2013 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2013, SOUTHAMPTON, UK

MUSIC GENRE CLASSIFICATION USING GAUSSIAN PROCESS MODELS

Konstantin Markov

Department of Information Systems
The University of Aizu
Fukushima, Japan

Tomoko Matsui

Department of Statistical Modeling
Institute of Statistical Mathematics
Tokyo, Japan

ABSTRACT

In this paper we introduce Gaussian Process (GP) models for music genre classification. Gaussian Processes are widely used for various regression and classification tasks, but there are relatively few studies where GPs are applied in the audio signal processing systems. The GP models are non-parametric discriminative classifiers similar to the well known SVMs in terms of usage. In contrast to SVMs, however, GP models produce truly probabilistic output and allow for kernel function parameters to be learned from the training data. In this work we compare the performance of GP models and SVMs as music genre classifiers using the ISMIR 2004 database. Audio preprocessing is the same for both cases and is based on Constant-Q spectrograms. The experimental results using linear as well as exponential kernel functions and different amounts of training data show that GP models always outperform SVMs with up to 5.6% absolute difference in the classification accuracy.

Index Terms— Music Genre Classification, Gaussian Process, SVM, Machine Learning

1. INTRODUCTION

A lot of music data has become available recently either locally or over the Internet and in order for users to benefit from them, an efficient music information retrieval technology is necessary. It consists of various tasks such as genre classification, artist identification, music mood classification, cover song identification, fundamental frequency estimation, melody extraction, etc. Each classification system consists of minimum two blocks: feature extractor and classifier. Studies in music genre classification have investigated various feature types and their extraction algorithms [1, 2, 3]. Carefully crafted music features such as chroma vectors are mostly used for specific tasks like music transcription or music scene analysis [4]. On the other hand, when it comes to classifying music patterns, spectrum and its derivatives are also widely used.

Various methods for building music genre classifiers have been studied ranging from conventional SVM to compressive sampling models [5]. Learning algorithms include instances of supervised, semi-supervised [6], and unsupervised [7] methods. However, parametric models are dominant in most of the studies.

Gaussian Processes have been known as non-parametric Bayesian models for quite some time, but just recently have attracted attention of researchers from fields other than statistics and machine learning [8]. One possible reason is the fact that several extensions and new models based on GPs have been developed lately. For example, the Gaussian Process latent variable models (GP-LVM) were introduced for non-linear dimensionality reduction [9], but have been also applied for image reconstruction [10] and human motion modeling [11]. Another promising extension is the Gaussian Process Dynamic Model (GPDM) [12]. It is a non-linear dynamical system which can learn the mapping between two continuous variables spaces. One of the first applications of GPDM in audio signal processing was for speech phoneme classification [13]. Although the absolute classification accuracy of the GPDM was not high, in certain conditions they outperformed the conventional hidden Markov model (HMM). In another recent work, GPDM is used as model for non-parametric speech representation and speech synthesis [14]. Similar to GPDM is the GP based state-space model [15, 16]. It is essentially a non-linear Kalman filter and is very useful for time series processing. Compared to some approximate Gaussian filters such as the Extended Kalman filter (EKF) and the Unscented Kalman filter (UKL), it gives exact expected values in the prediction and filter steps. When applied for non-linear regression tasks, the Gaussian Processes allow an analytic solution to be obtained for the output value distribution. This can be used for voice activity detection and speech enhancement in the time domain [17]. For the GP classification models, however, approximations are needed in order to obtain class label probabilities [18, 19]. While GP have been used in computer vision, for example, for object categorization [20], we are unaware of any prior work on music genre classification. In our system we use the GP models as discriminative binary classifiers in a setting very similar to how conventional SVM are used in such multi-class tasks. We compare the performance of GP classification models with SVMs in one-versus-all training mode using several different

978-1-4799-1180-6/13/$31.00

amounts of training data.

2. GAUSSIAN PROCESS MODELS

By definition, the Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution [8]. It is completely specified by its mean function and covariance function. Given a real process $f(\boldsymbol{x})$, the mean function $m(\boldsymbol{x})$ and the covariance function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ are defined as

$$\begin{aligned} m(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})] \\ k(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \mathbb{E}[(f(\boldsymbol{x}_i) - m(\boldsymbol{x}))(f(\boldsymbol{x}_j) - m(\boldsymbol{x}))] \end{aligned}$$

and we can write the GP as

$$f(\boldsymbol{x}) \sim \mathcal{N}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x})). \quad (1)$$

In this case, the random variables represent the value of the function $f(\boldsymbol{x})$ at position $\boldsymbol{x}$.

Often, it is assumed that the mean is zero, i.e. $m(\boldsymbol{x}) = 0$, while the covariance function $k(\boldsymbol{x}, \boldsymbol{x})$ can be any appropriate kernel function depending on the application.

