Music Retagging Using Label Propagation and Robust Principal Component Analysis

Yi-Hsuan Yang*
Academia Sinica
Taipei, Taiwan
yang@citi.sinica.edu.tw

Dmitry Bogdanov, Perfecto Herrera,
Mohamed Sordo
Universitat Pompeu Fabra
Barcelona, Spain
{dmitry.bogdanov, perfecto.herrera,
mohamed.sordo}@upf.edu

ABSTRACT

The emergence of social tagging websites such as Last.fm has provided new opportunities for learning computational models that automatically tag music. Researchers typically obtain music tags from the Internet and use them to construct machine learning models. Nevertheless, such tags are usually noisy and sparse. In this paper, we present a preliminary study that aims at refining (retagging) social tags by exploiting the content similarity between tracks and the semantic redundancy of the track-tag matrix. The evaluated algorithms include a graph-based label propagation method that is often used in semi-supervised learning and a robust principal component analysis (PCA) algorithm that has led to state-of-the-art results in matrix completion. The results indicate that robust PCA with content similarity constraint is particularly effective; it improves the robustness of tagging against three types of synthetic errors and boosts the recall rate of music auto-tagging by 7% in a real-world setting.

Categories and Subject Descriptors

H.5.5 [**Sound and Music Computing**]: Methodologies and techniques, Systems

General Terms

Algorithms, Performance

1. INTRODUCTION

Development of effective technologies to deal with large volume of multimedia objects is a fundamental applied-research target for our current digital society. Music, as one of the involved multimedia modalities, has attracted specific research that, during the last decade, has made it possible to extract musical information from audio files or from text documents dealing with musical issues [8]. Applications such as music recommendation, playlist generation or finding musical mates can be addressed thanks to different combinations of state-of-the art techniques of music audio analysis, text analysis, network analysis and knowledge representation. In this context, a frequent distinction can generally be made between content-based music analysis and retrieval and text-based music analysis and retrieval. Features for the former (e.g., spectrum, timbre, rhythm, pitch, tonality) are extracted from audio information with different levels of reliability and effectiveness. We nevertheless are still far from a complete or even acceptable representation of the musical features that humans are able to compute and use in order to perceive, enjoy or describe a musical excerpt. Several limitations restrict its practicability. Firstly, its precision is usually unsatisfactory because of the semantic gap between computable audio music features and high-level semantic concepts. Secondly, a proper music knowledge representation requires a multi-feature, multidimensional and multi-resolution approach (and, currently, this is hardly feasible or authors prefer using simpler paths). Thirdly, content that is present in the audio does not exhaust all the information associated with it, as there are contextual and cultural data that have to be sought outside the audio (e.g., in texts, images or in listener behaviors). Content-based music analysis and retrieval makes possible to search by example (using audio-based similarity) or to retrieve cover versions of a song, for example. In contrast, tag-based music information retrieval solely adopts text information to carry through the audio music indexing and search [20]. Compared with audio information, text is essentially a simplified representation of music and audio contents that benefits of a certain low dimensionality (standard listeners do not use more than some hundred tags to describe music) and of its semantic character (hence helping to bridge the semantic gap, up to a certain point). Combining content-based and text-based strategies for music retrieval has been shown to increase the efficacy of music information retrieval (MIR) systems and the satisfaction of their users, compared to narrow strategies dealing exclusively with one or another [15, 40].

Text information used for music retrieval can be acquired, for a given song, from different sources: existing metadata embedded in the files, surrounding text in web pages, lyrics, and user tagging. Especially with the prevalence of music recommendation and music sharing communities such as Last.fm,[1] Soundcloud,[2] or Freesound[3] which host vast collections of music and audio files with user-provided tags, tag-based music retrieval has become potentially popular and practical in different scenarios aimed to different lis-

[*] This work was conducted while Yi-Hsuan Yang visited the Music Technology Group of the Universitat Pompeu Fabra.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

[1] http://www.last.fm/

[2] http://soundcloud.com/

[3] http://www.freesound.org/

tener's needs [20]. Contrasting to the other text sources, tags are created and assigned "socially" (i.e., in a process of *multi-way negotiation* of meaning between each individual and the community he/she is belonging to). This evolvable social meaning negotiation is probably one of the paths to revealing aspects of the collective mind that emerges when remote people share musical concepts and content by means of information and communication technologies [13, 37].

The social nature of this sharing process creates a new situation with new challenges [16]. For example, the lack of formal editorial processes often results in poor quality tags. We can observe that sometimes tags describe a specific musical aspect happening in a specific temporal moment (e.g., "guitar solo") [23], while in other cases tags are very imprecise (e.g., "seen live") and they may not even refer to musical content but to highly personal and subjective experiences with that musical excerpt. Tags also tend to have uneven densities (i.e., some content items may get many tags and a large support for each tag while many items will get very few; some tags will be very popular while many of them will not). Additionally, many music files are incompletely tagged and, from the existing community-based tag vocabulary, only a few of the potentially correct tags have been used for a specific file (this is sometimes referred to as the *weak labeling* problem [38]). Practical constraints (the huge files-to-taggers ratio) impede an accurate and complete textual description of music content by human listeners/taggers. Systems that are able to capitalize on existing robust knowledge about musical content, textual musical information, and social computing [9, 32, 43], in order to refine tags (i.e., add potentially correct but currently unassigned tags and wipe wrongly assigned tags) are highly desirable for developing powerful music retrieval systems and applications. Because of that, music tag propagation has become a typical problem in music information research. Here, though, we address what could be considered as a pre-requisite for the successful propagation of tags: maximizing the quality of available tags. This has been called "retagging" in the domain of image processing [22, 50] and, up to our current knowledge, has not been addressed for music retrieval yet.

