IMPROVING MUSICAL CONCEPT DETECTION BY ORDINAL REGRESSION AND CONTEXT FUSION

Yi-Hsuan Yang, Yu-Ching Lin, Ann Lee, Homer Chen
National Taiwan University
affige@gmail.com, vagante@gmail.com, an918tw@yahoo.com.tw, homer@cc.ee.ntu.edu.tw

ABSTRACT

To facilitate information retrieval of large-scale music databases, the detection of musical concepts, or auto-tagging, has been an active research topic. This paper concerns the use of concept correlations to improve musical concept detection. We propose to formulate concept detection as an ordinal regression problem to explicitly take advantage of the ordinal relationship between concepts and avoid the data imbalance problem of conventional multi-label classification methods. To further improve the detection accuracy, we propose to leverage the co-occurrence patterns of concepts for context fusion and employ concept selection to remove irrelevant or noisy concepts. Evaluation on the cal500 dataset shows that we are able to improve the detection accuracy of 174 concepts from 0.2513 to 0.2924.

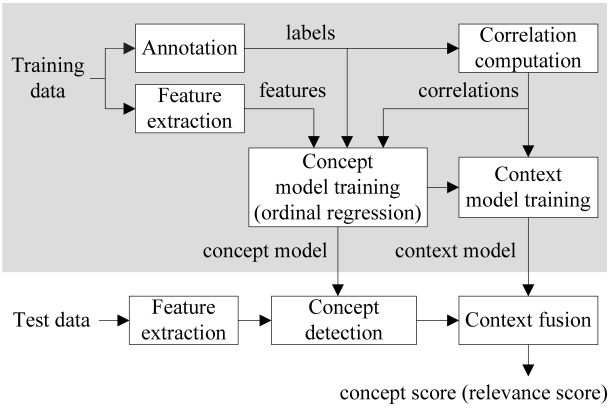

Figure 1. A schematic diagram of the proposed musical concept detection system.

1. INTRODUCTION

Music plays an important role in human's history, even more so in the digital age. Never before has such a large collection of music been created and accessed daily by people. Bridging the semantic gap–the chasm between raw data (signals) and high-level semantics (meanings)–is essential for exploiting the growing music content. Toward this goal, recent research has focused on building detectors for detecting musical *concepts* such as genre, emotion, and instrumentation using a pre-defined lexicon and a sufficient number of annotated examples [1–8]. Once trained, these detectors can be used to semantically tag and index music content in a fully automatic fashion. A user can then query music by *semantic description* [2], such as "find me a song that is brit poppy and alternative, features male vocal, and has a nice distorted electric guitar solo."

Early attempts to musical concept detection formulated the problem as a multi-label binary classification problem and trained detector independently for each concept [1–3]. The training data is annotated by human subjects and the relationship between ground truth and audio features is learnt by machine. Subsequent efforts went one step forward and utilized the correlation between concepts (either positive or negative) to improve concept detection. Duan *et al*. proposed a collective annotation scheme that trains additional models for the pairs of concepts that have strong correlations [4]. Bertin-Mahieux *et al*. studied a second-stage learning and a correlation reweighting scheme to boost the result of concept detection [5]. Aucouturier *et al*. [6] used decision tree to refine the result of individual detectors. Chen *et al*. built anti-models to exploit the negative correlations of concepts [7]. Modeling concept correlation has been shown effective for improving musical concept detection.

It is, however, noted that most existing works focus on the refinement of the individual detectors by training additional models rather than focus on the direct incorporation of concept correlation in training the individual detectors. Evidently, there are different levels of correlation between concepts, by which we can divide the training data into more than two categories; some of the training pieces should be more relevant to a target concept than other pieces. Consider the following toy example. We are training a concept detector of "happy" based on three training pieces a, b and c, which are annotated with "happy," "tender" and "sad" respectively. Conventional approaches formulate the problem as a *flat* binary classification, using a as positive example and b, c as negative examples. However, since "happy" is semantically closer to "tender," there should be an ordinal scale among them, $\mathsf{a} \succ \mathsf{b} \succ \mathsf{c}$, where $\succ$ denotes a relevance relationship. Such ordinal information is neglected by treating b and c the same.

