Learning Naive Bayes Classifiers for Music Classification and Retrieval

Zhouyu Fu, Guojun Lu, Kai Ming Ting, Dengsheng Zhang
Gippsland School of IT, Monash University
{*zhouyu.fu,guojun.lu,kaiming.ting,dengsheng.zhang*}*@infotech.monash.edu.au*

Abstract

In this paper, we explore the use of naive Bayes classifiers for music classification and retrieval. The motivation is to employ all audio features extracted from local windows for classification instead of just using a single song-level feature vector produced by compressing the local features. Two variants of naive Bayes classifiers are studied based on the extensions of standard nearest neighbor and support vector machine classifiers. Experimental results have demonstrated superior performance achieved by the proposed naive Bayes classifiers for both music classification and retrieval as compared to the alternative methods.

1. Introduction

Music classification is an emerging area in multimedia and information retrieval. A key problem in music classification is how to efficiently and effectively extract low level audio features for high level classification. Current music classification systems follow a local model for audio feature extraction [6, 3]. The audio signal for a song is split into local windows and features are extracted from each window. A single feature vector is obtained for each song by aggregating the features extracted from the local windows. This process is illustrated in Figure 1, where $\mathbf{x}_1,\ldots,\mathbf{x}_n$ denote local feature vectors, $\mathbf{x}$ represents the global song-level feature vector produced from local features $\mathbf{x}_j$'s using the aggregation function f. f usually takes simple forms such as mean or median [6, 3]. The song level feature vector $\mathbf{x}$ can then be used for classification by different classifiers like the K-Nearest Neighbor (KNN) [6] and Support Vector Machine (SVM) classifiers [3].

Despite the simplicity and effectiveness of the above feature aggregation approach, there is inevitable loss of discriminant information in compressing the feature set into a single feature vector. To fully exploit the information from all local features, we need a classification framework that can deal with the classification of feature sets. The Naive Bayes (NB) classifier is one such framework that has been widely used in text and image classification [5, 1]. It assumes that each feature vector in the feature set is independently generated with identical distribution. Class labels are then predicted by maximizing the likelihood function for the posterior probability. Since NB provides a general classification framework, the classifier can take different forms, depending on how the distribution is modeled. We have studied two such variants for music classification in this paper, namely NB Nearest Neighbor (NBNN) and NB Support Vector Machine (NBSVM). To the best of our knowledge, the use of NB framework for music classification based on feature set classification has not yet been investigated.

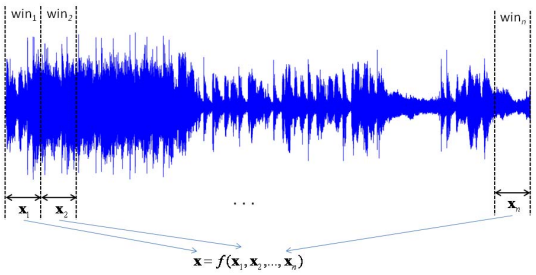

Figure 1. Local model for audio feature extraction

2. Naive Bayes Classification Framework

In this section we first derive the general form of the NB classification framework and subsequently introduce two different implementations using nearest neighbor and SVM. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote the set of local feature vectors obtained from a query song, we want to find its class C. This can be formulated as a Maximum-a-Posteriori (MAP) problem, where feature set X is assigned to the class that maximizes the posterior probability $p(C|X)$. Assuming equal class prior $p(C)$, the MAP problem then reduces to Maximum

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.1121

Likelihood (ML) estimation of the conditional probability $p(X|C)$. The NB assumption specifies a simple probability model that the feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ extracted from each song are generated independently from each class C with identical distribution Therefore we have the following general classification rule

$$\begin{aligned}\hat{C} &= \arg\max_C p(X|C) \\ &= \arg\max_C \prod_j p(\mathbf{x}_j|C) \\ &\equiv \arg\min_C -\sum_j \log p(\mathbf{x}_j|C)\end{aligned} \tag{1}$$

Hence, the problem is converted to the estimation of $p(\mathbf{x}_j|C)$, the conditional probability of local feature $\mathbf{x}_j$ for each class C. Depending on how $p(\mathbf{x}_j|C)$ is modeled, different NB classifiers can be defined.

Note that Equation 1 describes an extension of the standard NB framework, which is used for the classification of features by assuming each attribute is generated independently given the class label. Here we have generalised the NB framework to deal with the classification of collections of feature vectors.

