CSE 255: Assignment 1 - Exploring Musical Tagging

Hannah Chen (A53045914), Huan Wang (A53055489)

February 23, 2015

Abstract

We explore two predictive tasks: (i) a measure of tag probability, and (ii) identifying a minimum tag set for more meaningful music classification on a 100,000 song dataset joined across complementary databases from the 1 Million Song Dataset ("MSD"). We conclude that a tag set size of around 50 tags is most meaningful and report many of our findings/analysis based on the top 50 tags. Using linear regression to predict tag probability results in roughly a 5% incorrect tag prediction for the top 50 tags.

1 Dataset Description

The Million Song Dataset ('MSD') [6] is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. It started out as a project with The Echo Nest and The Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) at Columbia University. Specifically we focus on two complementary datasets: Last.fm [3], which consists of a collection of song-level tags recognized by Last.fm and precomputed song-level similarity, and The Echo Nest Taste Profile [1], which consists of anonymized listener data in the form of a tuple (user, song, play count) on a subset of the MSD tracks. Finally, we join these with Yahoo! Music data which provides a label for predicting ratings.

Track Description A comprehensive field list consisting of over 50 features for each song datum in MSD can be found at [5]. The features range from basic metadata (artist information, sample rate), to musical qualities (timbre, tatum, tempo), to algorithmic values estimated and defined by The Echo Nest ("hotttnesss", "familiarity", "danceability"). Useful features we've identified for our model are experimented with and explained further in Section 4.

1.1 Last.fm Dataset

Last.fm is a music recommendation service and community that collects data to what songs a user listens. Ultimately, they use this data to analyze what songs/artists are played most often or liked by a user during certain time periods of listening activity. The Last.fm dataset provides song-level tags and precomputed song-level similarity where trackID has been matched to tracks in the MSD.

Statistics

- 943,347 matched tracks between MSD Last.fm
- 505,216 tracks with at least one tag

- 584,897 tracks with at least one similar track
- 522,366 unique tags
- 8,598,630 (track - tag) pairs
- 56,506,688 (track - similar track) pairs

Interesting Findings Tags on Last.fm are user submitted and can range anywhere from genre labels (rock, jazz) to subjective musical descriptions (e.g. happy) to complete garbage. The large 500k+ tag set results in an extreme long tail for tag counts which cannot be meaningfully interpreted. While the top tag "Rock" has been tagged 101,071 times, the bottom 340,000 tags have only been tagged once or twice and have essentially meaningless names like "zzzzznoise" or "My List Of Artists". The average count per tag includes: 16.460929693 (mean), 2 (median), and 1 (mode with 258,523 occurances). This longtail of tags drives our predictive task to find the minimal set of meaningful tags. Interesting figures and tables corresponding to Last.fm include Table 1, and Figs 1, 2, 3, and 4.