In this paper, we propose to formulate concept detec-

tion as an *ordinal regression* problem [9–11] and train a concept model to estimate the *relevance score* of a song with respect to a target concept. A higher relevance score represents a higher probability of the song being annotated with the concept. The advantage of this approach is two-fold. First, we can make better use of the training data (whose collection process is fairly time-consuming and labor-intensive) by explicitly leveraging the ordinal relationship between concepts. Second, conventional classification algorithms are hampered by the so-called data imbalance problem: the performance of a classifier degrades significantly when the number of training data is not uniformly distributed across classes. For example, when 95% of the training data is negative, a classifier can achieve a 95% accuracy by simply classifying everything as negative, which is highly undesirable. This problem is usually observed for infrequent concepts such as "genre-swing" and "instrument-organ." Ordinal regression is free of this problem because the objective function of learning is not minimizing classification errors and because the training pieces that are annotated with semantically close concepts can still be leveraged in learning.

The second contribution of the paper is the investigation of context fusion and concept selection to improve the detection result. The basic idea of context fusion is to leverage the co-occurrence patterns between target semantic and peripherally related concepts to improve the result of an initial model. It has been successfully applied to improve visual concept detection and image search [12–14]. Because of the assumption that the result is presented in an ordered form, context fusion can be combined with ordinal regression in an elegant way. We also study a concept selection method to remove irrelevant concepts to improve context fusion. The number of selected concepts is target concept-dependent. For a concept that lacks strongly correlated concepts, context fusion is not applied.

A schematic diagram of the overall system is shown in Fig. 1. In the train phase, the annotations, features, and concept correlations are utilized to train the individual concept detectors by ordinal regression. We then exploit the contextual patterns among concepts to train a context detector for each concept. The concepts utilized in context fusion are selected based on concept correlations. In the test phase, we extract the features of the test data and then apply concept detection and context fusion in cascade to generate the detection result.

The paper is organized as follows. In Section 2 we describe the corpus adopted in this work and the concept correlations therein. The correlations are then used in Section 3 for ordinal regression and in Section 4 for context fusion. We report the experimental results in Section 5. Section 6 concludes the paper.

2. CORPUS AND CONCEPT CORRELATION

We use the Computer Audition Lab 500-Song (cal500) data set [2] in this study for it is publicly available. [1] The collection is made of 502 recent Western songs by 502 different artists chosen to cover a large amount of acoustic variation. 66 paid students were recruited to annotate the songs with a fixed vocabulary of 135 musical concepts, with each song annotated by at least three respondents. A song is annotated with a concept if there is at least 80% agreement between the respondents. The concept lexicon spans six semantic categories: 29 instruments, 22 vocal characteristics, 36 genres, 18 emotions, 15 acoustic qualities, and 15 usage terms. The concepts of emotions and acoustic qualities are further broken down into bipolar ones (e.g., "emotion-happy" and "emotion-NOT happy"), resulting in a total of 174 concepts [2].

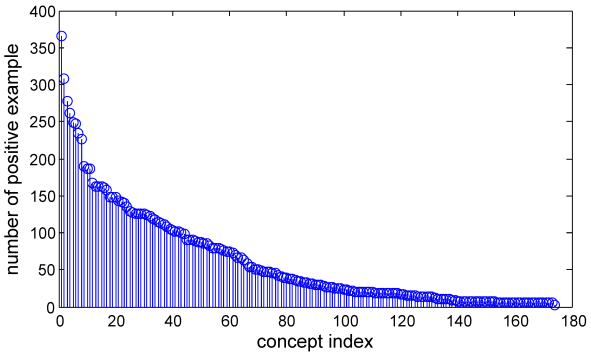

Figure 2. Concept frequency distribution of a subset (413 songs) of cal500 [2]. Note the concepts have been sorted by the number of positive examples.

We collect the audio files of 413 songs of cal500 and analyze the frequency of each concept. As Fig. 2 shows, rare concepts form a long tail in the concept frequency distribution. While frequent concepts (e.g., "song-recorded" and "instrument-male lead vocals") have more than 300 positive examples, 37 concepts have less than 10 positive examples. A preliminary evaluation also shows the detection accuracy of the infrequent concepts are particularly low. Because of this data imbalance problem, we also use a subset of concepts that have more than 50 positive examples in this study. The resulting lexicon, which consists of 69 concepts, is denoted as cal500-lite hereafter.

