CS229a Final project: Automatic Song Tagging

Jose Carvallo, Elie El Noune, Jorge Herrera

12/15/2011

1 Introduction

Automatic song tagging remains one of the hot topics in the area of Music Information Retrieval (MIR)[1]. Indeed, music listeners always look for new ways of searching for music, and labeling their music libraries. They also are very involved in the process of labeling music and contribute everyday to expanding the wealth of online tags through Internet radios like Last FM and Pandora. Our project, "Automatic Song Tagging" – a.k.a. AutoTag, applies the machine learning algorithms we have learned in class to automatically generate audio tags. Although this is a classification, it is certainly not a simple one, since dealing with audio data adds another dimension of complexity. In fact, the training data for our model is time series of feature vectors, which will have to be compressed to one or a few vectors per song. Furthermore, another problem is gathering ground truth data that is reliable. In this report, we summarize the progress we have made during the last three weeks exploring the literature and experimenting with the CAL500[5] data set, Logistic Regression and Neural–Networks, which were rarely used from the literature we surveyed. We explain the procedure we used to choose this data set, how and why we preprocessed the features and labels, and the results obtained with our classifiers.

2 Literature Review

In order to become more acquainted with the topic, we read some papers on automatic music tagging, such as [1], [2], [3] and [4], to cite a few. The root of the problem is not in choosing the right algorithm itself, but rather in choosing the right features, and the right training data. Among the most popular features used were the Mel Frequency Cepstrum Coefficient (MFCC) and delta MFCC. We found algorithms like Support Vector Machines and Logistic Regression to be among the most popular. The performance of the algorithms was highly dependent on the vocabulary of tags, and the quality of the label data used. This particular detail alarmed us to search for a reliable data set.

3 Dataset

At the beginning of the quarter, our intent was to apply a simple learning algorithm on a huge data set, like the Million Song Data set (MSD), which had just been released. We considered algorithms like Logistic Regression, Neural Networks, and SVM. Concerning the tags labels, we were considering the Echo Nest Million Song Data set, the CAL10k (10,000 songs) and the CAL500 (500 songs), hosted by Columbia University and UCSD respectively. The last dataset was popular in the literature we have read, and presented 500 songs and a vocabulary of 174 tags, which is a good start to get a feel for the data, and good for comparing with the rest of

literature. As for the learning algorithm, professor Ng's advise was that the most of the time, a lot of time is wasted trying to determine which algorithm performs best on a certain task, and that it's much more optimal to pick any one of the popular algorithms and to keep improving on it. For this reason, we chose to work with Logistic Regression on the CAL500 dataset at first, and only once we had enough insights on the data and the problem we decided to switch to Neural–Networks.

4 Experiment

4.1 Initial approach

For the initial exploration we decided to use Logistic Regression classifiers, because it is a simple yet powerful algorithm. This way we were able to quickly gain insightful information about the data and the problem itself.

We approached the problem as a multi-label classification problem. Therefore, we trained a set of binary classifiers, each one trained to classify an individual tag. Instead of using the traditional one-vs-all method, which makes sense when an example (i.e. a song) can only have one tag applied to it, we are simply running all the classifiers in parallel, and selecting all the tags that apply to the song analyzed.

The results of this first approach were reported in the Milestone Report submitted on November 18th, 2011.

4.2 Final design

The initial exploration made us realize of some peculiarities of the dataset we are working with. After analyzing the results obtained using Logistic Regression, we noticed that the results reported were very good because the dataset was very skewed. In other words, for each tag there were a few positive examples and many many more negative examples. Formally:

$$\sum_{i=1}^{m} y_k^{(i)} \ll \sum_{i=1}^{m} (1 - y_k^{(i)}), \forall k \tag{1}$$

where $y_k^{(i)} = 1$ if the i–th training example for the k–th tag is a positive example and $y_k^{(i)} = 0$ if the i–th training example for the k–th tag is a negative example. Therefore, the classifiers were predicting 0 all the time, but since most of the examples in the validations set—as well as the test set—were negative examples (the condition in eq. 1 was also true for both these sets), this "constant" prediction was achieving an artificially high validation accuracy. This fact was confirmed by analyzing the True Positives in all the trained classifiers, which were 0 for almost all of them. Conversely, the number of False Negatives was equal to the number of positive training examples for almost every tag.

Considering this observation, we made some decisions to improve the results of our classifiers:

Use only tags with positive and negative examples The Cal500 dataset contains 174 tags. These are grouped into categories, such as *emotions*, *genre*, *instrument* and others. Songs annotated with a tag are considered positive examples of such tag and all the rest of the songs in the dataset are considered negative examples. As stated before, this produces a skewness problem. Fortunately, the tags in the *emotions* group, are paired between the emotion and the negated emotion. In other words, there is tag labeled as *Emotion-Sad* and a counterpart labeled *NOT-Emotion-Sad*, for example. We decided to focus exclusively on

this group of tags. Out of the 174 original tags, 36 belong to the Emotions group, (18 pairs of tags/NOT-tag).

