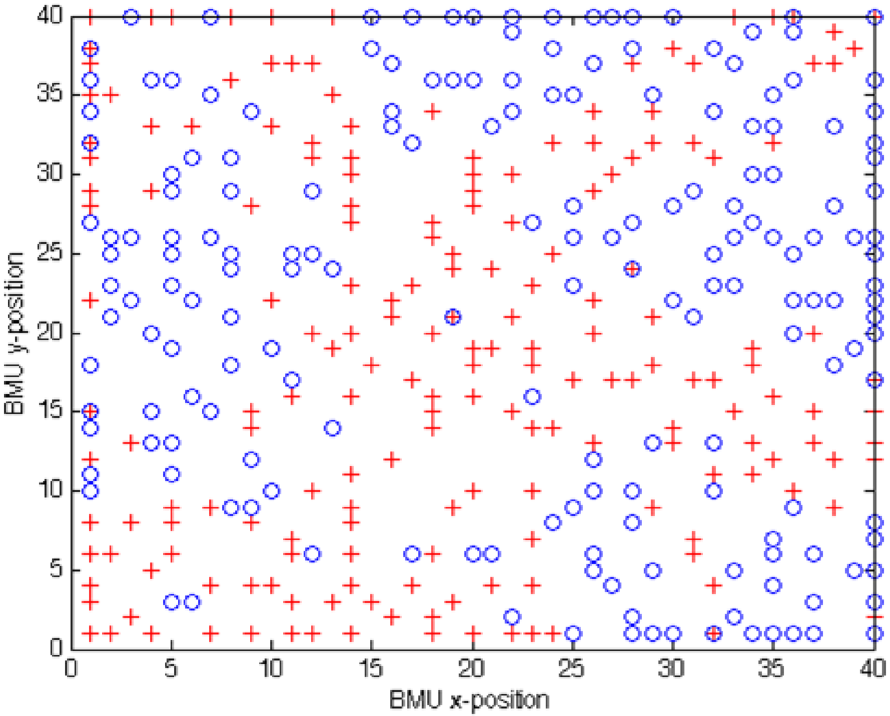


note: +/o agree with color and input classification, no cluster membership indicated

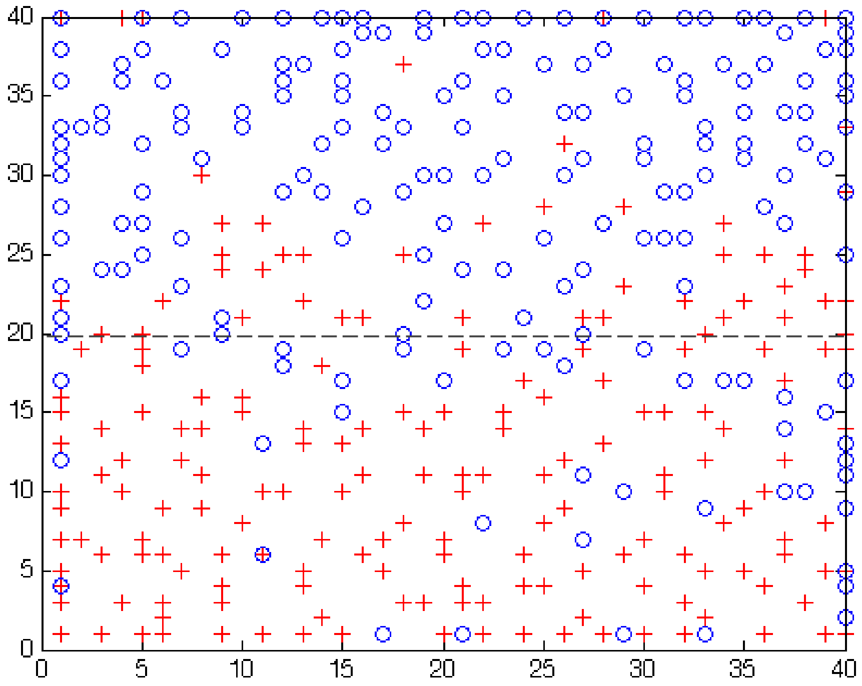
All 400 inputs

Used simple clustering heuristic: y > 20 || y <= 20
(y > 20) purity: 154/198, 0.778
(y <= 20) purity: 156/202, 0.772
Overall purity: 310/400, 0.775

Raw data inputs



z-score inputs



Conclusions

Ultimately: Genre classification is a difficult and subjective task

Questions?

```
Eg: Why did Sebastian get a haircut?
```

All of le sources

Sage's neural nets

Million song dataset and matlab intro, plus Genre Data

Convolutional NNets and Deep Learning in MATLAB

Self Organizing Feature Maps and extra info

Growing Neural Gas paper