Given a concept lexicon $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ and N annotated examples $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, we can measure the pairwise correlation ρ_{mn} between two concepts c_m and c_n from the annotations A, which are represented by a $|\mathcal{C}| \times N$ binary matrix, with $A_{mi} = 1$ indicating that d_i is annotated with c_m. We compute ρ_{mn} by the Pearson's correlation coefficient of A_m and A_n,

$$\rho_{mn} = \frac{E((A_m - \mu_{A_m})(A_n - \mu_{A_n}))}{\sigma_{A_m}\sigma_{A_n}}. \qquad (1)$$

$\rho_{mn} \in [-1, 1]$ and $\rho_{mn} > 0$ iff there is a positive correlation. Interestingly, we find the correlation values generally follow a Laplacian-like distribution: the number of concept pairs decreases exponentially along with the absolute values of correlation. See Fig. 3.

[1] Available at: http://cosmal.ucsd.edu/cal/projects/AnnRet

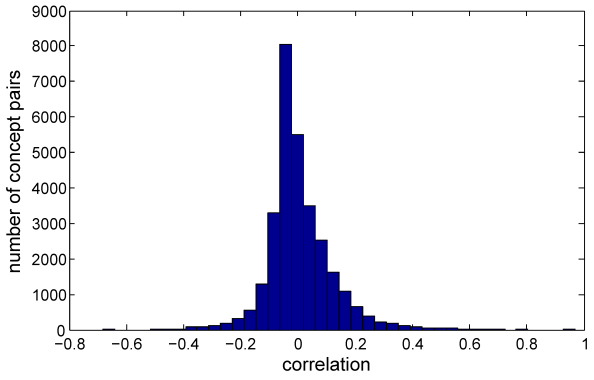

Figure 3. Distribution of the correlation values of cal500.

3. ORDINAL REGRESSION

3.1 Brief Review

Unlike classification, ordinal regression defines a number of classes that exhibits an *ordinal scale* among them. For example, the preference of a song can be categorized to "very dislike," "dislike," "neutral," "like," and "very like." The outcome space can be denoted as $\mathcal{Y} = \{r_1, \ldots, r_K\}$, with ordinal classes $r_K \succ_{\mathcal{Y}} r_{K-1} \succ_{\mathcal{Y}} \ldots \succ_{\mathcal{Y}} r_1$, where K is the number of classes. A closely related problem is ranking, which presents ordered results to a user in response to a query. A common example is the ranking of search results from the search engine (e.g., Google). Both ordinal regression and ranking assign each object a *relevance score*, by which the object is ranked. The difference is ordinal regression needs a further step that determines the class membership of each object with respect to the discrete ordinal classes.

In the seminal work of Herbrich *et al*., the ordinal classes were modeled by intervals on the real line [9]. A discriminative function $f : \mathcal{X} \mapsto \mathbb{R}$ was trained to predict the relevance score $\hat{y}_i = f(\mathbf{x}_i) = (\mathbf{w} \cdot \mathbf{x}_i)$, where $\mathbf{x}_i$ is a feature vector of an object and $\mathbf{w}$ is a vector of weights. However, because the outcome space $\mathcal{Y}$ is discrete, Herbrich *et al*. determined the rank boundary $\theta(r_k)$ between classes r_k and r_{k+1} on the real line according to the following heuristics,

$$\theta(r_k) = \frac{1}{2}(f(\mathbf{x}_1) + f(\mathbf{x}_2)), \tag{2}$$

$$(\mathbf{x}_1, \mathbf{x}_2) = \underset{(\mathbf{x}_i, \mathbf{x}_j) \in \Theta(k)}{\arg\min} [f(\mathbf{x}_i) - f(\mathbf{x}_j)], \tag{3}$$

where $\Theta(k)$ is the set of object pairs $(\mathbf{x}_i, \mathbf{x}_j)$ with $y_i = r_k$, $y_j = r_{k+1}$, and $(\hat{y}_i - \hat{y}_j)(y_i - y_j) \geq 0$. In other words, the optimal threshold $\theta(r_k)$ for rank r_k lies in the middle of the estimates of the closest objects of rank r_k and r_{k+1} that can be correctly ranked by $f(\cdot)$. After the estimation of the boundaries $\theta(r_k)$ a new object is assigned to an ordinal class according to the following equation,

$$g(\mathbf{x}_i) = r_k \Leftrightarrow f(\mathbf{x}_i) \in [\theta(r_{k-1}), \theta(r_k)]. \tag{4}$$

To learn $f(\cdot)$, Herbrich *et al*. viewed the problem as the classification of object pairs into two categories (correctly ranked and incorrectly ranked) and trained a support vector machine (SVM) to minimize the classification error $\sum_{i,j}^{N}(\hat{y}_i - \hat{y}_j)(y_i - y_j)$. Though this algorithm, generally called rankSVM, offers advantages, it is time-consuming as the operation on every possible pair is $O(N^2)$.