The positive examples of a tag (e.g. *Emotion-Sad*) are the positive examples for the classifier and the positive examples of the corresponding negated tag (e.g. *NOT-Emotion-Sad*) are considered as the negative examples in the set.

Use different training sets for different tags As a side effect of the previous decision, we were left with different sets of songs for each tag. Not all the songs that appear as either positive or negative examples of a particular tag will necessarily appear in the set of songs tagged as either positive or negative example for a different tag.

Train Neural–Networks (NN) for the classification task As mentioned in the previous report, Logistic Regression was used initially because it provides a simple and easy classifier to train, but we wanted to train Neural–Networks, mainly because after our literature review they don't seem to have been used for the automatic tagging task in the MIR field. We also decided to use a single hidden layer, but to try different size for this layer. Later we describe how we determined the optimal number of units in the hidden layer.

Train single-output NNs Finally, since each tag has a different set of songs to be trained on, instead of attempting to train a single NN with 18 outputs, as typically used in a multi–label classification problem, we decided to train 18 different NN, each one with a single output.

4.3 Data preprocessing

While initially we worked only with the 12 chroma features included in the set, for this final training we used the 12 chroma features plus the 39 Delta–MFCC features plus the 52 Dynamic–MFCC included in the dataset. Summarizing, we used 103 features.

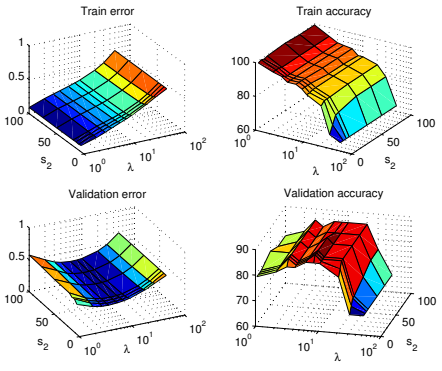
The first preprocessing done was feature normalization, basically removing the mean and normalizing the variance to 1 for each feature independently. After feature normalizing, we applied PCA to reduce the number of features. We kept the first 51 principal components, which account for 94.8% of the variance.

For each tag, we split the set into a training (60%), validation (20%) and test (20%) subsets.

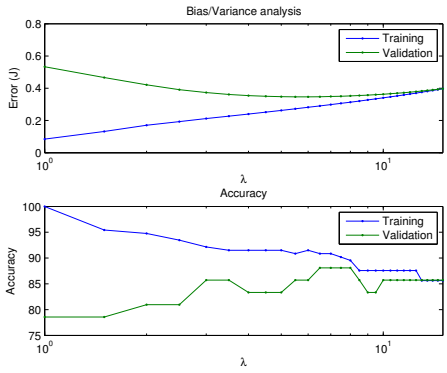
4.4 Training details

Since after applying PCA we were left with 51 features, we used 51 units in the input layer ($s_1 = 51$). As mentioned before, we trained single–output NN with a single hidden layer ($s_3 = 1$). Since the number of units in the hidden layer (s_2) is a parameter that needs to be specified, as well as the value of the regularization parameter (λ), we decided to train the NN using different combinations of these parameters and plot the error and accuracy (for both the training and validation sets) as 3D surfaces in Matlab. Using these figures we were able to identify the best combination of λ and s_2 (see Figure 1(a), for example). For this step, we used 500 for maximum number of training iterations (`MAXITER`). Once the optimal values for λ and s_2 were identified, we retrained the NNs setting `MAXITER` = 2000, to make sure that the weights of the NN ($\Theta^{(1)}$, $\Theta^{(2)}$) have converged.

In general, we discovered that for $s_2 \geq 20$ there was no significant change in either the error or the accuracy, so we set $s_2 = 20$ for all the classifiers. For λ, on the other hand, each classifier seemed to have a distinct optimal value. Moreover, both accuracy and error were sensitive to small changes to λ. Therefore, we did a second set of plots, this time in 2D, to determine the



(a) Tag: Arousing/Awakening



(b) Tag: Arousing/Awakening

Figure 1: Optimal s_2 and λ identification: (a) 3D plots to determine the optimal s_2 and λ; (b) 2D plots for fine tuning the optimal λ

optimal value for λ. This time we used a narrower range (derived from the 3D plots generated earlier), but using more resolution in the regularization parameter space (see Figure 1(b), for example). Using these plots we determined the optimal λ.

5 Results

Finally, we were able to train all 18 classifiers, getting good overall results. In two of the classifiers (*Bizarre/Weird* and *Light/Playful*) we still had a skewness problem, basically because the "negated tag" count was far larger than the positive examples for those tags. Figures 2 and 3 show the train, validation and test F_1–scores and accuracies for the other 16 classifiers trained.