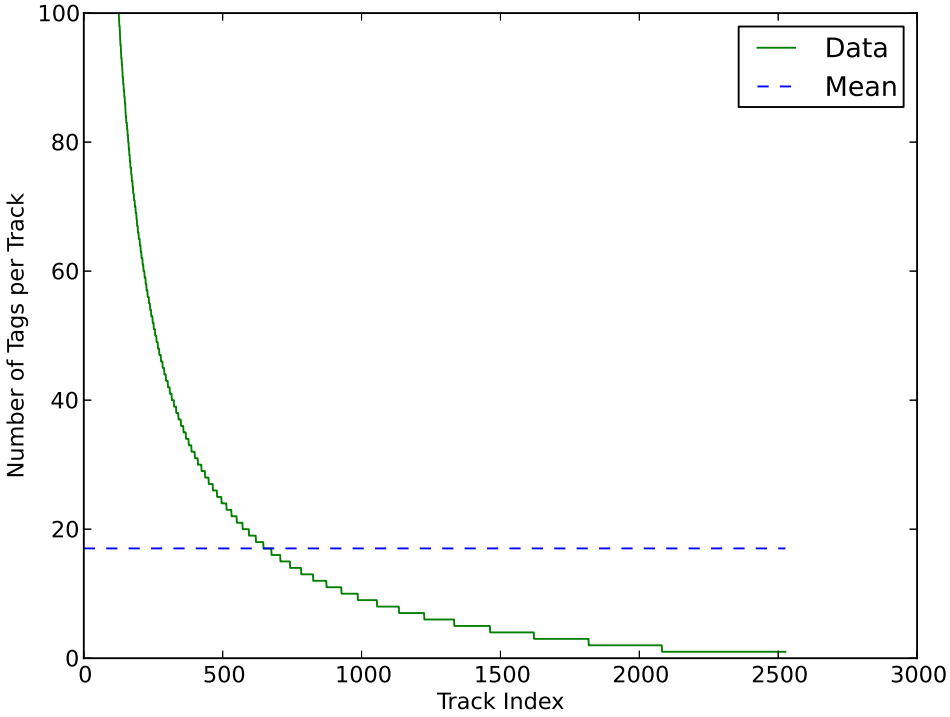

Figure 1: Tags Per Song (Sorted by Tag Count, Intervals of 200)

Metric	Mean	Median	Mode,Occurance	Max	Min
Tags per Track	17.0197103813	6	(1, 88821)	100	1

Table 1: Averages of Tags/Track

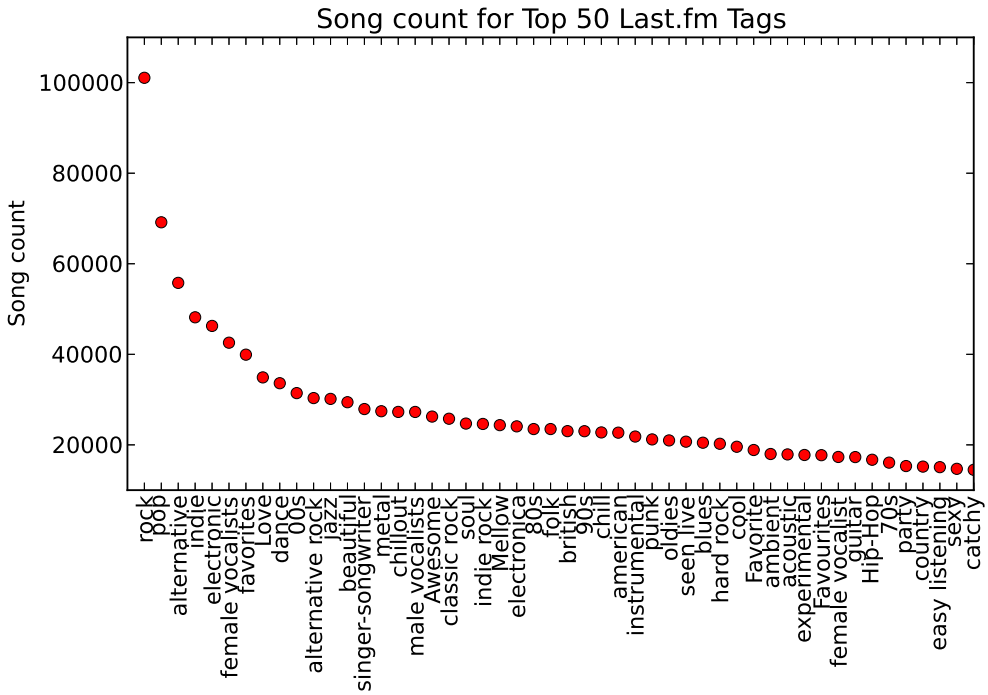

Figure 2: Number of Songs Tagged for Top 50 Tags

1.2 Echo Nest Taste Profile

The Echo Nest is a music intelligence platform that synthesizes billions of musical/audio and user data points to gain an understanding into music behavior/content and listener interaction with music.

