# MUSIC TAGGING WITH REGULARIZED LOGISTIC REGRESSION

**Bo Xie**
GTCMT
Georgia Tech
Atlanta, GA, USA
bo.xie
@gatech.edu

**Wei Bian**
QCIS
Univ of Tech Sydney
Sydney, NSW, Australia
brian.weibian
@gmail.com

**Dacheng Tao**
QCIS
Univ of Tech Sydney
Sydney, NSW, Australia
dacheng.tao
@uts.edu.au

**Parag Chordia**
GTCMT
Georgia Tech
Atlanta, GA, USA
ppc
@gatech.edu

## ABSTRACT

In this paper, we present a set of simple and efficient regularized logistic regression algorithms to predict tags of music. We first vector-quantize the delta MFCC features using k-means and construct "bag-of-words" representation for each song. We then learn the parameters of these logistic regression algorithms from the "bag-of-words" vectors and ground truth labels in the training set. At test time, the prediction confidence by the linear classifiers can be used to rank the songs for music annotation and retrieval tasks. Thanks to the convex property of the objective functions, we adopt an efficient and scalable generalized gradient method to learn the parameters, with global optimum guaranteed. And we show that these efficient algorithms achieve state-of-the-art performance in annotation and retrieval tasks evaluated on CAL-500.

## 1. INTRODUCTION

Automatic tagging of music is a popular topic in recent years, with applications in music information retrieval, description of music, etc. The task is to associate a song with a few relevant labels (or tags), e.g. pop, male vocal and happy. We want to predict confidence values that accurately estimate the strength of the association between the labels and audio contents. Given a song, these confidence values can be used to rank the tags by relevance, and this is the music annotation task. In the music retrieval task, we rank the songs according to their relevance to a specific query tag.

The challenge mainly lies in two parts. One is how to represent a song or a song segment that best summarizes its content. The most popular audio feature is the Mel-Frequency Cepstral Coefficient (MFCC) that only describes a 23ms time window. While these very short "frames" cannot be used directly as features for songs, they make up the building blocks for more advanced features. [7] summarized the frame-level features over a segment by means and covariances and other features were combined by AdaBoost. Spectral covariances over a segment were also proposed and achieved better results than means and covariances of MFCC [6]. Other methods tried to estimate the probability distribution of the MFCC feature space and use this as song-level features [1, 3]. At the same time, time series model [5] attempted to incorporate the temporal information but the complex structures in music are difficult to capture because of the rich patterns of multiple time scales.

The other difficulty is the multitude of the labels. The large number of tags and relatively few tags per song result in severe label imbalance, presenting a challenging problem for most discriminative methods such SVM and AdaBoost [7, 13]. These methods tend to score high in classification by predicting most new test songs as negative samples. However, we found, with empirical evaluation, that logistic regression appears to be more robust in such situations in that it tries to maximize the conditional probability rather than to minimize the classification error directly.

Currently, most state-of-the-art methods are probabilistic models. Gaussian Mixture Models (GMM) [3] approximate the probability distribution of features conditioned on each tag with a mixture of Gaussian distributions. Then the Bayesian rule is applied to calculate the posterior probability of a tag given a new song. One shortcoming of the generative model is that it does not fully utilize the label information compared with discriminative methods. Recently, a more "discriminative-flavored" probabilistic model, Codeword Bernoulli Average (CBA) [1], was proposed and it achieved state-of-the-art performance on annotation and retrieval tasks. Although CBA is efficient and effective, the EM algorithm used in estimating its parameters only converges to a local optimum and as a result the learnt parameters will depend on different initializations.

We propose to use regularized logistic regression to ad-

dress the music tagging problem. First, song-level statistics are summarized in the "bag-of-words" of quantized delta MFCC features. Then, we apply logistic regression to learn the correlations of tags and music content by exploiting the label information. Different regularization terms are incorporated in logistic regression to reduce overfitting and improve generalization. Our approach enjoys the benefit of convex optimization with global optimum guarantee. Also, by using first-order methods, the proposed model can be learnt in a short time and it scales linearly to large dataset. Moreover, experiments demonstrate that our regularized logistic regression can achieve state-of-the-art performance in CAL-500 dataset [2].