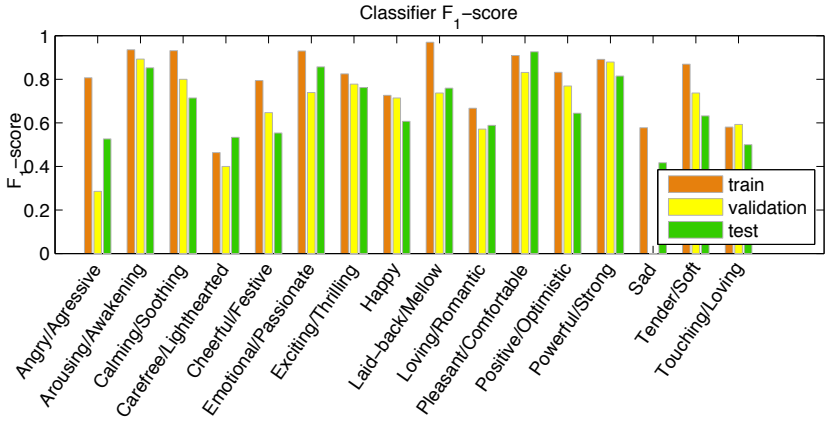


Figure 2: F_1–scores for different classifiers

Table 1 summarizes the results for previously "unseen" samples in the test sets (i.e. they were not used at all during the preliminary analysis, parameter selection or the final training stages):

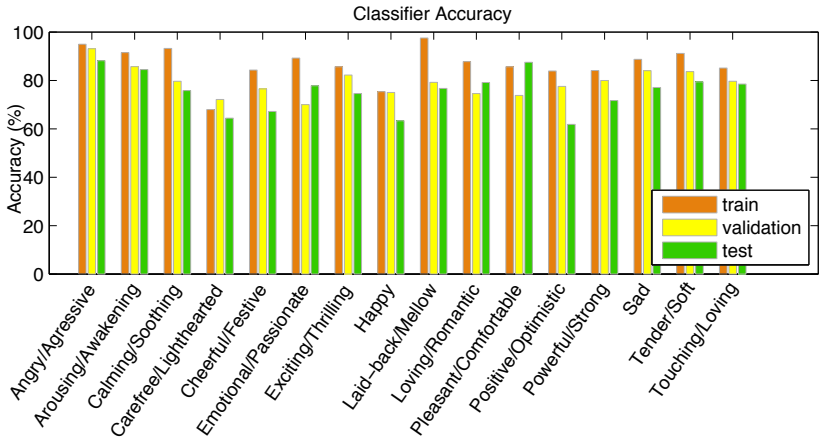


Figure 3: Accuracies for different classifiers

As can be seen in these Figures and Tables, we successfully trained 16 classifiers to automatically determine emotion tags for songs using exclusively audio features.

Table 1: Results summary: only F_1–scores and Accuracy for the test set are shown.

Tag name	# training examples	# validation examples	# test examples	F_1–score	Accuracy
Angry/Aggressive	219	72	76	0.52632	0.88158
Arousing/Awakening	153	42	58	0.85246	0.84483
Calming/Soothing	177	59	66	0.71429	0.75758
Carefree/Lighthearted	159	43	59	0.53333	0.64407
Cheerful/Festive	172	51	64	0.55319	0.67188
Emotional/Passionate	139	40	54	0.85714	0.77778
Exciting/Thrilling	160	45	59	0.7619	0.74576
Happy	162	48	60	0.60714	0.63333
Laid-back/Mellow	162	48	60	0.75862	0.76667
Loving/Romantic	180	59	67	0.58824	0.79104
Pleasant/Comfortable	153	42	56	0.92632	0.875
Positive/Optimistic	143	40	55	0.64407	0.61818
Powerful/Strong	132	40	53	0.81481	0.71698
Sad	168	50	61	0.41667	0.77049
Tender/Soft	181	61	68	0.63158	0.79412
Touching/Loving	175	54	65	0.5	0.78462

6 Conclusion

Although Neural–Networks haven't been used previously in the task of automatic tagging of music, we have shown that they provide a good alternative to the commonly used methods, giving comparable results.

The problem of data skewness was addressed by selecting only emotion tags, for which the number of positive and negative examples is comparable. In future work we plan to address this issue in a more general way, so we can train classifiers for other tags.

References

[1] T. Bertin-Mahieux, D. Eck, and M. Mandel. Automatic tagging of audio: The state-of-the-art. In Wenwu Wang, editor, *Machine Audition: Principles, Algorithms and Systems*. IGI Publishing, 2010.

[2] Wei Bian Bo Xie, Dacheng Tao, and Parag Chordia. Music tagging with regularized logistic regression. In *12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[3] G. Marques, M. Domingues, T. Langlois, and F. Gouyon. Three current issues in music autotagging. In *International Society for Music Information Retrieval Conference*, Miami, 2011.

[4] Chris Sanden and John Z. Zhang. An empirical study of multi-label classifiers for music tag annotation. In Anssi Klapuri and Colby Leider, editors, *ISMIR*, pages 717–722. University of Miami, 2011.

[5] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Towards musical query-by-semantic description using the CAL500 data set. In *ACM Special Interest Group on Information Retrieval Conference (SIGIR '07)*, 2007.