Alternatively, we employ the listNet [11] algorithm in this work. It uses score lists directly as learning instances and minimizes the listwise loss between the ground truth ranking list and the estimated one. In this way, the optimization is performed directly on the list and the computation complexity is reduced to $O(N)$. More specifically, to define a listwise loss function, the top-one probability is employed to transform a list of relevance scores into a probability distribution. The top one probability $P(y_i)$ of the ith object, defined as follows, represents the probability of the object being ranked on the top,

$$P(y_i) = \frac{\Phi(y_i)}{\sum_{i=1}^{N} \Phi(y_i)} = \frac{\exp(y_i)}{\sum_{i=1}^{N} \exp(y_i)}, \tag{5}$$

where $\Phi(\cdot)$ is an increasing and strictly positive function such as the exponential function. Modeling the list of scores as a probabilistic distribution, a metric such as the cross entropy can be used to measure the distance (listwise loss) between the ground truth list and the estimated one,

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{N} P(y_i) \log(P(f(\mathbf{x}_i))), \tag{6}$$

where $\mathbf{y} = \{y_i\}_{i=1}^{N}$ and $\hat{\mathbf{y}} = \{f(\mathbf{x}_i)\}_{i=1}^{N}$. The algorithm then learns the weighting vector $\mathbf{w}$ by updating it at a learning rate η by gradient descent,

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \times \Delta\mathbf{w}, \tag{7}$$

$$\Delta\mathbf{w} = \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{w}} = \sum_{i=1}^{N} (P(f(\mathbf{x}_i)) - P(y_i))\mathbf{x}_i. \tag{8}$$

It has been shown that listNet is more efficient and effective than rankSVM for a variety of ordinal regression and ranking problems, such as image/video search [13].

3.2 Concept Model Training by Ordinal Regression

Given the ground truth A_m of concept c_m and the feature representation of $\mathcal{D}$, typically a binary classifier $b_m(\cdot)$ is trained by treating $\mathcal{D}_m^+ = \{d_i | A_{mi} = 1\}$ as positive examples and $\mathcal{D}_m^- = \{d_i | A_{mi} = 0\}$ as negative examples. However, such a dichotomy of the training data loses many valuable information embedded in A_m, as we have illustrated in Section 1. We can in fact divide the training data to multiple ($K \geq 2$) ordinal classes according to concept correlations and then employ listNet to train a concept model $f_m(\cdot)$ for each concept c_m,

$$\hat{y}_{mi} = f_m(\mathbf{x}_i) = (\mathbf{w} \cdot \mathbf{x}_i). \tag{9}$$

Two such implementations are employed in this work, $K = 2$ and $K = 4$. The first one simply dichotomizes $\mathcal{D}$ as the binary classification setting. That is, $\mathcal{D}_m^{r_2} = \mathcal{D}_m^+$ and $\mathcal{D}_m^{r_1} = \mathcal{D}_m^-$. We then set $y_{mi} = r_k$ if $d_i \in \mathcal{D}_m^{r_k}$.

In this way, the concept correlations are not explicitly utilized, but thanks to the ordinal regression algorithm (which minimizes a listwise loss instead of clarification error) the data imbalance problem is avoided. The second implementation uses $K = 4$ and partitions $\mathcal{D}$ to four classes according to the following rules, which are listed in descending order of precedence,

- $\mathcal{D}_m^{r_4} = \mathcal{D}_m^+$
- $\mathcal{D}_m^{r_1} = \{d_i | A_{ni} = 1, \rho_{mn} \leq l, d_i \in \mathcal{D}_m^-\}$
- $\mathcal{D}_m^{r_3} = \{d_i | A_{ni} = 1, \rho_{mn} \geq u, d_i \in \mathcal{D}_m^- \setminus \mathcal{D}_m^{r_1}\}$
- $\mathcal{D}_m^{r_2} = \mathcal{D}_m^- \setminus \bigcup\{\mathcal{D}_m^{r_1}, \mathcal{D}_m^{r_3}\}$