## 2. SONG-LEVEL FEATURE REPRESENTATION

We choose a simple "bag-of-words" representation, the same as in [1, 11] and many other image classification algorithms [10], as our song-level feature. This simple representation facilitates efficient and scalable prediction of music tags for a large set of data.

Our primary features are the 39 dimension delta MFCC features over 23ms time-window. Each delta MFCC feature is concatenated from one MFCC feature, its first derivative and its second derivative. As a preprocessing step, we first normalize all the delta MFCC features to have zero mean and unit variance in each dimension. We then apply k-means to learn $K$ cluster centroids as "audio dictionary" $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K] \in \mathbb{R}^{p\times K}$ in the $p$ dimensional feature space, where $p = 39$. The centroids act as "representatives" of typical audio frames.

Let $\{\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \cdots, \mathbf{v}_{i,N_i}\}$ denote the set of delta MFCC vectors for song $i$. We count the number of feature vectors for song $i$ that are nearest to dictionary item $\mathbf{d}_j$ in Euclidean distance

$$n_{i,j} = \left| \left\{ k : j = \arg\min_t \|\mathbf{v}_{i,k} - \mathbf{d}_t\|_2^2 \right\} \right|. \quad (1)$$

The counts $n_{i,j}$ can be considered as a discrete approximation to the probability distribution on the feature space. Compared with the parametric model [3], our non-parametric representation is more flexible and easier to implement.

We then normalize the counts to cancel out the effect of varying song lengths. The frequency of the $j$-th "audio word" in the $i$-th song is calculated as

$$r_{i,j} = \frac{n_{i,j}}{\sum_{k=1}^{K} n_{i,k}}. \quad (2)$$

Finally, the $i$-th song is represented as $\mathbf{x}^{(i)}$ whose $j$-th element is $x_j^{(i)} = r_{i,j}$.

The most time consuming part of song-level feature representation is k-means clustering. However, this is done offline and can be speeded up by using a subset of samples or using hierarchical clustering. When a new song arrives, we just need to assign each of its delta MFCC features to one of the centroids and construct the histogram, whose time complexity is linear in the number of delta MFCC features.

## 3. LOGISTIC REGRESSION WITH REGULARIZATION

Given the "bag-of-words" representation of each song, we train a linear classifier to predict the labels. We choose logistic regression because its loss function is less sensitive to noise and label imbalance compared with others, such as hinge loss in SVM or exponential loss in AdaBoost.

### 3.1 Multi-label Logistic Regression

In the automatic music tagging problem, there are $m$ labels/tags, and we want to learn a vector-valued prediction function $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})]^T : \mathcal{X} \mapsto \mathcal{Y}$, where the input space $\mathcal{X}$ is the $K$ dimensional vector space of "bag-of-words" and the label space $\mathcal{Y}$ is $\{1, -1\}^m$. Here, we are interested in the family of linear classifiers and $\mathbf{f}(\mathbf{x})$ can be written as

$$\mathbf{f}(\mathbf{x}) = \mathrm{sgn}(\mathbf{B}\mathbf{x} + \mathbf{c}), \quad (3)$$

where $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m]^T \in \mathbb{R}^{m\times K}$ is the coefficient matrix for the prediction function and $\mathbf{c} \in \mathbb{R}^m$ is the bias vector. Note that row $l$, $\mathbf{b}_l^T$, is the classifier coefficients for the $l$-th label.

With logistic regression model, the conditional likelihood $\Pr(y_l|\mathbf{x}; \mathbf{b}_l, c_l)$ is give by

$$\Pr(y_l|\mathbf{x}; \mathbf{B}, \mathbf{c}) = \frac{1}{1 + \exp\left(-y_l\left(\mathbf{b}_l^T\mathbf{x} + c_l\right)\right)}. \quad (4)$$

And the learning of optimal parameters $(\mathbf{B}^*, \mathbf{c}^*)$ based on a training dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}|i = 1, 2, \cdots, n)\}$ can be performed by minimizing the negative log likelihood plus a regularization term $\mathcal{R}(\mathbf{B})$,

$$(\mathbf{B}^*, \mathbf{c}^*) = \arg\min_{\mathbf{B},\mathbf{c}} -\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{m} \log\Pr\left(y_l^{(i)}|\mathbf{x}^{(i)}; \mathbf{B}, \mathbf{c}\right) + \lambda\mathcal{R}(\mathbf{B}), \quad (5)$$

where $\lambda$ is a weighting parameter for the regularization.