2.1. Naive Bayes Nearest Neighbor (NBNN)

One option is to use the nearest neighbor classifier within the NB framework [1]. According to [1], $p(\mathbf{x}|C)$ in Equation 1 is estimated by a kernel density estimator with Gaussian kernel

$$\hat{p}(\mathbf{x}|C) = \frac{1}{N_c}\sum_i \exp\left(-||\mathbf{x} - \mathbf{x}_i^C||^2\right) \tag{2}$$

where $\mathbf{x}_i^C$ is the ith local feature vector taken from class C, N_c is the total number of local features in class C. Since only the instances close to $\mathbf{x}$ would contribute significantly to the density estimator $\hat{p}(\mathbf{x}|C)$, the above equation can be further simplified by keeping only the nearest instance in the summation term on the right-hand-side (RHS).

$$\hat{p}(\mathbf{x}|C) = \exp\left(-||\mathbf{x} - NN^C(\mathbf{x})||^2\right) \tag{3}$$

where $NN^C(\mathbf{x})$ denotes the nearest neighbor of local feature $\mathbf{x}$ from class C. The classification rule in Equation 1 can then be reduced to

$$\hat{C} = \arg\min_C \sum_j ||\mathbf{x}_j - NN^C(\mathbf{x}_j)||^2 \tag{4}$$

2.2. Naive Bayes SVM (NBSVM)

In this section, we derive a novel NB classifier based on the Support Vector Machine (SVM). Suppose the conditional probability terms $p(\mathbf{x}_j|C)$ in Equation 1 are from exponential family distributions, we have

$$p(\mathbf{x}_j|C) = \frac{1}{Z}\exp\left(\mathbf{w}^C\phi(\mathbf{x}_j) + b^C\right) \tag{5}$$

where $\phi(.)$ specifies a transformation on feature $\mathbf{x}$, $\mathbf{w}^C$ and b^C are the parameters, $Z = \int_{\mathcal{X}} \exp\left(\mathbf{w}\phi(\mathbf{x}_j) + b\right)$ is the partition function that makes $p(\mathbf{x}|C)$ a proper density. Note Z is a constant factor and can thus be ignored in subsequent computation. Substituting $p(\mathbf{x}_j|C)$ in the above equation into the NB classifier defined in Equation 1, we then have

$$\begin{aligned}\hat{C} &= \arg\max_C \sum_j f^C(\mathbf{x}_j) \\ f^C(\mathbf{x}_j) &= \mathbf{w}^C\phi(\mathbf{x}_j) + b^C\end{aligned} \tag{6}$$

Here $f^C(\mathbf{x}_j)$ can be regarded as the generalized linear discriminant function defined on the local feature space for class C. The larger the value of $f^C(\mathbf{x}_j)$, the more likely that feature $\mathbf{x}_j$ is generated from class C. The final classification rule on feature set X is basically the aggregation of decision values from individual local instances. This is inherently different from feature level aggregation mentioned in the introduction section, since all features are used to produce intermediate decisions to vote on the final decision.

Ideally, we want $f^C(\mathbf{x}_j)$ to be large for the correct class and small for the incorrect class with a large margin between them. This can be naturally formulated within the framework of SVM. For class C, the discriminant function $f^C(\mathbf{x})$ is learned by solving the following optimization problem

$$\begin{aligned}\min_{w^C, b^C} \quad & \lambda||w^C||^2 + \sum_i \ell(y_i, X_i) \\ \ell(y_i, X_i) &= \max\left(1 - y_i\sum_j (w^C\phi(\mathbf{x}_{i,j}) + b^C), 0\right)\end{aligned} \tag{7}$$

where y_i defines an auxiliary label for example i, $y_i = 1$ if and only if example i belongs to class C. $\ell(y_i, X_i)$ specifies the set-level loss function. The above is the primal formulation of NBSVM with implicit feature mapping ϕ. It can be converted to the following dual form [1]

$$\begin{aligned}\max_{\alpha^C} \quad & 2\sum_i \alpha_i^C - \sum_{i,p} \alpha_i^C\alpha_p^C y_i y_p \mathcal{K}(X_i, X_p) \\ \mathcal{K}(X_i, X_p) &= \sum_{\mathbf{x}_{i,j}\in X_i}\sum_{\mathbf{x}_{p,q}\in X_p} k(\mathbf{x}_{i,j}, \mathbf{x}_{p,q}) \\ s.t. \quad & 0 \le \alpha_i^C \le \frac{1}{2\lambda} \quad \text{and} \quad \sum_i \alpha_i^C y_i = 0\end{aligned} \tag{8}$$