Formally speaking, while *music tagging* is often described as a process that manually or automatically assigns tags to music objects (could be artists, albums, tracks, or segments of a track) from a vocabulary of music tags, *music retagging* is an automatic process that refines the raw, original tagging of music by exploiting the intrinsic structure of the music content space and the induced music tag space to modify the assignment and structure of tags such that a music object can be better identified, catalogued, and retrieved in the refined tag space. Retagging modifications comprise enriching, denoising or assessing confidence of content-tags association measures, in addition to merging or removing tags. Fig. 1 shows a schematic diagram of music retagging.

The goal of this paper is to set up a framework that employs and quantitatively evaluates music retagging. Specifically, we investigate the use of label propagation [48] to exploit the content similarity of music and the use of robust principal component analysis [21] to find a low-dimensional structure of the music tag space. These two algorithms can be readily applied to a raw music tagging without extra request of supervision or affiliated metadata. Our experiment on refining the human tagging of music tracks shows that retagging indeed improves the quality of tag assignment and facilitates applications such as tag-based music retrieval.

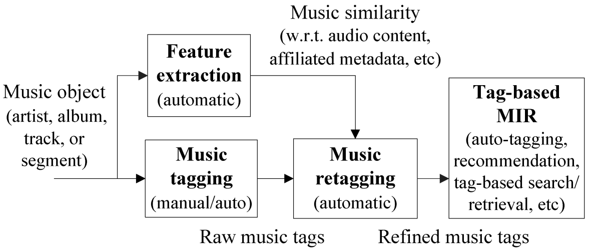


Figure 1: A schematic diagram of music retagging.

The rest of the paper is organized as follows. Section 2 reviews the related work on music tagging and image tag refinement. In Section 3, we introduce the music retagging framework and discuss the properties of music that can be taken into account. Section 4 describes the label propagation and robust principal analysis algorithms that can be employed for music retagging. Experimental results on two music auto-tagging datasets are reported and analyzed in Section 5. Section 6 concludes the paper.

2. RELATED WORK

Our work is motivated by the research on image tag refinement, which is receiving increasing attention in the computer vision community. Existing approaches usually consider both the semantic correlation between tags and the visual similarity between images and use graph-based models such as random walk with restart [42] or multi-graph reinforcement [14] to refine tags. Some approaches estimated tag correlation directly from the original tagging matrix (e.g., tag co-occurrence) [22], while others proposed the use of external resources such as WordNet, Wikipedia, or Google since the tag assignment in the original tagging matrix can be imprecise and incomplete [11, 35, 50]. Robust principal component analysis has also been utilized for image tag refinement by Zhu *et al.* [50], where an optimization framework based on accelerated proximal gradient is proposed. We also utilize this framework in our study (cf. Section 4.3).

To our best knowledge, few attempts if any have been made to address music retagging. Music retagging is different from image tag refinement at least in the following aspects: First, music has a temporal dimension and some tags assigned to a song may be only associated with a short segment of the song (e.g., "saxphone solo") [23]. Second, it is more difficult to measure the semantic correlation between music tags because many of them are fairly *specific* (e.g., "a dominant bass riff," "a jazz waltz feel," "interweaving vocal harmony," or "lyric-centric composition" [38]) or *subjective* (e.g., "happy," "sad," "weird," or "going to sleep" [16, 41, 47]). Third, music is a complex phenomenon and there can be multiple facets, such as genre, instrument, mood, and acoustic quality, underlying a folksonomy of music tags [35].

The automatic classification of music audio items in terms of high level concepts such as genres and moods is sometimes termed "auto-tagging." Classification is carried out by learning models automatically from the mapping between (low level) audio features and semantic labels, or tags. A number of approaches have been proposed in the literature [3]. Most of them rely on the bag-of-frames approach [1, 44] whereby audio features, after being computed on a short-time basis, are aggregated into a "bag" using their first statistical moments. Finally, an optimized small subset of features is used to train a classifier, using a database of labeled/tagged audio excerpts. Tag databases can be obtained from different sources [39]: conducting human surveys (e.g., CAL500), deploying games with a purpose (e.g., MajorMiner[4], Magnatagatune[5]), collecting web documents or harvesting social tags (e.g., Million Song Dataset[6]). In the last few years, there has been a growing interest in the use of two–stage algorithms [24]. Typically, a two–stage algorithm uses the output of a content–based algorithm as input feature vectors to model tag co-occurrences in the vocabulary [2]. A number of authors report on the performance improvements using this method [26, 28].

3. MUSIC RETAGGING

Given the raw music tagging matrix Y, either from human tagging or music auto-tagging, the objective of music retagging is to generate a matrix A with better quality in tag assignment and thereby the semantic description of music objects. The raw tagging matrix consists of the tag assignment $Y \in \{0,1\}^{n\times m}$ of m tags $\{w_j\}_{j=1}^{m}$ from a vocabulary of music tags to n music objects $\{d_i\}_{i=1}^{n}$, such that $Y_{ij} = 1$ if the tag w_j is associated with the music object d_i. The retagged matrix A is typically of the same dimensionality of Y, but one can employ some algorithms to merge relevant tags or remove unimportant tags to reduce the size of the tag vocabulary. In this work we do not attempt to modify the tag vocabulary and define $A \in \mathbb{R}^{n\times m}$ such that A_{ij} denotes the confidence score of the association between w_j and d_i. The retagged matrix A can be converted to a binary one if necessary by for instance selecting the top K tags with highest confidence scores for each music object.