To predict the labels of a new song $\widehat{\mathbf{x}}$, we compute the conditional likelihood $\Pr(y_l|\widehat{\mathbf{x}}; \mathbf{B}^*, \mathbf{c}^*)$ with Eq. 4, which shows the confidence of the label $y_l$.

### 3.2 Different Regularizations

Regularization plays an important role in incorporating prior information and reducing model complexity to avoid overfitting. Adopting different regularization terms will lead to models with different interpretations and performance.

A common choice is the $l_2$ term that contains model complexity, i.e.

$$\mathcal{R}(\mathbf{B}) = \|\mathbf{B}\|_2^2 = \sum_{j=1}^{K}\sum_{i=1}^{m} B_{ij}^2. \quad (6)$$

Recently, sparsity inducing norms are very popular and have wide applications in machine learning and music information retrieval [8, 14]. So, we also consider $l_1$ norm regularization that encourages sparsity of model parameters. Technically, the regularizer is

$$\mathcal{R}(\mathbf{B}) = \|\mathbf{B}\|_1 = \sum_{j=1}^{K}\sum_{i=1}^{m} |B_{ij}|. \quad (7)$$

## 4. FIRST-ORDER OPTIMIZATION METHOD

We adopt efficient first-order methods to learn the parameters. Thanks to convexity, the convergence of our algorithm to a global minimum is guaranteed.

### 4.1 Gradient Descent for $l_2$

Since the original objective function with $l_2$ regularization is smooth, we can update the parameter by gradient descent

$$\mathbf{B}_{t+1} = \mathbf{B}_t - \eta\left(\nabla\mathcal{L}_n(\mathbf{B}_t) + 2\lambda\mathbf{B}_t\right), \quad (8)$$

$$\mathbf{c}_{t+1} = \mathbf{c}_t - \eta\nabla\mathcal{L}_n(\mathbf{c}_t), \quad (9)$$

where $\nabla\mathcal{L}_n(\cdot)$ is the derivative of the loss function and $\eta$ is the step size.

### 4.2 Generalized Gradient Descent for $l_1$

Due to the non-smoothness of $l_1$ norm, at iteration step $t$, we update $\mathbf{B}$ by

$$\mathbf{B}_{t+1} = \arg\min_{\mathbf{Z}} \langle\nabla\mathcal{L}_n(\mathbf{B}_t), \mathbf{Z} - \mathbf{B}_t\rangle + \frac{1}{2\eta}\|\mathbf{Z} - \mathbf{B}_t\|^2 + \lambda\|\mathbf{Z}\|_1, \quad (10)$$

where $\eta > 0$ and $1/\eta$ is set larger than the Lipschitz constant of $\nabla\mathcal{L}_n$ [9]. Here we omit $\mathbf{c}$ because it is not in the $l_1$ norm and can be solved by standard gradient descent (Eq. 9).

The above procedure is the generalized gradient descent scheme because when $\lambda = 0$, it is easy to see Eq. 10 reduces to $\mathbf{B}_{t+1} = \mathbf{B}_t - \eta\nabla\mathcal{L}_n(\mathbf{B}_t)$.

Denote $\mathbf{B}_{t+1} = [\mathbf{b}_1^*, \mathbf{b}_2^*, \cdots, \mathbf{b}_p^*]$, $\mathbf{B}_t = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_p]$ and $\nabla\mathcal{L}_n(\mathbf{B}_t) = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_p]$ and Eq. 10 can be solved by $p$ separate sub-problems. According to [9], each sub-problem is solved by

$$\mathbf{b}_j^* = \mathcal{T}_{\lambda\eta}\left(\mathbf{b}_j - \eta\mathbf{h}_j\right), \quad (11)$$

where $\mathcal{T}_\alpha(\cdot)$ is the soft thresholding operator. And it is defined by

$$\mathcal{T}_\alpha(\mathbf{x})_i = (|x_i| - \alpha)_+ \operatorname{sgn}(x_i), \quad (12)$$

where $(x)_+ = x$ if $x > 0$ and $(x)_+ = 0$ otherwise.

The detailed procedure of generalized gradient descent is illustrated in Alg. 1.