Statistics

- 96,747,172 track listens
- 1,019,318 unique user taste profiles
- overlap with 384,546 unique MSD songs
- 48,373,586 (user, song, play count) tuples
- at least 10 unique MSD song activity per taste profile

Interesting tables and figures can be found in Tab 2 and Fig 5

Metric	Mean	Median	Mode,Occurance	Max	Min
Num of total play count per track	360.633690118	32	(1,22084)	726,885	1
Num of different listeners per track	125.794016841	13	(1, 31781)	110,479	1
Num songs listened to per user	136.051990645	73	(15, 12155)	13,132	10

Table 2: Average/Max/Min for (user,track,play count) Triplets

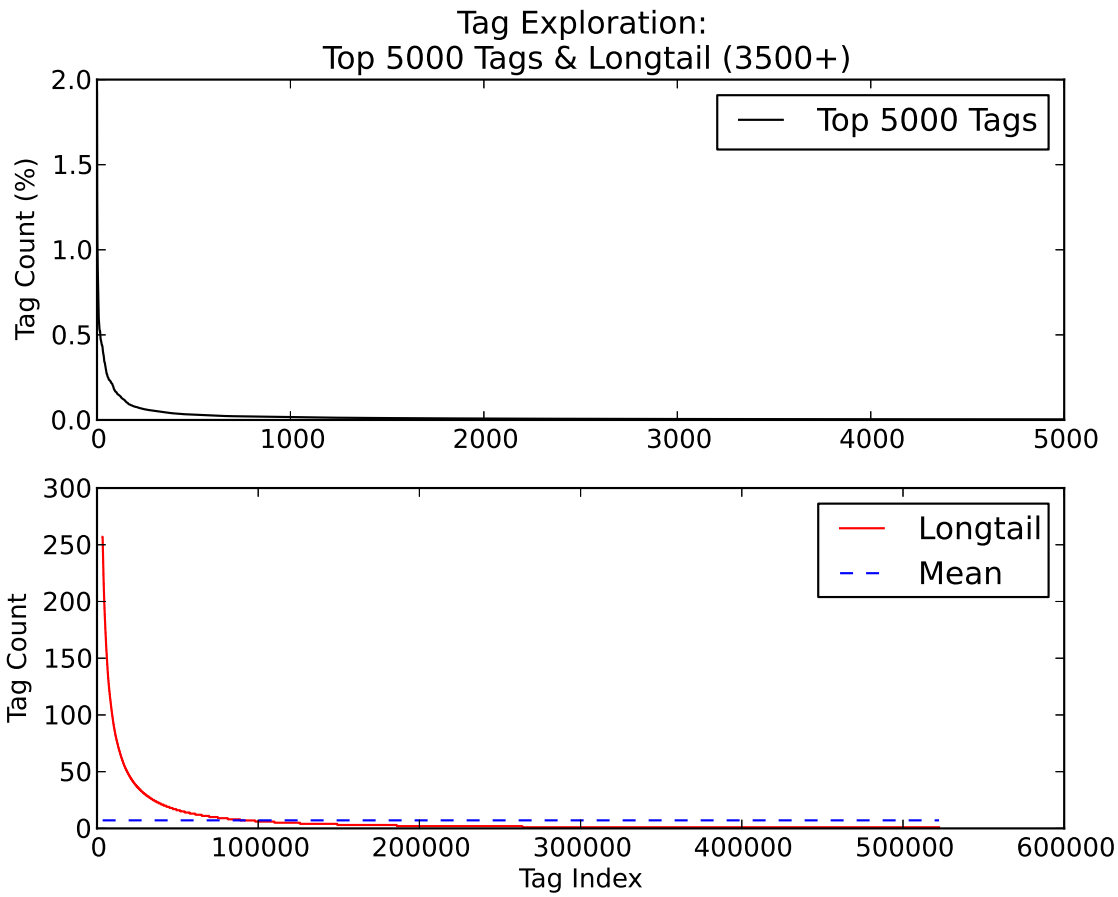

Figure 3: Tag Count (absolute and percentage) for top 5000 tags and longtail (3500+)

2 Predictive Tasks

2.1 Tag Probability

Using track and artist information and listener play count as features, we aim to predict the probability that a tag will appear for each track using logistic regression.