We exploit the following properties for retagging,

- **Content similarity**. Music objects similar in content are typically associated with similar tags. This property is frequently exploited in current music auto-tagging systems that seek a correspondence between music content space and music tag space. Once the correspondence is learnt, one can propagate tags to a previously un-annotated song by referring to its similar songs. Content similarity can be measured with respect to audio signal content such as timbre, rhythm, melody, and harmony, or with respect to affiliated metadata such as artist, album, genre, locale, release-year, or popularity, just to name a few.

- **Low-rank**. Due to the semantic redundancy of the descriptors (tags) we use to annotate music, it is possible to approximate the music tagging space by a smaller subset of "latent" tags derived from the original space. Such low-rank approximation reduces the *complexity* of the tagging space and therefore reduces the quantity of outliers and unnecessary tags.

- **Error sparsity**. The discrepancy between the raw tagging matrix and the refined one is hypothetically sparse in support (with most entries being zero) because the original tagging should be accurate to a certain degree and because a music object is unlikely associated with a large number of tags. That is to say, what we want to remove or add to the tagging matrix should be small in quantity.

Given the raw tagging matrix, it suffices to perform feature extraction and construct the n by n similarity matrix to exploit the above three properties for retagging, as the latter two can be directly inferred from the tagging matrix. While music tagging can be done either manually or automatically, the processes of feature extraction and music retagging are totally automatic, as depicted in Fig. 1.

4. ALGORITHMS

Three algorithms [21, 48, 50] are employed and quantitatively compared under the framework of retagging in this study.

4.1 Label Propagation

Based on the assumption that songs similar in content often share similar semantic meaning, one straightforward yet effective approach to music autotagging is to compute the pairwise similarity between songs based on music content features and approximate the tagging of a song by its neighboring ones. For example, the authors in [34] used a weighted vote from the k-nearest neighbors (most similar) of a song to determine the tagging of that song.

Many label propagation algorithms have been proposed in the literature [7]. In this work we adopt the graph-based label propagation algorithm proposed in [48] for its simplicity and well-known effectiveness. This algorithm can be considered as an iterative process where adjacent nodes (songs) exchange information (tagging) in proportion to the weight (similarity) of the interlinking edge. During each iteration each node receives the information from its neighbors and also retains some of its initial information. When converged the tagging of adjacent songs would be smooth with respect to the intrinsic structure collectively revealed by well-tagged and poorly-tagged songs.

Given the original tagging matrix Y, the result of label propagation Y^* can be computed by [48],

$$Y^* = (1-\alpha)(I-\alpha S)^{-1}Y, \tag{1}$$

where $S = D^{-1/2}WD^{-1/2}$ is the Laplacian matrix computed from the affinity matrix W on the dataset, W_{ij} is a measure of the similarity between song i and j with W_{ii} being zero to avoid self-reinforcement, and D is a diagonal matrix with $D_{ii} = \sum_{j\neq i} W_{ij}$ [48]. $\alpha \in [0,1]$ is a parameter controlling the propagation rate, or the relative amount of the information from neighbors and one's initial information.

A great many approaches have been proposed for measuring music similarity W, e.g., [4, 25, 31]. In this work we utilize a novel similarity measurement working on high-level semantic descriptors (genres, musical culture, moods, instrumentation, rhythm and tempo) inferred by support vector machines from low-level timbral, temporal, and tonal audio features [4]. Specifically, classification results form a high-level semantic descriptor space, which contains the probability estimates for each class of each classifier, and each track can be represented as a point in this semantic space. We compute music similarity by means of weighted Pearson

[4] http://majorminer.org/

[5] http://tagatune.org/Magnatagatune.html

[6] http://labrosa.ee.columbia.edu/millionsong/

correlation between tracks. In our previous work we have found that with this measurement we are able to achieve state-of-art performance for a variety of music information retrieval tasks. We refer the interested reader to [4] for details on the employed distance measure.

4.2 Robust Principal Component Analysis

Another common general scheme in approximating a noisy target matrix Y is to select a matrix A that minimizes some combination of the *complexity* of A and the *discrepancy* $Y - A$. The most common notion of complexity of a matrix is its rank (as in classical principal component analysis or latent semantic analysis) [10, 36], or the maximum number of linearly independent column vectors of A. For example, it has been well-known that by omitting all but the r largest singular values of the singular value decomposition (SVD) of Y, one obtains a low-rank representation of Y with minimal entrywise discrepancy [12],

$$Y_r = U_r \Sigma_r V_r^T = \mathop{\arg\min}_{A,\ \mathrm{rank}(A) \leq r} ||Y - A||_F^2, \tag{2}$$

where $\Sigma_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ consists of the r-largest singular values of Y (the eigenvalues of YY^T), U is an $n \times r$ matrix whose columns are orthogonal, V is an $m \times r$ matrix which is also orthogonal, and $||E||_F = \sqrt{\mathrm{Tr}(EE^T)} = \sum_{ij}(E_{ij})^2$ is the Frobenius norm.