In other words, $\mathcal{D}_m^{r_1}$ consists of songs that are annotated with any of the concepts $\mathcal{C}_m^{r_1}$ that are strongly negatively correlated with c_m, and $\mathcal{D}_m^{r_3}$ consists of songs that are annotated with any of the concepts $\mathcal{C}_m^{r_3}$ that are strongly positively correlated with c_m. In this work, we set $l = \mu_\rho - \sigma_\rho$, $u = \mu_\rho + \sigma_\rho$, where $\mu_\rho \simeq 0.01$ and $\sigma_\rho \simeq 0.11$ are the mean and the standard deviation of all the correlation values of the concept corpus (see Fig. 3).

Table 2 shows some highly correlated concepts for four different target concepts. It can be found that most of the correlated concepts are intuitively correct.

4. CONTEXT FUSION

The nature of concept detection makes it possible to discover co-occurrence patterns through mining ground truth annotations and utilize the patterns to improve concept detection. For example, if a song has the concepts "song-high energy" and "song-heavy beat," it is very likely that it also has the concept "song-fast tempo." If the relevance score of "song-fast tempo" is somehow detected low (maybe the detector is less reliable), we can modify the result by increasing the value. We refer to such a model that learns the co-occurrence patterns as the context model.

Following the idea of discriminative model fusion (DMF) [12, 13], we train a context model for each concept based on the output of the concept models. For each song, the $|\mathcal{C}|$ concept models are employed to predict the relevance score of song d_i with respect to each concept; this results in a $|\mathcal{C}|$-dimensional *model vector* $\mathbf{v}_i = \{\hat{y}_{ni}\}_{n=1}^{|\mathcal{C}|} = \{f_1(\mathbf{x}_i), \cdots, f_{|\mathcal{C}|}(\mathbf{x}_i)\}$. We use the model vectors to train the context model $\tilde{f}_m(\cdot)$ for each concept c_m by minimizing the loss between $\{y_{mi}\}_{i=1}^N$ and $\{\tilde{f}_m(\mathbf{v}_i)\}_{i=1}^N$ using listNet. We then replace $\hat{y}_{mi}$ with $\tilde{f}_m(\mathbf{v}_i)$. That is,

$$\hat{y}_{mi} \leftarrow \tilde{f}_m(\mathbf{v}_i) = (\tilde{\mathbf{w}} \cdot \mathbf{v}_i) = \sum_{n=1}^{|\mathcal{C}|} \tilde{w}_n f_n(\mathbf{x}_i). \quad (10)$$

Therefore, $\tilde{f}_m(\mathbf{v}_i)$ can be regarded as the weighted combination of the relevance scores of d_i with respect to other concepts. Intuitively, the absolute value of $\tilde{w}_n$ would be large if c_n is highly correlated with c_m. A total of $|\mathcal{C}|$ context models are trained.

TRAINING PHASE

INPUT: training data $\mathcal{D}$, A, $\{\mathbf{x}_i\}_{i=1}^N$, parameters K, θ
  compute correlations $\{\rho_{mn}\}_{m,n}^{|\mathcal{C}|}$ by Eq. 1.
  for $m = 1$ to $|\mathcal{C}|$
    partition $\mathcal{D}$ to K classes by A_m and $\{\rho_{mn}\}_{n=1}^{|\mathcal{C}|}$
    set $y_{mi} = r_k$ if $d_i \in \mathcal{D}_m^{r_k}$
    train $f_m(\cdot)$ by minimizing $L(\{y_{mi}\}, \{f_m(\mathbf{x}_i)\})$
  end
  for $m = 1$ to $|\mathcal{C}|$
    construct $\mathbf{v}_{mi} = \{f_n(\mathbf{x}_i)\}_{n:\text{abs}(\rho_{mn})\geq\theta}$
    train $\tilde{f}_m(\cdot)$ by minimizing $L(\{y_{mi}\}, \{\tilde{f}_m(\mathbf{v}_i)\})$
  end
OUTPUT: concept and context models $\{f_m, \tilde{f}_m\}_{m=1}^{|\mathcal{C}|}$

TEST PHASE

INPUT: test data $\{\mathbf{x}_z\}$
  for $m = 1$ to $|\mathcal{C}|$
    predict $\hat{y}'_{mz} = f_m(\mathbf{x}_z)$
  end
  for $m = 1$ to $|\mathcal{C}|$
    construct $\mathbf{v}_{mz} = \{\hat{y}'_{mz}\}_{n:\text{abs}(\rho_{mn})\geq\theta}$
    predict $\hat{y}_{mz} = \tilde{f}_m(\mathbf{v}_{mz})$
  end
OUTPUT: concept scores $\{\hat{y}_{mz}\}$ (one can get binary result with the boundary $\theta(r_{K-1})$; see Eqs. 2–4)

Table 1. Pseudo codes of the concept detection framework.