**Algorithm 1** Generalized Gradient Descent Algorithm

**Input**: Training set $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}|i = 1, 2, \cdots, n)\}$
**Output**: Model parameters $\mathbf{B}^* \in \mathbb{R}^{m\times p}$, $\mathbf{c}^* \in \mathbb{R}^m$
**Initialize** $t = 0$,$\eta$,$\mathbf{B}_0$,$\mathbf{c}_0$
Repeat until convergence:

1. Compute the partial gradient $\nabla_{\mathbf{B}}\mathcal{L}_n(\mathbf{B}_t, \mathbf{c}_t)$.
2. For $j = 1$ to $p$
   2.1 Calculate $\mathbf{w} = \mathbf{b}_j - \eta\mathbf{h}_j$.
   2.2 Calculate the $j$-th column of $\mathbf{B}_{t+1}$ by $\mathcal{T}_{\lambda\eta}(\mathbf{w})$.
3. Compute the partial gradient $\nabla_{\mathbf{c}}\mathcal{L}_n(\mathbf{B}_t, \mathbf{c}_t)$.
4. Update $\mathbf{c}_{t+1} = \mathbf{c}_t - \eta\nabla_{\mathbf{c}}\mathcal{L}_n(\mathbf{B}_t, \mathbf{c}_t)$.

## 5. EXPERIMENTS ON ANNOTATION AND RETRIEVAL

We evaluated our three versions of logistic regression on two tasks: music annotation and retrieval. Compared with binary classification tasks, these two tasks are more closely related with real scenarios.

The music data comes from CAL-500 Dataset [2]. There are 500 Western polyphonic songs and the annotations were collected from more than three human subjects per song. When training the classifier, we only use the binary annotations with $\{0, 1\}$ (transformed to $\{-1, 1\}$ for learning) to indicate whether the tag is relevant to the song.

We are more interested in predicting more "useful" tags rather than very obscure ones. Following the same setting in [4, 5], we only evaluate on the 78 tags that have at least 50 examples and 97 top popular tags.

### 5.1 Annotation and Retrieval

Using similar experimental setting as in [4, 5], we used 5-fold cross validation. In each round, we first learned our model parameters $\mathbf{B}^*, \mathbf{c}^*$ with the 400-song training set and predicted confidence ratings on the remaining 100-song test set. The conditional probability (confidence rating) of a tag being assigned to a song was then calculated using Eq. 4. To compensate for non-uniform label prior, we adopted the same heuristic used in [1] by introducing a "diversity factor"

| Model | Precision | Recall | F-score | P3 | P5 | P10 | MAP | AROC |
|---|---|---|---|---|---|---|---|---|
| CBA | 0.361 | 0.212 | 0.267 | 0.463 | 0.458 | 0.440 | 0.425 | 0.691 |
| GMM | 0.405 | 0.202 | 0.269 | 0.456 | 0.455 | 0.441 | 0.433 | 0.698 |
| Context-SVM | 0.380 | 0.230 | 0.286 | 0.512 | 0.487 | 0.449 | 0.434 | 0.687 |
| DirMix | 0.441 | **0.232** | **0.303** | **0.519** | 0.501 | 0.470 | 0.443 | 0.697 |
| LogRegr | 0.396 | 0.196 | 0.262 | 0.407 | 0.428 | 0.424 | 0.404 | 0.671 |
| $l_1$ LogRegr | 0.416 | 0.202 | 0.272 | 0.414 | 0.413 | 0.417 | 0.411 | 0.673 |
| $l_2$ LogRegr | **0.446** | 0.227 | 0.301 | 0.515 | **0.512** | **0.485** | **0.459** | **0.719** |

**Table 1**. Experimental results for top 97 popular tags. The results of Codeword Bernoulli Average (CBA), Gaussian Mixture Models (GMM), Context-SVM and Dirichlet Mixture (DirMix) are reported in [4]. Our results are non-regularized (LogRegr), $l_1$ regularized ($l_1$ LogRegr) and $l_2$ regularized ($l_2$ LogRegr) logistic regressions, respectively.