Nevertheless, in real-world problems such as matrix completion [5, 30], one often needs to recover a low-rank matrix from a corrupted one with *gross* (i.e., with arbitrary large magnitude) but *sparse* (i.e., most entries being zero) errors E. This is also the case of music retagging because music tags are by nature sparse and because human tagging is usually accurate to a certain extent only due to multi-way negotiation. In particular, it has been found that using trace norm as a surrogate to measure the complexity of A and $l1$-norm to measure the sparsity of E leads to the following convex optimization problem that can be solved efficiently,

$$\min_{A,E:\ Y=A+E} ||A||_* + \lambda ||E||_1, \tag{3}$$

where $|| \cdot ||_*$ represents the trace (nuclear) norm of a matrix (the sum of its singular values), $|| \cdot ||_1$ is the l_1 norm that denotes the sum of the absolute values of matrix entries, and λ is a positive weighting parameter. The above problem has been dubbed robust principal component analysis (RPCA) as its solution is robust to gross errors or outliers [6].

Efficient algorithms such as accelerated proximal gradient (APG) [21] and augmented Lagrange multipliers [49] have been proposed to compute A and E in an iterative fashion. In this work we adopt the APG algorithm as it can be easily extended to incorporate the content consistency constraint, which will be detailed below.

4.3 Robust Principal Component Analysis with Content Consistency Constraint

According to the theory of graph embedding [45, 51], the content consistency among the tracks can be enforced by solving the following optimization problem,

$$\min_A \sum_{ij} ||A_i - A_j||^2 W_{ij}, \tag{4}$$

which is equivalent to minimizing the trace of APA^T, where $P = D - W$. By adding this regularizer to the formulation of RPCA, one obtains the following optimization problem,

$$\min_{A,E:\ Y=A+E} g(A, E) = ||A||_* + \lambda_1 ||E||_1 + \lambda_2 \mathrm{Tr}(APA^T), \tag{5}$$

which reduces to RPCA when λ_2 is set to 0. Instead of directly solving Eq. 5, it is computational expedient to relax the equality constraint $Y = A + E$ and solve,

$$\min_{A,E} \ \mu g(A, E) + \frac{1}{2} ||A + E - Y||_F^2. \tag{6}$$

As the relaxation parameter μ approaches 0_+, any solution to Eq. 6 approaches the solution set of Eq. 5.

The APG algorithm proposed in [21] decomposes the above problem and optimizes for A and E in turn by taking advantage of the following propositions in trace norm minimization and $l1$ norm minimization [5],

$$\begin{aligned} U\mathcal{S}_\varepsilon(\Sigma)V^T &= \mathop{\arg\min}_X \ \varepsilon ||X||_* + \frac{1}{2}||X - G||_F^2, \\ \mathcal{S}_\varepsilon(G) &= \mathop{\arg\min}_X \ \varepsilon ||X||_1 + \frac{1}{2}||X - G||_F^2, \end{aligned} \tag{7}$$

where $U\Sigma V^T$ is the SVD of some matrix G, ε is a positive parameter, and $\mathcal{S}_\varepsilon(x) = \mathrm{sgn}(x)\max(|x| - \varepsilon, 0)$ is the soft-thresholding or the shrinkage operator [6]. The relationship between Eqs. 6 and 7 can be developed by using $G_t = Z_t - L^{-1}\nabla f(Z_t)$, where Z_t is a quadratic approximation of $X_t = [A_t^T, E_t^T]^T$, $f(X_t) = \mu\lambda_2 \mathrm{Tr}(A_t P A_t^T) + \frac{1}{2}||A_t + E_t - Y||_F^2$, and L is a constant with which $\nabla f(\cdot)$ is Lipschitz continuous, or $||\nabla f(X_1) - \nabla f(X_2)|| \leq L_f ||X_1 - X_2||$. In particular, the optimal A_t and E_t in each iteration t (with a monotonically decreasing μ_t) can be computed by,

$$\begin{aligned} A_{t+1} = U\mathcal{S}_{\frac{\mu_t}{L}}(\Sigma)V^T =&\ \mathrm{SVD}_{\frac{\mu_t}{L}}(Z_t - L^{-1}\nabla_A f(Z_t)), \\ E_{t+1} =&\ \mathcal{S}_{\frac{\mu_t \lambda_1}{L}}(Z_t - L^{-1}\nabla_E f(Z_t)), \end{aligned} \tag{8}$$

where $Z_t = X_t + \frac{b_{t-1}-1}{b_t}(X_t - X_{t-1})$, $b_t = \frac{\sqrt{4b_{t-1}^2+1}}{2}$, $L = \sqrt{4\mu\lambda_2\sigma_{\max}^2(P) + 6}$, and $\sigma_{\max}(\cdot)$ represents the maximum singular value of a matrix. The above algorithm has a convergence rate of $O(t^{-2})$ [29]. It can be further sped up by computing a partial SVD, instead of the full SVD, using packages such as `PROPACK` [17], due to the soft-thresholding of the singular values. Readers are referred to [6, 21, 50] for more theoretical and algorithmic details of APG.

Note that all the algorithms described above produce a refined tagging matrix A which is real-valued. Because we do not impose any constraint on the value of A, some entries of A may be negative when RPCA is used. As these negative entries tend to be sparse and of small magnitude, one may simply neglect them and consider A_{ij} as a confidence measure of tag association.

5. EXPERIMENTS

Instead of directly dealing with large-scale social tags, as a preliminary study we have decided to experiment on two popular auto-tagging datasets, CAL500 [41] and CAL10k [38][7] for rapid evaluation. For CAL500 we would consider synthetic noises and evaluate in terms of the accuracy of *tag-based music retrieval*; for CAL10k we would apply retagging on the original tagging matrix and evaluate in terms of the accuracy of *auto-tagging*.