2.1. Classification with GP

For binary classification, given training data vectors $\boldsymbol{x_i} \in \mathbb{R}^d$ with corresponding labels $y_i \in \{-1, +1\}$, we would like to predict the class membership probability for a test point $\boldsymbol{x_*}$. This is done using an unconstrained latent function $f(\boldsymbol{x})$ distributed according to Eq.(1) and mapping its value into the unit interval $[0, 1]$ by means of a sigmoid shaped function [19]. Common choice for such function is the logistic function or the cumulative density function of a standard Gaussian distribution Φ. When the sigmoid is point symmetric, the likelihood $p(y|\boldsymbol{x})$ can be written as $\mathrm{sig}(y \cdot f(\boldsymbol{x}))$.

Let $\boldsymbol{X} = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}]$ be the training data matrix, $\boldsymbol{y} = [y_1, \ldots, y_n]^T$ be the vector of target values, and $\boldsymbol{f} = [f_1, \ldots, f_n]^T$ with $f_i = f(\boldsymbol{x_i})$ be the vector of latent function values. Given the latent function, the class labels are assumed independent Bernoulli variables and therefore the likelihood can be factorized as

$$p(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{n} p(y_i|f_i) = \prod_{i=1}^{n} \mathrm{sig}(y_i f_i) \quad (2)$$

Since our latent functions represent GP, their joint distribution $p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{K})$ is Gaussian with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{K}$ with elements $k(\boldsymbol{x_i}, \boldsymbol{x_j})$. Using the Bayes' rule, we can express the posterior distribution over the latent values as

$$\begin{aligned} p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{X}) &= \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{X})}{\int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{X})d\boldsymbol{f}} \\ &= \frac{\mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{K})}{p(\boldsymbol{y}|\boldsymbol{X})} \prod_{i=1}^{n} \mathrm{sig}(y_i f_i) \end{aligned} \quad (3)$$

Unfortunately, both the likelihood $p(\boldsymbol{y}|\boldsymbol{f})$ and the marginal $p(\boldsymbol{y}|\boldsymbol{X})$ are non-Gaussian and analytic calculation is impossible. Approximations in this case are either based on a Gaussian approximation to the posterior or Markov Chain Monte Carlo (MCMC) sampling.

For a test vector $\boldsymbol{x}_*$, we first find the predictive distribution for the corresponding latent variable f_* by marginalizing over the training set latent variables

$$p(f_*|\boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X}) = \int p(f_*|\boldsymbol{f}, \boldsymbol{x}_*, \boldsymbol{X}) p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{X}) d\boldsymbol{f} \quad (4)$$

where the conditional prior

$$\begin{aligned} p(f_*|\boldsymbol{f}, \boldsymbol{x}_*, \boldsymbol{X}) = \\ \mathcal{N}(f_*|\boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{f}, k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \boldsymbol{k}_*) \end{aligned} \quad (5)$$

is Gaussian and $\boldsymbol{k}_* = [k(\boldsymbol{x}_1, \boldsymbol{x}_*), \ldots, k(\boldsymbol{x}_n, \boldsymbol{x}_*)]^T$.

Finally, the predictive class membership probability is obtained by averaging out the test latent variable

$$\begin{aligned} p(y_*|\boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X}) &= \int p(y_*|f_*) p(f_*|\boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X}) df_x \\ &= \int \mathrm{sig}(y_* f_*) p(f_*|\boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X}) df_x \end{aligned} \quad (6)$$

A Gaussian approximation to the posterior of Eq.(3), $q(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\bar{\boldsymbol{f}}, \boldsymbol{A})$ gives rise to an approximate predictive distribution for test data, i.e. $q(f_*|\boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X}) = \mathcal{N}(f_*|\mu_*, \sigma_*^2)$, with mean and variance

$$\begin{aligned} \mu_* &= \boldsymbol{k}_*^T \boldsymbol{K}^{-1} \bar{\boldsymbol{f}} \\ \sigma_*^2 &= k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^T (\boldsymbol{K}^{-1} - \boldsymbol{K}^{-1} \boldsymbol{A} \boldsymbol{K}^{-1}) \boldsymbol{k}_* \end{aligned} \quad (7)$$

When the cumulative Gaussian density function Φ is used as a likelihood function, the approximate probability of $\boldsymbol{x}_*$ having label $y_* = +1$ can be calculated analytically

$$\begin{aligned} q(y_* = +1|\boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X}) &= \int \Phi(f_*) \mathcal{N}(f_*|\mu_*, \sigma_*^2) df_* \\ &= \Phi\left(\frac{\mu_*}{\sqrt{1 + \sigma_*^2}}\right) \end{aligned} \quad (8)$$

The parameters $\bar{\boldsymbol{f}}$ and $\boldsymbol{A}$ of the posterior approximation can be found using either the Laplace's method or the Expectation Propagation (EP) algorithm [18].