We also study concept selection by removing concepts whose absolute correlation values to the target concept are below a threshold θ. That is,

$$\mathbf{v}_{mi} = \{f_n(\mathbf{x}_i)\}_{n:\text{abs}(\rho_{mn})\geq\theta}, \quad (11)$$

where abs$(\cdot)$ is an operator that takes the absolute value. Intuitively, the number of selected concepts $|\mathbf{v}_{mi}|$ decreases as θ is set larger and the actual number of $|\mathbf{v}_{mi}|$ depends on c_m but not on d_i. When $\theta = 0$, no concept selection is performed and all the concepts are utilized; when $\theta = 1$, no context fusion is conducted. For a concept that does not have strongly correlated concepts, $|\mathbf{v}_{mi}|$ would equal zero and we do not apply context fusion to it.

The algorithmic descriptions of the proposed concept detection framework is shown in Table 1.

5. EXPERIMENTAL RESULT

5.1 Experiment Setup

For fair comparison, each songs is converted to a standard format (22,050 Hz sampling frequency, 16 bits precision and mono channel) and represented by a 30-second segment starting from the initial 30th second of the song, a common practice in music classification.

For feature representation of a song we use the computer program MA toolbox [15] to extract Mel-frequency cepstral coefficients (MFCC), one of the most popular feature representation for audio signal processing. It is computed by taking the cosine transform of the short-term log

target concept $c_m = \mathcal{C}_m^{r_4}$	strongly positively correlated concepts $\mathcal{C}_m^{r_3}$	strongly negatively correlated concepts $\mathcal{C}_m^{r_1}$
emotion: angry/agressive	emotion: exciting/thrilling, powerful/strong genre: metal/hard rock, hip hop/rap, punk instrument: drum machine, electric guitar (distorted) song: fast tempo, heavy beat, high energy;	emotion: calming, laid-back, happy, loving, positive, tender instrument: piano song: positive feelings, texture acoustic
emotion: sad	emotion: calming/soothing, emotional/passionate instrument: female lead vocals song: quality, texture acoustic usage: going to sleep, intensive listening	emotion: arousing, carefree, happy instrument: drum set, male lead vocals song: high energy, positive feelings usage: cleaning the house
genre: jazz	emotion: calming, laid back, pleasant, tender, touching genre: bebop, contemporary R&B, cool jazz, swing instrument: piano, saxophone, trombone, trumpet	emotion: not calming, not loving genre: rock instrument: male lead vocals
song: very danceable	emotion: arousing, carefree, exciting, happy, light genre: dance pop, funk, swing, hip-hop/rap, pop, R&B usage: at a party, exercising	emotion: calming, laid-back, sad, tender genre: alternative, soft rock, rock

Table 2. Using the rules described in Section 3.2, we can obtain the highly correlated concepts for a target concept and partition the training data to four classes. This table shows some (only partial) highly correlated concepts for four concepts.

power spectrum expressed on a nonlinear perceptual-related mel-frequency scale. We use the default 23ms frame size with half overlapping to compute a bag of 20-dimensional MFCC vectors and then collapse the sequence of feature vectors into a single feature vector by taking the mean and standard deviation. As prior works [2], we also take the first-order derivatives of the MFCC vectors to capture temporal information, resulting in a 80-dimensional feature vector $\mathbf{x}_i$ for each song.