| Model | P | R | F-score | AROC | MAP | P10 |
|---|---|---|---|---|---|---|
| CBA | 0.41 | 0.24 | 0.29 | 0.69 | 0.47 | 0.49 |
| HEM-GMM | **0.49** | 0.23 | 0.26 | 0.66 | 0.45 | 0.47 |
| HEM-DTM | 0.47 | 0.25 | 0.30 | 0.69 | 0.48 | 0.53 |
| LogRegr | 0.44 | 0.23 | 0.30 | 0.67 | 0.45 | 0.48 |
| $l_1$ LogRegr | 0.46 | 0.23 | 0.31 | 0.68 | 0.46 | 0.49 |
| $l_2$ LogRegr | 0.48 | **0.26** | **0.34** | **0.72** | **0.50** | **0.54** |

**Table 2**. Experimental results for top 78 popular tags. The results of Codeword Bernoulli Average (CBA), hierarchical EM Gaussian Mixture Models (HEM-GMM) and hierarchical EM Dynamic Texture Model (HEM-DTM) are reported in [5]. Our results are non-regularized (LogRegr), $l_1$ regularized ($l_1$ LogRegr) and $l_2$ regularized ($l_2$ LogRegr) logistic regressions, respectively.

$d = 1.25$. For each predicted confidence rating, we subtracted $d$ times the mean confidence for that tag. We then assigned each song with the top 10 most confident tags.

Annotation was evaluated by mean precision and recall over the tags. Given the 10 annotations per song in the test set, we calculated precision and recall for each tag and then averaged across all considered tags. The final result was averaged over 5 rounds of cross validation. In addition, F-score, the harmonic mean of precision and recall, was computed to summarize the two aspects of precision and recall.

For retrieval, we first ranked the songs in the descending order according to confidence ratings for a specific tag. Better retrieval result corresponds to cases that more relevant songs appear at the top of the ranking list. Then, we calculated precision at every position down the ranking list via dividing the number of true positives found so far by the total number of songs so far. Evaluation was conducted through averaged precision and *precision at* $k$ ($k = 3, 5, 10$) as in [4]. Averaged precision was computed by taking the average of all the positions down the ranking list where new true positives were found. Precision at $k$ was $k$-th precision that we calculated on the ranking list.

### 5.2 Experiment Results and Dicussions

#### *5.2.1 Comparison with State-of-the-art*

We compare our results with state-of-the-art performance on the CAL-500 dataset. For the 97 tags setting, we compare with CBA [1], GMM [3], Context-SVM [12] and Dirichlet Mixtures (DirMix) [4]. Their results were originally reported in [4] and are copied in Table 1 for more convenient comparison. For the 78 tags setting, CBA, HEM-GMM (the same as GMM) and HEM-DTM [5] were compared. Their original results reported in [5] and copied in Table 2.

The results of our three variants of logistic regression under the 97 tags setting are also reported in Table 1. All our methods were based on $K = 2000$ dictionary size "bag-of-words" representation, with the cluster centroids trained on a random subset of 100,000 samples from all the delta MFCC features provided in the dataset. Non-regularized logistic regression was equivalent to setting $\lambda = 0$. The parameter $\lambda$ in the two regularized algorithms were set to the optimum. For $l_1$ logistic regression, it was set to 0.001 and for $l_2$ logistic regression, it was set to 0.01.

From Table 1, we can see that non-regularized logistic regression performed the worst but still had reasonable results. $l_1$ regularization improved the performance by 0.01 or 0.02 for some measures. $l_2$ regularization introduced greater improvement over the $l_1$ regularized variant, achieving best performance in retrieval even than the state-of-the-art. And it was comparable with the Dirichlet Mixture model in annotation task. Note that the Dirichlet Mixture model exploited the label correlations explicitly while our method incorporated no such schemes to utilize context information.

For the 78 tags case illustrated in Table 2, the simple logistic regression performed better than CBA in the annotation task. $l_1$ regularization consistently improved the performance by 0.01 for most measures. Again, $l_2$ regularized logistic regression outperformed other approaches in

all measures except for precision. However, by comparing the F-score which summarizes the overall annotation score, all three variants performed better than or on par with the state-of-the-art. Considering the fact that HEM-DTM benefited from information over the 23ms time window, our algorithms' performance are even more encouraging.