Baseline and Assessment The baseline model is a simple predictor where we simply tag all songs with the top k tags of the training set, where k is the mean number of tags per song. A step further would be to tag all songs using a weighted distribution of the top 300 tags, where $nTerms = 300$ has been pre-calculated by MSD [2]. Validity of the tag popularity prediction will be accessed by percentage of incorrect tags predicted.

2.2 Minimum Tag Set

From our exploratory analysis, we find that over half of the tag set is irrelevant and only used once or twice within the dataset and there is a threshold of 100 tags per track allowed by Last.fm. As tags are user

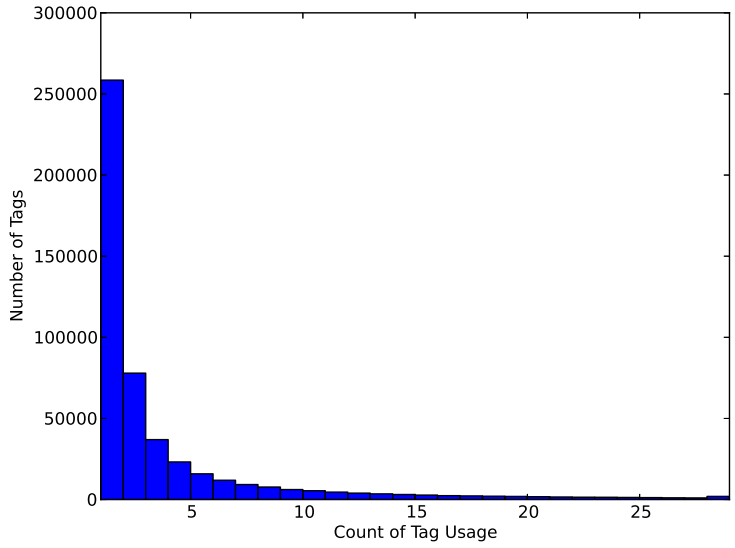

Figure 4: Insignificance of Tags in Longtail

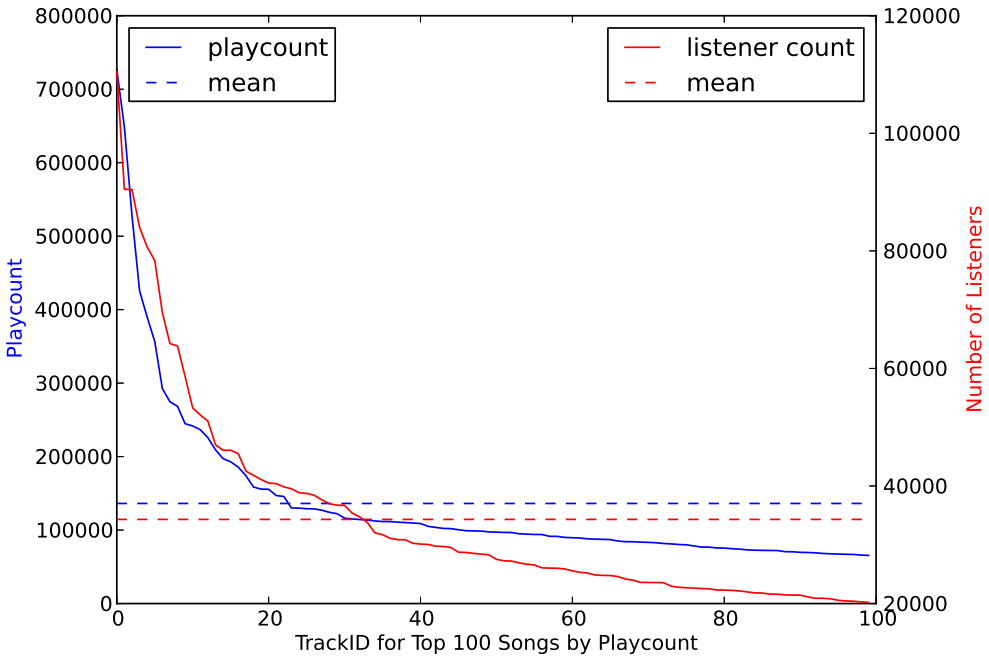

Figure 5: Playcount and Number of Listeners for Top 100 Listened Tracks

submitted data, it would be beneficial to eliminate and prevent useless, one-time tags and can help identify a minimum valuable tags set to be enforced by Last.fm. Additionally, we want to find the upper limit of how many tags are actually relevant to a song because we believe there is a diminishing return in information as more tags are associated with each track. These two tasks try to isolate a sweet spot of tags per track.