We randomly hold out 100 songs as the test set and use the remaining 313 songs for training. The evaluation process is repeated 100 times to compute the average accuracy, which is measured by average precision (AP), the approximation of the area under the recall/precision curve [10]. Let $\hat{\mathbf{p}}_m = \{\text{rank}(\hat{y}_{mi})\}_{i=1}^N$, where $\text{rank}(\hat{y}_{mi})$ is the ranking order of d_i in $\mathcal{D}$ according to $\hat{y}_{mi}$, we have

$$AP(\hat{\mathbf{p}}_m, A_m) = \frac{1}{rel} \sum_{j:A_{mj}=1} Prec@j, \qquad (12)$$

where $rel = |i : A_{mi} = 1|$ is the number of relevant objects (true positives) of concept c_m, and $Prec@j$ is the percentage of relevant objects in the top j objects in predicted ranking $\hat{\mathbf{p}}_m$. AP equals 1 when all the relevant objects are ranked at top. Since AP only shows the performance of a concept, we evaluate the performance in terms of mean average precision (MAP), the mean of APs for all concepts.

5.2 Evaluate Ordinal Regression

We first compare the performance of ordinal regression and multi-label classification. We use listNet for ordinal regression and SVM for multi-label classification. [2] Table 3 shows the MAP of different learning algorithms. It can be found that listNet(K=2) significantly outperforms SVM (p-value<0.01) for both the cal500 and cal500-lite lexicons, showing the effectiveness of ordinal regression. Setting $K = 4$ and leveraging concept correlation to the training process further improves the accuracy. The relative gain of listNet(K=4) over SVM is +10.89% and +9.35% for cal500 and cal500-lite, respectively.

[2] We use SVM for its superior performance in classification problems. We implement it based on the LIBSVM library [16]. The parameters are tuned by a cross validation procedure to achieve better result: for SVM, we set the cost parameter C to 1000 and the gamma γ in the RBF kernel to 0.01; for listNet, we set the learning step η to 0.05.

	SVM	listNet(K=2)	listNet(K=4)
cal500	**0.2513**	0.2769	**0.2787**
cal500-lite	0.4323	0.4687	0.4727

Table 3. Evaluation of ordinal regression.

	SVM	listNet(K=4)
the 40 most freq. cpts	0.5113	0.5523 (+8.02%)
the medium freq. cpts	0.2182	0.2460 (+12.74%)
the 40 least freq. cpts	0.0690	0.0818 (+18.47%)
average	0.2513	0.2787 (+10.89%)

Table 4. The accuracy of concept detection for the cal500 concepts of different concept frequencies.

To investigate the detection accuracy of ordinal regression for concepts of different frequencies, we break down the cal500 concept lexicon to three groups: the 40 most frequent ones, the 40 least frequent ones, and the others. Table 4 shows the MAP of the concept groups of SVM and listNet(K=4). The correlations between concept frequency, accuracy of concept detection, and the relative performance gain of listNet(K=4) over SVM are salient. The detection accuracy is generally higher for frequent concepts, while the relative performance gain of listNet(K=4) is generally higher for rare concepts. This implies that the data imbalance problem is mitigated by listNet.

5.3 Evaluate Context Fusion and Concept Selection

We then evaluate the performance of context fusion (using DMF) with and without concept selection. We use listNet to train both the concept models and context models and vary the value of the concept selection threshold θ. Results shown in Table 5 lead to the following obser-

	cal500	cal500-lite
listNet(K=4)	0.2787	0.4727
listNet(K=4)+DMF(θ=0)	0.2829	0.4873
listNet(K=4)+DMF(θ=0.1)	0.2911	**0.4882**
listNet(K=4)+DMF(θ=0.2)	**0.2924**	0.4854
listNet(K=4)+DMF(θ=0.3)	0.2856	0.4824
listNet(K=4)+DMF(θ=0.5)	0.2784	0.4754

Table 5. Evaluation of context fusion with different values of threshold θ (smaller θ selects more concepts).

vations. First, with mild concept selection, context fusion greatly improves concept detection. The MAP reaches 0.2924 (+4.92%) for cal500 and 0.4882 (+3.28%) for cal500-lite. This degree of performance gain is similar to that of applying context fusion to visual concept detection [13]. Second, without concept selection (θ=0), the performance of context fusion for cal500-lite is similar to the optimal one 0.4882, which may result from the fact that the detection accuracy of cal500-lite is generally high and thus directly leveraging all concepts is effective. On the contrary, due to the rather inconsistent accuracy, the detection of cal500 calls for concept selection to remove irrelevant concepts. Finally, setting θ too large removes most of the concepts and degrades accuracy. A mild value of θ exhibits the best result.