[7] `http://cosmal.ucsd.edu/cal/projects/AnnRet/`

Note that the presented evaluation is only considered as preliminary. The next step is to extend the corpus to large-scale social tags, whose characteristics may be different from either CAL500 or CAL10k. To evaluate retagging on social tags one needs *assured* ground truth that is validated by human annotators [50], which is particularly difficult to obtain for music [19]. This issue would be addressed in our future work by evaluating how the refined social tags lead to better tag-based MIR applications [20] such as style classification or music recommendation.

5.1 Evaluation on CAL500

We first evaluate the robustness of retagging against three different types of synthetic noises on CAL500, or the Computer Audition Lab 500, which is made of 502 songs by 502 different artists. Each song is annotated by at least 3 listeners using a vocabulary of 135 tags. A song is labeled with a tag if there is at least 80% agreement between all listeners. According to [27], we consider the tags with more than 30 examples and this reduces the vocabulary to 97 tags, including 11 genres, 14 instruments, 25 acoustic qualities, 6 vocal characteristics, 35 emotions and 6 usages. On average each song is annotated with 23.6±5 tags.

The following three types of noises are considered to simulate different types of errors in social tagging:

- Random deletion (RD): a tag is randomly deleted with a probability ξ. RD is used to simulate the weak labeling property of social tagging.

- Random insertion (RI): a tag is randomly added with a probability ξ. RI is used to simulate the inaccuracy of social tagging. Note that we have assumed a uniform distribution of tag noise in this work for simplicity. For future work we intend to take into account the correlation between tags and draw random tags for addition from a nonuniform distribution, which may better capture the "real" noises induced by human tagging.

- Random deletion and insertion (RDI): the cascade of RD and RI; tags are randomly deleted and then randomly added, both with a probability ξ. We have interchanged the order of RD and RI and found that it does not matter.

The higher the noise rate ξ, the noisier the tagging matrix. We denote the corrupted tagging matrix as Y_ξ.

As suggested in [18] and [24], in addition to using traditional accuracy measures such as precision and recall [41] as evaluation criteria of music autotagging, a better alternative is to measure directly the extent to which an algorithm captures the *sought-after quantity* of music autotagging, which is a set of tags that can sufficiently identify a song, so that it can be readily cataloged and retrieved by users [18]. Therefore, we propose to use the following *tag-based music retrieval* scenario to evaluate the performance of retagging: for each song, rank the other songs in a descending order of semantic similarity in the tag space. To prevent the result from being dominated by popular tags, we apply term frequency-inverse document frequency weighting [31] to the tags and then compute the Euclidean distance. We then measure the consistency between the ranking order computed from the raw tagging matrix (which is considered as clean) and the one computed from the refined (retagged) tagging matrix using the Spearman rank correlation coefficient (or Spearman's ρ) [46]. Higher correlation indicates better recovery from the imposed noise. The evaluation process is repeated 10 times to get the average results.

We compare the following retagging methods,

- Baseline: do nothing.

- Content similarity (CS): use the graph-based label propagation algorithm described in Section 4.1.

- Low rank (LR): use the robust principal component analysis algorithm APG described in Section 4.2.

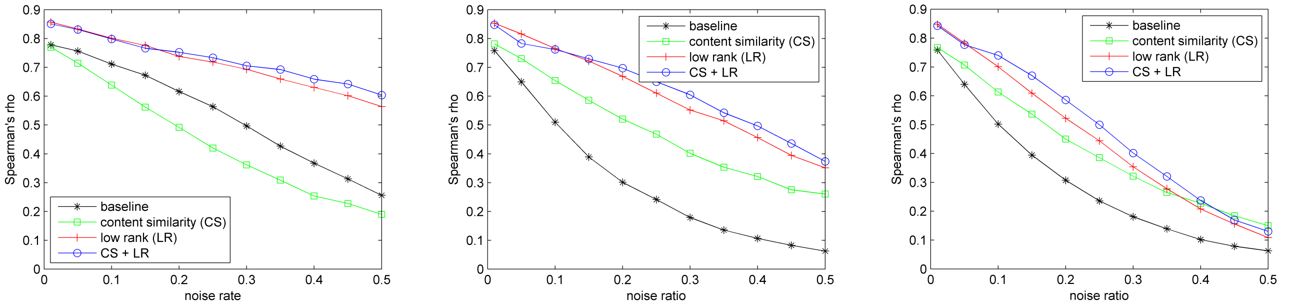
- CS+LR: use APG with content consistency constraint algorithm described in Section 4.3.

In other words, in the baseline method we directly use the corrupted matrix Y_ξ to perform tag-based music retrieval and compare the resulting ranking of songs to that resulted from using the raw matrix Y. While Y_ξ and Y are both binary matrices, the refined matrix A computed by the three other algorithms is real-valued. For a fair comparison, we rank the tags of each track based on their confidence scores and retain only the top K tags as the ultimate tags. We then compute the similarity between songs based on the binary version of A. In this work K is simply set to average number of tags per song observed in Y, namely 23.6.