[1] Due to page limit the details of derivation are omitted here. The derivation is quite similar to that of the standard SVM, which can be found in the tutorial article [2]

where $k(\mathbf{x}_{i,j}, \mathbf{x}_{p,q}) = \langle \phi(\mathbf{x}_{i,j}), \phi(\mathbf{x}_{p,q}) \rangle$ is the kernel function defined on the local features. It represents the inner product between the two local feature vectors in the transformed feature space defined by the implicit feature map $\phi(.)$.

The dual problem for NBSVM defined above is very similar to the dual problem for standard SVM. The only difference is in the kernel function $\mathcal{K}(X_i, X_p)$, which is defined on feature sets in NBSVM. The solution to the above dual problem naturally leads to the solution of the primal problem in Equation 7. The discriminant function $f^C(\mathbf{x})$ is then given by

$$f^C(\mathbf{x}) = \sum_i \alpha_i^C y_i \sum_{j \in X_i} k(\mathbf{x}, \mathbf{x}_{i,j}) + b \qquad (9)$$

For multiclass problems, a function $f^C(.)$ is learned for each class C using the same technique presented above. The final prediction rule is specified by Equation 6 by plugging the $f^C(.)$ functions.

3. Experimental Results

We now present results of the experiments conducted to demonstrate the performances of the proposed NBNN and NBSVM against standard NN and SVM. The benchmark GTZAN data set was used in the experiment [6]. It contains 1000 song clips in 30 seconds of length from 10 genres. The audio signal of each song was split into 1 second local windows. From each local window, the Mel-Frequency Cepstrum Coefficient (MFCC) summary feature vector [3] was extracted to form the set of local features. For the standard NN and SVM classifiers, a song-level feature vector was produced for each song which takes the average of local features in the song.

The performance of each algorithm has been evaluated for two different tasks of music classification and retrieval. Both tasks involve classifier training and testing. For each task, we repeated training and testing 20 times with different random partitioning of training and testing data. For each training and testing round, half of the examples in the data set were randomly selected for training and the remaining for testing. All methods being compared were tested on the same training and testing set in each round. The LibSVM package [2] was used for SVM training. The Gaussian kernel in the RHS of Equation 2 was adopted for both SVM and NBSVM. SVM parameters γ and λ were chosen via 5 fold cross validation on the training data.

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

3.1. Classification Performance

For music classification, we are interested in the overall prediction accuracy for the methods under comparison. Hence the classification accuracy rates were used as the performance measure. Table 1 reports the accuracy rates produced by different methods averaged over 20 rounds along with the standard deviation values. It shows that both NB classifiers achieved higher accuracy rates than their non-NB counterparts. Whereas NBNN has just obtained a margin of advantage over NN, NBSVM is significantly better than SVM in the classification performance. To further justify this, we have run two paired t-tests, one on the accuracy rates returned by NN and NBNN over the 20 rounds, and the other on accuracies returned by SVM and NBSVM. The t-test results show that the differences in classification accuracies are statistically significant within 95% of confidence interval for both cases - NBNN versus NN and NBSVM versus SVM. This is clearly indicated from the returned p values in Table 1.

NN	NBNN	p-value
$\mathbf{61.07 \pm 1.68\%}$	$\mathbf{62.34 \pm 1.93\%}$	$\mathbf{0.029}$
SVM	NBSVM	p-value
$\mathbf{73.33 \pm 1.97\%}$	$\mathbf{77.54 \pm 1.69\%}$	$\mathbf{1.69e-11}$

Table 1. Comparison of classification performance for standard classifiers and their naive Bayes versions.

3.2. Retrieval Performance

Retrieval is inherently a ranking problem for binary classes. Given a target class, a retrieval function is learned from the labeled data set of relevant and irrelevant examples. The learned retrieval function is then used to rank the testing examples. For NBSVM, the summation of discriminant functions $\sum_{\mathbf{x} \in X} f^C(\mathbf{x})$ in Equation 9 can be used as the retrieval function to produce ranking scores for class C. The higher the score, the more likely the feature set belongs to class C. For NBNN, we have defined the following retrieval function for class C.

$$r^C(X) = \frac{1}{Z} \exp\left(-\gamma \sum_j ||\mathbf{x}_j - NN^C(\mathbf{x}_j)||^2\right) \quad (10)$$

where Z is the normalization term that makes $r^C(X)$'s add up to one for all classes, γ is empirically set to the sample mean of the summation term on RHS.