Baseline and Assessment We use the same baseline as defined in the previous model Par 2.1. Each tag can be thought of as a dimension, and identifying the minimum tag set is analogous to keeping the features with the highest variance. Finding the minimum tag set can be assessed in two manners: (i) calculation of the Jaccard index to compare similarity between our predicted tag set and the test set, and (ii) comparing our selected predictive features to those of a lower, same-dimensional space with PCA on the test set.

3 Literature

The Million Song Dataset was released in February 2011, and since then a number of contests and projects have taken place to explore the system for recommendation systems as well as an insight into musical qualities. The type of data (user, music item) pairs easily lend way to recommendation systems. Pandora relies on trained musical annotators that manually characterize songs with a music genome they developed to allow for individualized radio play. Moving beyond Pandora and manual labeling, the following projects mentioned instead experiment with music and listener analysis via machine learning. Our predictive task differs from the literature below-we do not attempt to make recommendations or understand tracks/genres based on lyrical data-and thus are not suitable for comparison.

3.1 Recommendation Systems

The MSD Challenge launched on Kaggle and ran from April - August 2012 with a goal to be the "best possible offline evaluation of a music recommendation system" by predicting which songs a user will listen to. A similar contest, KDD-Cup'11, with The Yahoo! Music Dataset occurred a year prior with the task of predicting users' ratings for tracks, albums, artists and genres. Both of these contests have a similar objective to the Netflix challenge, and successful methods to decrease RMSE are shared amongst the three challenges: blending multiple techniques, such as nearest neighbor, restricted Boltzmann machines, matrix factorization and modeling temporary changes in behavior and popularity.

3.2 Additional Projects and Similar Datasets

MSD also includes four community contributed complementary datasets that works with cover songs, lyrics, song-level tags and similarity, and user data. Members at Last.fm [4] have explored the following ideas:

- Analysis of lyrical content and lyric cloud creation, or overlap of words across genres, to compare genres via word popularity
- Genre neighborhood mapping using ideas from search engine ranking, and
- Distinctive word idenfication for genres

4 Features

Feature selection is only experimented with for our first predictive task (tag probability) because our second task aims to identify important features (tags).

4.1 Tag Probability

Out of more than 50 attributes associated with each track in MSD, we use a subset of song hottness, artist familiarity, tempo, artist hotness, key, year, duration, loudness, and mode as our maximum features. We run varying models based on a subset of these, such as features made up of: all track information, all artist information, some derived Echo Nests values, and a mixture of the track/artist features. We estimate that a track's playcount is one of the most important features and test this by comparing the models with and without playcount.

The Echo Nest derives two values: artist "hotttnesss" and artist familiarity. Hotttnesss is a measure for artist popularity calculated based on mentions in social media, music reviews and play count. This measure corresponds to how much buzz an artist currently experiences. Familarity corresponds to how well known in artist is and can be interpreted as the likelihood at a randomly selected user will have heard of an artist.

4.2 Justification from Exploratory Analysis

Though we believe we have a basic understanding of musical tastes due to the experience of listening to music in our personal lives, we had a hard time pre-determining which track features are actually useful. MSD includes track data that corresponds to more musically algorithmic qualities, such as track segmentation into audio features (timbre, tatum), but we did not want to explore (i.e. further process the data) pure audio algorithmic approaches. Some features are wellknown: classical pieces tend to be much longer in duration than top 40 hits which mostly share the same track duration, likely due to the influence of the radio industry. Additionally, indie songs tend to a smaller loudness feature compared to metal or punk songs.