A grid-search strategy has been employed to find the optimal set of parameters for each method. We find that the performance of CS is not very sensitive to α and setting it to 0.5 seems to perform well. For LR and CS+LR we search for λ_1 and λ_2 from $\{2^{-8}, 2^{-6}, \ldots, 2^0\}$ and find that setting $\lambda_1 = 2^{-2}$ and $\lambda_2 = 2^{-6}$ empirically performs well. All the above methods are implemented in Matlab. When executed on a Windows server with two octo-core AMD CPUs, the computation time for CS+LR is about 6.26±2.32 seconds. CS is very efficient as it has a closed-form solution.

From Fig. 2 we see that all retagging methods greatly outperform the baseline, except for CS when the noise is RD (more on this later). We can also observe that LR consistently outperform CS in most cases, and that the combination of CS+LR further boosts the performance slightly. We also see that even with rather severe noise and the tagging matrix being close to full or random, CS+LR is still able to discover some hidden patterns and recover the matrix to a certain level without using any additional or external information other than the content similarity of music pieces. When $\xi = 0.3$, the relative gain in terms of Spearman's ρ are 42.1%, 238%, and 122% for RD, RI, and RDI, respectively.

By comparing the results of the baseline method in Figs. 2(a)–(c) we find that the accuracy of tag-based music retrieval is more sensitive to random insertion noise than to random deletion one. This is possibly because, even when some relevant tags are removed, two songs could still be claimed similar if they share a critical and specific subset of tags. On the other hand, if too many irrelevant tags are added, the tag space would be severely corrupted. We also observe that the results of the baseline method are very similar in Figs. 2(b) and (c). This implies that RI exerts a greater influence than RD on tag-based music retrieval. Interestingly, a similar observation that "small and clean" tags performs better than "large but noisy" tags for music similarity applications has also been made before [16].



(a) ‘Random deletion’ noise (b) ‘Random insertion’ noise (c) ‘Random deletion and insertion’ noise

Figure 2: Performance of different retagging methods for tag-based music retrieval under different noise rates, using CAL500 and a vocabulary of 97 tags [41].

When the noise rate is small (e.g., ξ=0.01), using CS would not lead to much refinement and the resulting Spearman's ρ is close to that of the baseline. With a larger noise rate, CS outperforms the baseline by a great margin for RI and RDI. CS even slightly outperforms LR for RDI when $\xi \geq$0.35. This result shows that the content similarity is indeed an important cue in removing irrelevant tags. However, for RD the performance of CS is even worse than that of the baseline. We find that the amount of information that can be propagated by the graph-based label propagation algorithm is highly dependent on the initial information. When many tags are deleted by RD, for some songs there are less than K tags that have nonzero confidence scores and under such a circumstance random tags will be added so that each song is tagged by K tags. These randomly added tags deteriorate the result of CS for RD.

Fig. 3 shows the tag distributions (number of songs labeled with each tag) of the original tagging matrix and some retagged ones. It can be found that the retagging of RD by CS is indeed sparse and dominated by a few popular tags. Many other tags are in fact randomly added in the final step of retagging. The result of RI by CS is also sparse, but many of them simply have zero counts since it does not incur random addition of tags. On the contrary, the CS+LR algorithm is found to be robust against either RD or RI (Figs. 3(d) and (e)), producing a tag distribution that is similar to the original uncorrupted one.

Among the three types of noises, RDI may be the closest to the real world case observed in social tagging websites such as Last.fm. From Fig. 2(c) we see that CS+LR greatly improves the quality of a tagging matrix that is corrupted by an RDI noise with $<$ 0.5 error rate. In our evaluation of tag-based music retrieval, the Spearman rank correlation coefficient between the ground truth matrix and the retagged one is above 0.5 when the error rate is smaller than 0.25.

Generally speaking our evaluation shows that LR is more effective and robust than CS for the task of retagging. Modifying LR by adding a content consistent constraint (CS+LR) leads to an even better performance.

5.2 Evaluation on CAL10k

To evaluate retagging on a larger dataset, we employed the CAL10k collection [38]. It comprises 10,870 partially annotated songs by 4,597 artists. Each song is annotated using a vocabulary of up to 1,053 tags by expert musicologists hired by the music service company Pandora.[8] We collected the audio files of 7,069 songs in this collection and extracted their content features. On average each song is annotated with 11.5±4 tags. Considering the sparsity, the percentage of nonzero elements in the tagging matrix for CAL10k is 1.26%, whereas for CAL500 it is 24.4%. Namely, CAL10k is 19.4 times sparser. However, as opposed to the CAL500, we do not intend to reduce the sparsity by removing rare tags because large-scale tagged music collections are characterized by such sparsity.

Moreover, instead of using synthetic noises we apply retagging directly on the *original* tagging matrix and evaluate how the refined matrix leads to better performance for music auto-tagging. Specifically, we randomly hold out 50% of the dataset and use the remaining 50% as training data to build a simple k-nearest neighbors based autotagging model [34]. The precision and recall are measured by comparing original tags of the test songs to those predicted by original tags and improved tags (by retagging) of the training songs. For each test song, only tags that appear at least βk times in its top k neighbors in the training data are proposed, where $\beta \in [0, \; 1]$ is a voting threshold that influences the number of proposed tags. Empirically we find that using $\beta = 0.1$ has a good trade-off between precision and recall.

Fig. 4 shows the result as we vary k from 5 to 500 with an increasing step of 10. It can be observed that the recall rate is much improved when the tagging matrix has been retagged by CS+LR. We also see that, after retagging, the performance of autotagging is less sensitive to the value of k. Note that in this evaluation recall is more important than precision as CAL10k suffers from the weak labeling problem and therefore precision can be underestimated.