We then performed the retrieval experiments for each of the 10 music genres over the 20 random partitions of data set. The Average Precision (AP) [4] was computed for each experiment as the retrieval performance

(a) NN vs. NBNN

Genre	NN	NBNN	p-value
blues	**0.773**	**0.797**	**0.037**
classical	**0.860**	**0.899**	**0.006**
country	0.625	0.624	0.887
disco	**0.479**	**0.587**	**0.000**
hiphop	**0.597**	**0.690**	**0.000**
jazz	**0.786**	**0.818**	**0.020**
metal	0.793	0.778	0.145
pop	**0.683**	**0.748**	**0.000**
reggae	**0.630**	**0.740**	**0.000**
rock	**0.497**	**0.365**	**0.000**

(b) SVM vs. NBSVM

Genre	SVM	NBSVM	p-value
blues	**0.819**	**0.835**	**0.037**
classical	**0.938**	**0.964**	**0.001**
country	0.724	0.735	0.131
disco	**0.648**	**0.723**	**0.000**
hiphop	**0.765**	**0.786**	**0.006**
jazz	**0.888**	**0.912**	**0.001**
metal	**0.901**	**0.920**	**0.000**
pop	**0.814**	**0.850**	**0.000**
reggae	0.751	0.756	0.372
rock	**0.511**	**0.548**	**0.000**

Table 2. Comparison of retrieval performance for different methods.

measure. A higher AP indicates better retrieval performance. Table 2 reports the average AP values for different methods. The bold rows indicate the genre classes for which the retrieval performance obtained by the two methods are statistically and significantly different. This is indicated by the p values returned by paired t-tests on AP values returned by NN and NBNN, as well as SVM and NBSVM. Within those genres with significantly different performances, NBSVM outperforms SVM in all cases, and NBNN also perform better than NN in all except one genres.

Figure 2 shows the mean precision-recall (PR) curves [4] of the four methods under comparison averaged over 10 genre classes, where the solid lines indicate PR curves of NBNN and NBSVM, and the broken lines indicate PR curves of NN and SVM. The mean AP value of each method is superimposed on the corresponding curve. It again confirms that the two NB classifiers outperform their non-NB standard versions in music retrieval, as indicated by higher mean AP values and larger areas under the PR curves.

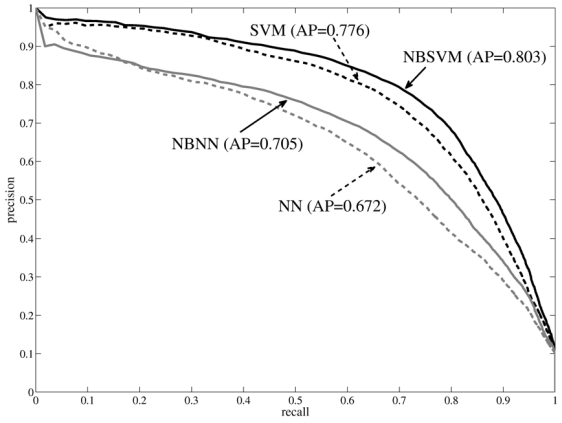

Figure 2. Mean PR curves. Horizontal and vertical axes represent recall and precision values respectively.

4. Conclusions

We have investigated the use of naive Bayes classifier for music classification and retrieval and proposed two NB classifiers, namely NBNN and NBSVM. Experimental results have shown the effectiveness of NB classifiers for both classification and retrieval. This also opens up the possibility for future research in a few directions. Different types of classifiers can be exploited within the NB framework, such as logistic regression and linear discriminant analysis. Moreover, the NB assumption specifies that local features are generated independently, which is quite restrictive in nature. It would be interesting to utilize the dependency between local features for classification and retrieval.

References

[1] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[2] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[3] M. Mandel and D. Ellis. Song-level features and svms for music classification. In *Intl. Conf. on Music Information Retrieval*, 2005.

[4] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge Uni. Press, 2008.

[5] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, 1998.

[6] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. Speech and Audio Processing*, 10(5):293–302, 2002.