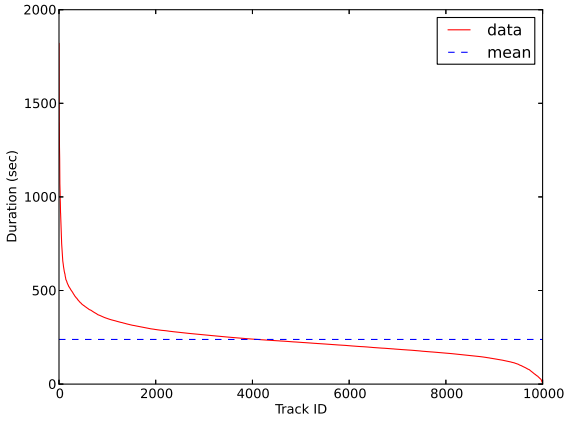

Figure 6: Duration (sec) for MSD Subset (10,000 tracks)

Ultimately, while our exploratory analysis helped us gain insight into the dataset, paving our ideas for our predictive experiments and assessments, we do not believe it helped to justify feature selection for our regression model, leading us to simply try different features and attempt to interpret the feature importance after the fact.

4.3 Pre-processing

The bulk of our data processing consisted of extracting the tracks included in the Taste Profile dataset from MSD and Last.fm to combine information for playcount, tracks, artists and tags. Matching issues of MSD tracks where some tracks were matched to the wrong songs were found within the MSD dataset. The error arises from how data was gathered by MSD: tracks are matched to songs which are subsequently matched to artists. Any mismatched song-tracks thus eventually leads incorrect artist tag and similar artist features. MSD identifes a quick fix to the matching issue by providing a list of song-track pairs that are not trusted, and we have pre-processed our data to remove these song-track pairs. In total, around 5,700 wrong matches have been identified. Additionally, only a subset of artists have known latitude and longitude features; we remove these to more easily evaluate and import the data into our model. Our first predictive task is based on a 50,000 track dataset, while our second predictive task is based on a 987,000 track dataset; each of these are split into a training and testing subset (80-20 ratio).

5 Models

Our main issue was data scalability: simply extracting the dataset was a task in itself and joining multiple datasets to allow for different features type (e.g. track information, artist information, listener data) complicated the process even further. MSD provided a 10,000 track subset to work with that was easily downloadable: this gave us a starting point for testing our code but our regression model yielded the exact same error values regardless of features we tried (ranging from a mix of 1-7 features). Thus we needed to attach an AWS instance to a 300Gb EBS volume and extract at least 50,000 datum points. We attempted to optimize our probabilistic model by using a range of features and comparing the prediction errors. The majority of time spent involved optimizing our performance time and extracting data in a manner that allowed for meaningful exploratory anaysis.

5.1 Tag Popularity

We choose linear regression as our model because we want to fit a numerical value to a varying set of features and can quickly rerun our model on somewhat-arbitrary features within the dataset, like track mode. It is a straightforward model to implement and assess, and leads itself to more extensibility for future experiments, for instance features of nonlinearly processed data can be merged as additional features to empower a regression model. A weakness of regression is the focus on the mean; part of what makes music interesting is how wide ranging it is across all genres and time periods-these extreme features may arise be dampened in linear regression and we end up losing the information we want to model.

We also considered using a naive bayes classifier, but this involves making an assumption about the conditional independence of tags and tracks. We know this is not the case: tags can signify anything from genre to mood to release year and assuming independence on a track level is too fine, as tracks are shared within an album and by artists. Naive bayes is also quite straightforward to implement, but as a generative model it may not be enough when working with our large dataset with high bias.