The main computational burden of APG is the singular value decomposition that needs to be performed in each iteration (cf. Eq. 8). The required CPU time for retagging the training set of CAL10k is 36.8±5.08 minutes, which is 353 times longer than that for CAL500 (contrastingly the size of CAL10k is 76.7 times larger than CAL500). Our simulation using Matlab programs shows that for a very large-scale tagged dataset whose size is 64k by 2.4k, APG requires around 8 hours to complete retagging. We are currently investigating other algorithms [30, 33] for better scalability.

[8] `http://www.pandora.com/`

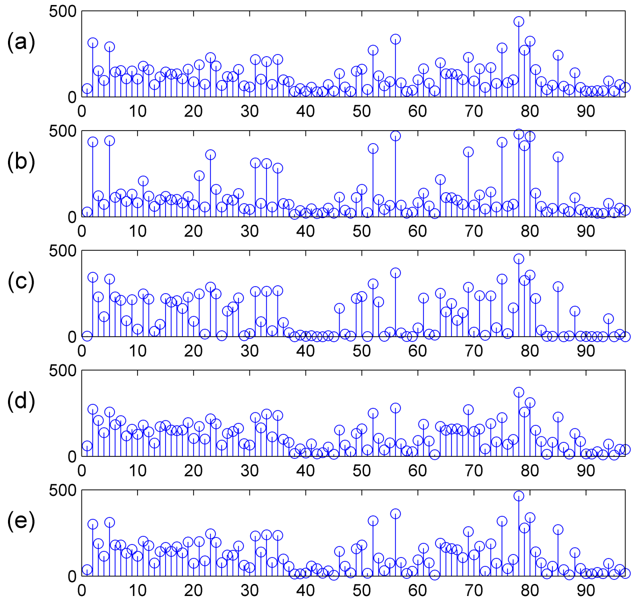


Figure 3: The tag distributions of (a) the original tagging of CAL500, (b) the retagged one of RD by CS, (c) the retagging of RI by CS, (d) the retagging of RD by CS+LR, (e) the retagging of RI by CS+LR. Numbers along the abscissa and the ordinate are the tag indexes and number of songs labeled with each tag, respectively. All the noise rate is 0.2.

6. CONCLUSIONS

In this paper we have introduced a novel music retagging framework for improving the performance of tag-based music information retrieval systems. We have also presented an empirical study that demonstrates the effectiveness of a number of music retagging algorithms. Our result shows that robust principal component analysis with content consistency constraint achieves the best performance and all the evaluated retagging algorithms improve the robustness of a tag-based system against erroneous assignment of music tags. Music retagging also improves the quality of expert-assigned tags and greatly increases the recall of a simple music auto-tagging algorithm. Experiments over larger-scale social tags is underway.

7. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for valuable comments that improved the quality of this paper. This work was supported by a grant from the National Science Council of Taiwan under contract NSC 100-2218-E-001-009 and partially supported by the following projects: Classical Planet: TSI-070100-2009-407 (MITYC), DRIMS: TIN2009-14247-C02-01 (MICINN) and MIRES: EC-FP7 ICT-2011.1.5 Networked Media and Search Systems, grant agreement No. 287711.

8. REFERENCES

[1] J. Aucouturier, B. Defreville, and F. Pachet. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *J. Acoustical Soc. America*, 122:881, 2007.

[2] J. Aucouturier, F. Pachet, P. Roy, and A. Beurivé. Signal+context=better classification. In *ISMIR*, 2007.

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

[4] D. Bogdanov, J. Serra, N. Wack, P. Herrera, and X. Serra. Unifying Low-Level and High-Level music similarity measures. *IEEE Trans. Multimedia*, 13(4):687–701, Aug. 2011.

[5] J. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optimization*, 20(4):1956–1982, 2010.

[6] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):1–37, June 2011.

[7] L. Cao, J. Luo, and T. S. Huang. Annotating photo collections by label propagation according to multiple similarity cues. In *ACM MM*, pages 121–130, 2008.

[8] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *PIEEE*, 96(4):668 –696, 2008.

[9] S.-K. Chai, J. Salerno, and P. L. Mabry. *Advances in Social Computing*. Springer, 2010.

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. American Soc. Info. Science*, 41(6):391–407, 1990.

[11] E. Gabrilovich and S. Markovitch. Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Int. Res.*, 34:443–498, 2009.

[12] G. Golub and C. Reinsch. *Handbook for Matrix Computation II, Linear Algebra*. Springer-Verlag, New York, 1971.

[13] J. Hollan, E. Hutchins, and D. Kirsh. Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.*, 7:174–196, 2000.

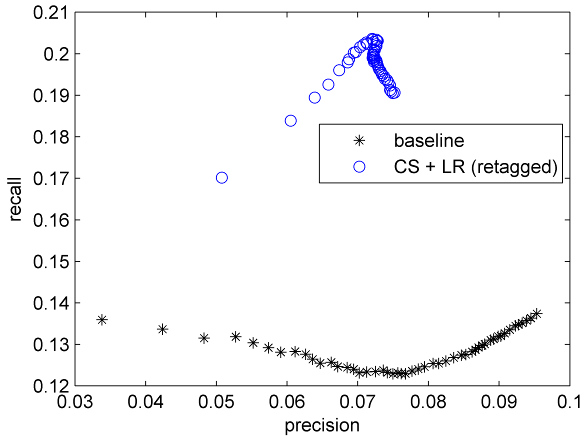


Figure 4: Precision and recall rates using the raw tagging (baseline) of CAL10k and the retagged one.