The performance of non-regularized logistic regression was limited because of the overfitting effect. $l_1$ regularization slightly improved the situation by constraining the complexity of the parameters. However, it appears that the "bag-of-words" representation does not have the hidden sparse structure which $l_1$ norm regularization can help reveal. Rather, the classifier coefficients should be dense to fully take into account all the details in the distribution. The $l_2$ norm was thus suitable for such situation where it constrained the model complexity in general and produced non-zero coefficients.

### 5.2.2 *Effect of Changing Dictionary Size $K$*

We also explored the effect of different dictionary sizes $K$. In the experiments, we ran $l_2$ regularized logistic regression with $\lambda$ set to 0.01 and under different $K$ (10, 20, 50, 100, 200, 500, 800, 1000, 2000 and 5000). Fig. 1 illustrates the performance on the two tag number settings for annotation and retrieval tasks.

From Fig. 1, we can see that as $K$ increases, the algorithm benefits from more accurate approximation to the distribution and achieves better performance. The biggest improvement occurs from 10 to 100 dictionary sizes. It appears that when $K$ increases over this range, the major structure in the distribution has been captured by the "bag-of-words" representation. As we go on to model the finer scales with even larger $K$, the performance continues to climb up until it gradually levels off when $K$ exceeds 2000. From $K = 2000$ to $K = 5000$, the improvement is less than 0.01 for retrieval while the computational cost is multiplied by 2.5 times. Therefore, we choose $K = 2000$ as our optimal dictionary size in the CAL-500 dataset.

### 5.2.3 *Effect of Different Regularization Parameter $\lambda$*

The regularization parameter $\lambda$ affects the performance by balancing the loss function and the regularization. Smaller $\lambda$ leads to more focus on the empirical error while larger $\lambda$ places more priority on keeping the model complexity low.

We varied $\lambda$ from $10^{-5}$ to 10 with equal stepsize in logarithm scale for $l_2$ regularized logistic regression under $K = 2000$. The effect is demonstrated in Fig. 2. For $l_2$ regularized logistic regression, the optimal $\lambda$ is 0.01. And we can see that the algorithm is relatively robust to the parameter change from $10^{-4}$ to $10^{-1}$. Note that since the values in the original normalized "bag-of-words" representation are too small, making them badly scaled compared with the bias, we multiply the "bag-of-words" by 100 and the parameter $\lambda$ is reported after such preprocessing.

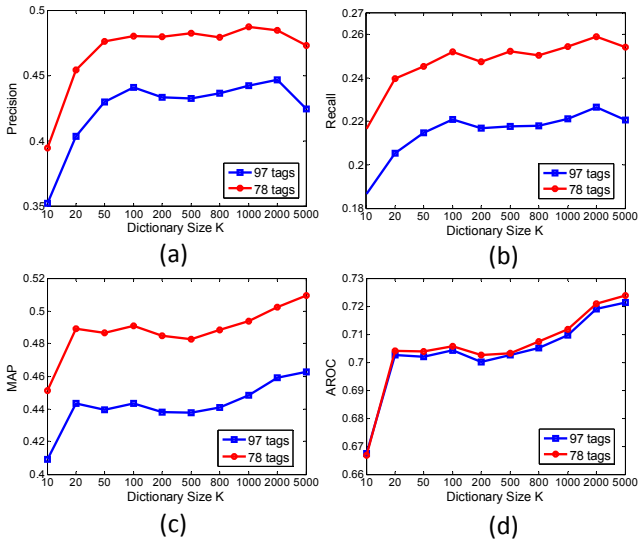


**Figure 1**. Effect of varying dictionary size $K$. The performance is evaluated on $l_2$ LogRegr with optimal parameter setting. (a) Annotation performance: precision; (b) Annotation performance: recall; (c) Retrieval performance: mean averaged precision; (d) Retrieval performance: area under the receiver operating characteristic curve.

## 6. CONCLUSIONS

We proposed to use regularized logistic regression algorithms to automatically tag music. Our approach enjoys convex formulations and can be solved efficiently by first-order methods. The convergence of our algorithm is guaranteed and it is scalable to large dataset. Empirical evaluation for music annotation and retrieval on the CAL-500 dataset has shown that $l_2$ regularized version with "bag-of-words" representation of quantized delta MFCC features achieves state-of-the-art performance.

Currently, no label correlations are considered in our framework and learning is done independently for each label. In future work, we are interested in modeling such correlations by using structure inducing norms for regularization. Also, instead of k-means clustering, dictionary learning approaches are promising in that more adaptive "audio words" can be learnt from data.