5.2 Minimum Tag Set

Our second model borrows aspects of distance functions and similarity comparisons from collaborative filtering: we minimize a distance function over all tracks based on varying tag sets. Firstly, we make the assumption that tag importance is ranked based on the number of occurances, as seen in Fig 3. Let k be the number of tags in the tag pool ranging from top 5 to top 500,000 in magnitudes of 10. We then let $g_{tid}(k)$ be the predicted set of tags for each track based on a tag pool of size k. For every track in our dataset, we calculate a distance measure (Jaccard and Euclidean) and take the sum of the distances measures. Our objective function is to find the tag pool size that minimizes this sum:

$$Obj = argmin_k \sum_{tid} (Distance(g(k) - tagset_{tid}))$$

.

We do not use cosine similarity because the vectors use to calculate distance $g(k)$ and $tagset_{tid}$ are not of the same size. The Pearson correlation also does not fit with how we have chosen to define our model and our vectors are not real valued. Based on this process of elimination, we use the Jaccard index and Euclidean distance to measure similarity between our model's tag set prediction and the test set's actual tags.

Alternative models we contemplated include PCA for tag selection and some hierarchical clustering that aggregates tags into super-tags. We choose to forgo these methods because they do not quite fit our defined predictive objective. If we interpret each tag as a feature so that our input data resides in a 550k-dimensional space, PCA would reduce this down to the k dimensions with the highest variance and maximize our data randomness. However, the cost of running PCA is not justified because we already know our tag has an extreme longtail effect, with a majority of tags carrying little to no meaning. It is much easier to simply truncate this tag set arbitrarily than to waste time with PCA. Additionally, the tags are human interpretable; we can quickly glance at the longtail tags and conclude they are useless. If we used PCA to reduce our dimensionality the tags would no longer have an easily interpretable value. Hierarchical clustering of tags into supertags could be meaningful, but we identified it as a next step to our model. An iterative approach of first determining minimal tag set size and secondly aggregating tags could be repeated, but we had a difficult time defining some objective function for this task.

6 Results and Conclusion

6.1 Tag Probabilities

After learning the feature weights from the training data, we set a probabilistic threshold of 0.5 to determine if a particular tag is predicted for each track. Percentage of incorrect tag predictions is used as an error measure, and we see a generally decreasing error rate from 25% down to around 5% for the top 50 occurring tags (Fig 7). In general, the error rate has the same shape regardless of which features were/were not included. We ran a mix of 12 different models based on: single/mix of musical features with/without artist values and year as features. The highly similar shapes across models implies that perhaps the simple MSD features chosen are perhaps too simple and thus not useful for predicting tags. For instance, loudness may be affected by audio encoding type and is not controlled. We do not take the encoding type into consideration to adjust this feature because we have little to no acoustic knowledge; most songs fall within a certain duration. We still believe that a track's musical characteristics affect its tag, but actual audio signals, which we do not explore, may be better indicators. Retrospectively, our hypothesis may have been too open ended because

we tend to mistakenly think of genre and tags to be synonymous. Tags are actually quite broad and may depend on a user's mood at time of tagging, so it is difficult to represent opinions or interpretations based on universally shared track features. Defining our model across the entire tag set may have been too general. If we could make tag predictions within supertags, it is more plausible that musical nuianaces are meaningfully encoded into track features.