[14] J. Jia, N. Yu, X. Rui, and M. Li. Multi-graph similarity reinforcement for image annotation refinement. In *IEEE ICIP*, pages 993–996, 2008.

[15] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, K. Seyerlehner, and G. Widmer. Augmenting text-based music retrieval with audio similarity. In *ISMIR*, 2009.

[16] P. Lamere. Social tagging and music information retrieval. *J. New Music Res.*, 37(2):101–114, 2008.

[17] R. M. Larsen. Lanczos bidiagonalization with partial reorthogonalization. Technical Report DAIMI PB-357, Aarhus Univ., 1998.

[18] E. Law. The problem of accuracy as an evaluation criterion. In *Workshop on Evaluation Methods in Machine Learning*, 2008.

[19] E. Law and L. von Ahn. Input-agreement: A new mechanism for collecting data using human computation games. In *ACM Int. Conf. Human Factors in Computing Systems*, pages 1197–1206, 2009.

[20] M. Levy and M. Sandler. Music information retrieval using social tags and audio. *IEEE Trans. Multimedia*, 11:383–395, April 2009.

[21] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. Technical Report UILU-ENG-09-2214, 2009.

[22] D. Liu, X.-S. Hua, M. Wang, and H.-J. Zhang. Image retagging. In *ACM MM*, pages 491–500, 2010.

[23] M. I. Mandel, D. Eck, and Y. Bengio. Learning tags that vary within a song. In *ISMIR*, pages 399–404, 2010.

[24] G. Marques, M. Domingues, T. Langlois, and F. Gouyon. Three current issues in music autotagging. In *ISMIR*, 2011.

[25] B. McFee and G. Lanckriet. Learning multi-modal similarity. *J. Mach. Learning Res.*, 12:491–523, 2011.

[26] R. Miotto, L. Barrington, and G. Lanckriet. Improving auto-tagging by modeling semantic co-occurrences. In *ISMIR*, 2010.

[27] R. Miotto, L. Barrington, and G. Lanckriet. Improving auto-tagging by modeling semantic co-occurrences. In *ISMIR*, pages 297–302, 2010.

[28] S. Ness, A. Theocharis, G. Tzanetakis, and L. Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *ACM MM*, pages 705–708, 2009.

[29] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[30] B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. In *Optimization Online*, 2011.

[31] M. Schedl, T. Pohle, P. Knees, and G. Widmer. Exploring the music similarity space on the web. *ACM Trans. Inf. Syst.*, 29:1–24, 2011.

[32] D. Schuler. Social computing. *Communications of the ACM*, 37(1):28–29, Jan. 1994.

[33] S. Shalev-Shwartz, A. Gonen, and O. Shamir. Large-scale convex minimization with a low-rank constraint. In *ICML*, pages 329–336, 2011.

[34] M. Sordo, O. Celma, and D. Bogdanov. MIREX 2011: Audio tag classification using weighted-vote nearest neighbor classification. In *Music Information Retrieval Evaluation eXchange*, 2011.

[35] M. Sordo, F. Gouyon, and L. Sarmento. A method for obtaining semantic facets of music tags. In *Workshop on Music Recommendation and Discovery, ACM Int. Conf. Recommender Systems*, 2010.

[36] N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *Annual Conf. Learning Theory*, 2005.

[37] L. Steels. Collaborative tagging as distributed cognition. *Pragmatics Cognition*, 14:287–292(6), 2006.

[38] D. Tingle, Y. E. Kim, and D. Turnbull. Exploring automatic music annotation with acoustically objective tags. In *ACM MIR*, 2010.

[39] D. Turnbull, L. Barrington, and G. Lanckriet. Five approaches to collecting tags for music. In *ISMIR*, 2008.

[40] D. Turnbull, L. Barrington, G. Lanckriet, and M. Yazdani. Combining audio content and social context for semantic music discovery. In *ACM SIGIR*, pages 387–394, 2009.

[41] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query by semantic description using the CAL500 data set. In *ACM SIGIR*, pages 439—446, 2007.

[42] C. Wang, F. Jing, L. Z. 0001, and H.-J. Zhang. Content-based image annotation refinement. In *IEEE CVPR*, 2007.

[43] F.-Y. Wang, K. M. Carley, D. Zeng, and W. Mao. Social computing: From social informatics to social intelligence. *IEEE Intelligent Systems*, 22:79–83, 2007.

[44] J.-C. Wang, H.-S. Lee, H.-M. Wang, and S.-K. Jeng. Learning the similarity of audio music in bag-of-frames representation from tagged music data. In *ISMIR*, 2011.

[45] S. Yan, D. Xu, B. Zhang, H. jiang Zhang, Q. Yang, S. Member, and S. Lin. Graph embedding and extension: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:40–51, 2007.

[46] Y.-H. Yang and H. Chen. Ranking-based emotion recognition for music organization and retrieval. *IEEE Trans. Audio, Speech, and Lang. Processing*, 19(99):762–774, 2011.

[47] Y.-H. Yang and H. H. Chen. *Music Emotion Recognition*. CRC Press, Feb. 2011.

[48] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *NIPS*, pages 321—328, 2004.

[49] L. W. Zhouchen Lin, Minming Chen and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical Report UILU-ENG-09-2215, 2009.

[50] G. Zhu, S. Yan, and Y. Ma. Image tag refinement towards low-rank, content-tag prior and error sparsity. In *ACM MM*, pages 461–470, 2010.

[51] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.