## 7. REFERENCES

[1] M. Hoffman, D. Blei and P. Cook: "Easy as CBA: A Simple Probabilistic Model for Tagging Music," *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, 2009.

[2] D. Turnbull, L. Barrington, D. Torres and G. Lanckriet: "Towards Musical Query-by-Semantic Description us-

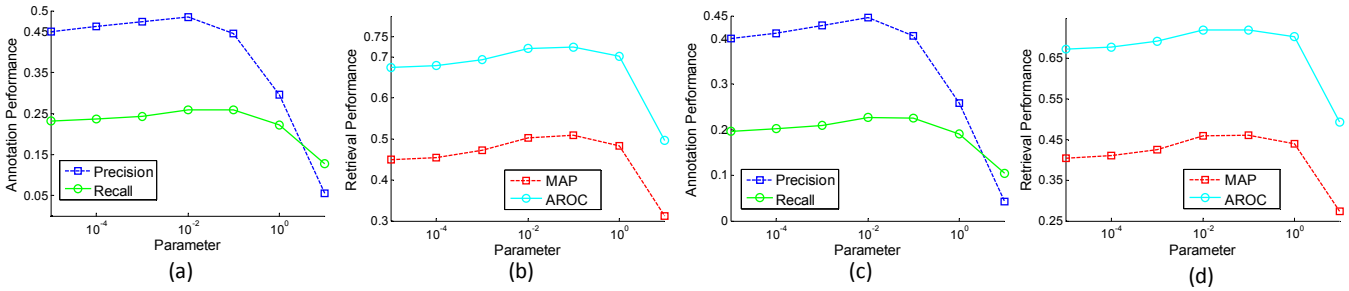


**Figure 2**. Effect of varying regularizaiton parameter $\lambda$. The performance is evaluated on $l_2$ LogRegr with $K = 2000$. (a) Annnotation performance for 78 tags setting; (b) Retrieval performance for 78 tags setting; (c) Annnotation performance for 97 tags setting; (d) Retrieval performance for 97 tags setting.

ing the CAL500 Data Set," *ACM Special Interest Group on Information Retrieval Conference (SIGIR '07)*, 2007.

[3] D. Turnbull, L. Barrington, D. Torres and G. Lanckriet: "Semantic Annotation and Retrieval of Music and Sound Effects," *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

[4] R. Miotto, L. Barrington and G. Lanckriet: "Improving Auto-Tagging by Modeling Semantic Co-Occurrences," *Proceedings of the 11th International Conference on Music Information Retrieval*, 2010.

[5] E. Coviello, A. Chan, L. Barrington and G. Lanckriet: "Automatic Music Tagging With Time Series Models," *Proceedings of the 11th International Conference on Music Information Retrieval*, 2010.

[6] J. Bergstra, M. Mandel and D. Eck: "Scalable genre and tag prediction with spectral covariance," *Proceedings of the 11th International Conference on Music Information Retrieval*, 2010.

[7] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl: "Aggregate features and AdaBoost for music classification," *Machine Learning*, 2006.

[8] Y. Panagakis, C. Kotropoulos and G. Arce.: "Music genre classification using locality preserving non-negative tensor factorization and sparse representations," *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, 2009.

[9] A. Beck and M. Teboulle: "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, 2009.

[10] G. Csurka, C. Dance, L.X. Fan, J. Willamowski and C. Bray: "Visual categorization with bags of keypoints," *Proc. of ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[11] M. Hoffman, D. Blei, and P. Cook: "Content-based musical similarity computation using the hierarchical Dirichlet process," *In Proc. International Conference on Music Information Retrieval*, 2008.

[12] S.R. Ness, A. Theocharis, G. Tzanetakis and L.G. Martins: "Improving automatic music tag annotation using stacked generalization of probabilistic SVM outputs," *In Proceedings of ACM Multimedia*, 2009.

[13] D. Turnbull and C. Elkan: "Fast recognition of musical genres using RBF networks," *IEEE Transactions on Knowledge and Data Engineering*, 2005.

[14] K. Koh, S.J. Kim and S. Boyd: "An Interior-Point Method for Large-Scale $l_1$-Regularized Logistic Regression," *Journal of Machine Learning Research*, 2007.