Table 6 shows the MAP of different semantic categories with and without context fusion. It can be found that context fusion with concept selection consistently improves all the semantic categories, especially for "emotion," "genre," and "usage." In particular, because the detection accuracy of "genre" and "usage" are relatively low, concept selection is prerequisite for context fusion to be effective. In addition, due to the lack of strongly correlated concepts, context fusion does not improve the category "vocal." Another interesting observation is the selected concepts often belong to "emotion," "song," or the same semantic category as the target concept. This evaluation demonstrates the importance of context fusion and concept selection.

6. CONCLUSION

In this paper, we have presented a novel framework of utilizing concept correlations to improve musical concept detection. A concept model is trained by an ordinal regression algorithm, which effectively utilizes the ordinal relationships among concepts and avoids the data imbalance problem of the commonly-used classification methods. A context model is then trained to improve the detection result by leveraging the co-occurrence patterns among concepts. We also employ a concept selection method to keep irrelevant concepts from being used in context fusion. Experimental results show that ordinal regression outperforms the conventional multi-label classification method by a great margin; a +10.89% relative gain in mean average precision is achieved. With mild concept selection, context fusion further improves the detection accuracy to 0.2924 for the 174 musical concepts of cal500.

	listNet	+DMF(θ=0)	+DMF(θ=0.2)
emotion	0.4272	0.4369 (+2%)	0.4522 (**+6%**)
genre	0.1731	0.1769 (+2%)	0.1890 (**+9%**)
instrument	0.2321	0.2345 (+1%)	0.2383 (+3%)
song	0.4233	0.4302 (+2%)	0.4345 (+3%)
usage	0.1753	0.1750 (0%)	0.1952 (**+11%**)
vocal	0.1981	0.1998 (+1%)	0.1997 (+1%)
average	0.2787	0.2829 (+2%)	0.2924 (+5%)

Table 6. The accuracy of concept detection for the cal500 concepts of different semantic categories.

7. ACKNOWLEDGEMENT

This work was supported by the National Science Council of Taiwan under contract NSC 97-2221-E-002-111-MY3.

8. REFERENCES

[1] B. Whitman and R. Rifkin: "Musical query-by-description as a multicalss leanring problem," *MMSP*, pp. 153–156, 2002.

[2] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet: "Semantic annotation and retrieval of music and sound effects," *IEEE Trans. Audio, Speech and Language Processing*, Vol. 16, No. 2, pp. 467–476, 2008.

[3] M. I. Mandel and D. P. W. Ellis: "Multiple-instance learning for music information retrieval," *ISMIR*, 2008.

[4] Z.-Y. Duan, L. Lu, and C.-S. Zhang: "Collective annotation of music from multiple semantic categories," *ISMIR*, pp. 237–242, 2008.

[5] T. Bertin-Mahieux et al: "Autotagger: A model for predicting social tags from acoustic features on large music databases," *J. New Music Research*, Vol. 37, No. 2, pp. 115–135, 2008.

[6] J.-J. Aucouturier, F. Pachet, P. Roy, and A. Beurivé: "Signal+context=better classification," *ISMIR*, 2007.

[7] Z.-S. Chen, J.-M. Zen, and J.-S. Jang: "Music annotation and retrieval system using anti-models," *AES Convention*, 2008.

[8] E. Law et al, "Tagatune: a game for music and sound annotation," *ISMIR*, pp. 361–364, 2007.

[9] R. Herbrich et al: "Support vector learning for ordinal regression," *ICANN*, pp. 97–102, 1999.

[10] Y. Yue et al: "A support vector method for optimizing average precision," *SIGIR*, pp. 271–278, 2007.

[11] F. Xia et al: "Listwise approach to learning to rank: Theory and algorithm," *ICML*, pp. 1192–1199, 2008.

[12] J. Smith, M. Naphade, and A. Natsev: "Multimedia semantic indexing using model vectors," *ICME*, pp. 445–448, 2003.

[13] Y.-H. Yang et al: "Online reranking via ordinal informative concepts for context fusion in concept detection and video search," *IEEE Trans. Circuits and Sys. for Video Tech.*, 2009.

[14] M. Naphade et al: "Large-scale concept ontology for multimedia," *IEEE Multimedia Magazine*, Vol. 13, No. 3, pp. 86–91, 2006.

[15] E. Pampalk: "A Matlab toolbox to compute music similarity from audio," *ISMIR*, 2004. http://www.ofai.at/ elias.pampalk/ma/.

[16] C.-C. Chang and C.-J. Lin: "LIBSVM: a library for support vector machines," 2001.