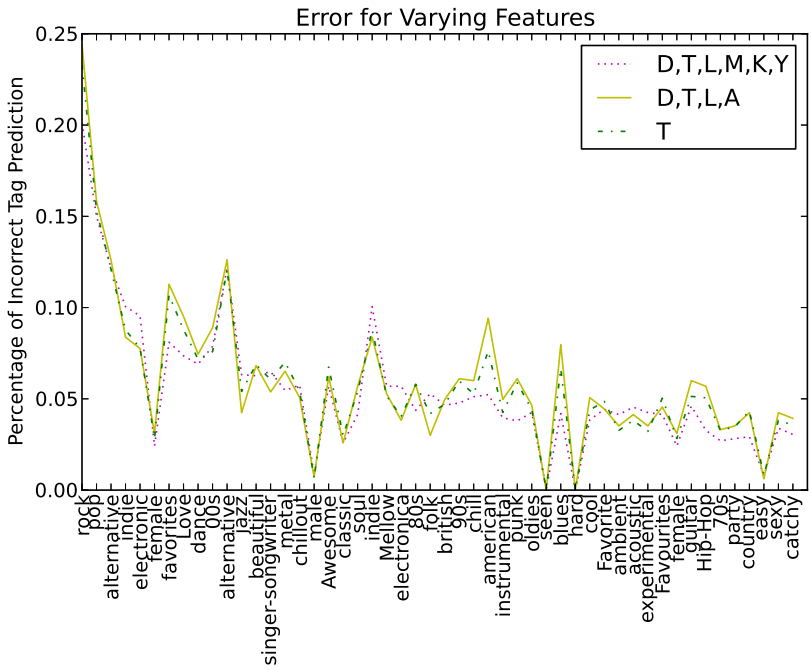

Figure 7: Key: D=Duration, T=Tempo, L=Loudness, Y=Year, A=ArtistInfo

A more interesting question for further study would be to incorporate NLP and use lyrical content, album names and user reviews to better characterize genres. A better dataset labeled by musical professionals, such as Pandora's Music Genome Project may provide more insight on track tagging.

6.2 Minimum Tag Set

Fig 8 shows a comparison of two distance measures (Jaccard index and Euclidean distance) for our predicted tags versus the tags found in the test set. It is useful to note that we plot the sum of the Jaccard indices over all tracks for a range of the size of the tag pool, which is why our y-axis is not between $[0, 1]$. This result verifys that Euclidean distance does favor smaller sets and that the sum of the Jaccard similarity over an increasing tag pool size is convex. From the plot, a tag set around 50 results in the largest in the highest Jaccard measure. This result is not so surprising as it could be easily determined by glancing at our exploratory graphs on tag occurance and tag/song statistics. As with the previous model, our problem definition may have been too general: we are more interested in the purpose of tags and modeling on the entire tag set is uneventful. The ability to differentiate subtags within a tag cluster and determine the optimal number of subtags seems more intuitive; but our original dataset is not presented at that level. Additional pre-processing for hierarchical clustering means adding a layer of complexity onto the true data.

We designed our second mondel because we wanted to test our hypothesis that there is a sweet spot of number of tags to use to best describe tracks. Based on this optimal tag number, tags could be used as a

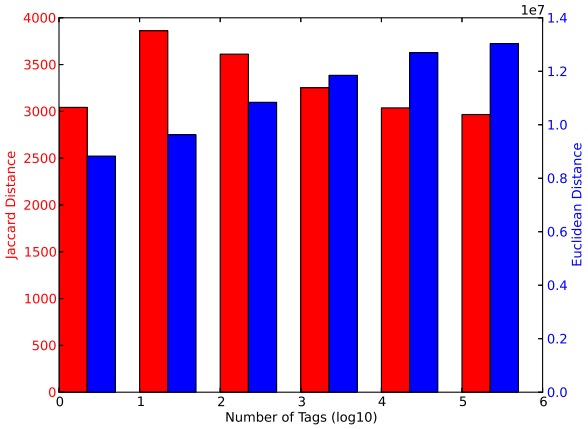

Figure 8: Comparison of Jaccard vs. Euclidean Distance of Minimum Tag Set

feature for further tasks, such as for use in a recommendation system. Our original questions that revolve around tag meanings/importance as feature for tracks lends itself to applying ideas from Lecture 6 on topic modeling of genres represented by multinomial distributions where each track is a mixture over tags and each tag could be represented by a mixture of musical/artistical features.

References

[1] The echo nest. `http://the.echonest.com`. Accessed: 2015-2-10.

[2] Github: Tasksdemo/tagging.

[3] Last.fm dataset. `http://labrosa.ee.columbia.edu/millionsong/lastfm`. Accessed: 2015-2-10.

[4] Lyric cloud, genre maps and distinctive words. `http://blog.last.fm/2011/06/22/lyric-clouds-genre-maps-and-distinctive-words`. Accessed: 2015-2-20.

[5] Msd track description field list. `http://labrosa.ee.columbia.edu/millionsong/pages/field-list`. Accessed: 2015-2-10.

[6] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.