2.2. Hyper-Parameter Learning

Until now, we have considered fixed covariance function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, but in general, it is parameterised by some parameter vector $\boldsymbol{\theta}$. This introduces *hyper-parameters* to the GP, which are unknown and, in practice, very little information about them is available. A Bayesian approach to their estimation would require a *hyper-prior* $p(\boldsymbol{\theta})$ and the evaluation of the following posterior

$$p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\boldsymbol{y}|\boldsymbol{X})} = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}} \quad (9)$$

where the likelihood $p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta})$ is actually the GP marginal likelihood (evidence)

$$p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{\theta})d\boldsymbol{f} \tag{10}$$

However, the evaluation of the integral in Eq.(9) can be difficult and as an approximation we may directly maximize Eq.(10) w.r.t the hyper-parameters $\boldsymbol{\theta}$. This is known as maximum likelihood II (ML-II) type hyper-parameter estimation and requires estimating the GP marginal likelihood. Again, Laplace or EP approximation can be used. For the maximization, good candidates are gradient based methods such as the conjugate gradient optimization or the BFGS algorithm.

2.3. Relation to SVM

For the soft margin support vector machine, the optimization problem is defined as

$$\begin{aligned} &\min_{\boldsymbol{w},w_0} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(1-y_i f_i) \\ &\text{s.t. } 1-y_i f_i \geq 0,\ i=1,\ldots,n \end{aligned} \tag{11}$$

where $f_i = f(\boldsymbol{x}_i) = \boldsymbol{w}\boldsymbol{x}_i + w_0$ and the solution has the form $\boldsymbol{w} = \sum_i \lambda_i y_i \boldsymbol{x}_i = \sum_i \alpha_i \boldsymbol{x}_i$. Thus, the square norm of $\boldsymbol{w}$ becomes

$$\|\boldsymbol{w}\|^2 = \sum_{i,j} \alpha_i \alpha_j \boldsymbol{x}_i \boldsymbol{x}_j \tag{12}$$

which in matrix form and using kernel $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ instead of $\boldsymbol{x}_i\boldsymbol{x}_j$ is

$$\|\boldsymbol{w}\|^2 = \boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{f}^T\boldsymbol{K}^{-1}\boldsymbol{f} \tag{13}$$

where $\boldsymbol{f} = \boldsymbol{K}\boldsymbol{\alpha}$. Then, substituting $\|\boldsymbol{w}\|^2$ in Eq.(11) we get the following objective function

$$\begin{aligned} &\frac{1}{2}\boldsymbol{f}^T\boldsymbol{K}^{-1}\boldsymbol{f} + C\sum_{i=1}^{n}(1-y_i f_i) \\ &\text{s.t. } 1-y_i f_i \geq 0,\ i=1,\ldots,n \end{aligned} \tag{14}$$

On the other hand, in the GP classification, during the posterior approximation we need to find the maximum a posteriori value $\bar{\boldsymbol{f}}$ of $p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X})$ by maximizing the $\log p(\boldsymbol{y}|\boldsymbol{f}) + \log p(\boldsymbol{f}|\boldsymbol{X})$ which becomes

$$\log p(\boldsymbol{y}|\boldsymbol{f}) - \frac{1}{2}\boldsymbol{f}^T\boldsymbol{K}^{-1}\boldsymbol{f} - \frac{1}{2}\log|\boldsymbol{K}| - \frac{n}{2}\log 2\pi \tag{15}$$

when using zero mean GP prior $\mathcal{N}(\boldsymbol{f}|0,\boldsymbol{K})$. Since the last two terms are constant when the kernel is fixed, it is equivalent to minimizing the following quantity

$$\frac{1}{2}\boldsymbol{f}^T\boldsymbol{K}^{-1}\boldsymbol{f} - \sum_{i=1}^{n}\log p(y_i|f_i) \tag{16}$$

Apparently, there is a strong similarity between the SVM optimization problem and the MAP maximization of the GP classifier. Thus, there is a close correspondence between their solutions [8].

One big advantage of the GP classifier is that the output it produces - the prediction for $p(y=+1|\boldsymbol{x})$, is clearly probabilistic. Furthermore, it provides a measure of uncertainty for this prediction, i.e. the predictive variance of $f(\boldsymbol{x})$. Although, it is possible to give probabilistic interpretation to the SVM outputs by wrapping them with sigmoid function, this is rather *ad hoc* procedure which also requires tuning of the sigmoid parameters [21].

3. EXPERIMENTS

3.1. Database and Feature Extraction

In these experiments we used the ISMIR 2004 audio corpus [22]. It contains of 729 whole tracks for training, but since the number of tracks per genre is non-uniform, the original nine genres are usually mapped into the following six classes: Classical, Electronic, Jazz-Blues, Metal-Punk, Rock-Pop and World. Another 729 tracks are used for testing.

All audio data are divided into 5 sec. pieces which were further randomly selected in order to make several training sets with different amount of data, keeping the same number of such pieces per genre. Table 1 summarizes the contents of the training data sets. For example, IS-20 is a data set from the training part of the ISMIR 2004 corpus consisting of 20 pieces per genre or 120 pieces in total. All sets are constructed in such way that each larger set contains all the pieces from the smaller set. There is only one test set and it consists of 250 pieces per genre randomly selected from the ISMIR 2004 test tracks.

When it comes to feature extraction for music information processing, in contrast to the case of speech, where the MFCC is dominant, there exists wide variety of approaches - from carefully crafted multiple music specific tonal, chroma, etc. features to plain and simple "don't care about the content" spectrum. In our experiments, we used spectral representation tailored for music signals, such as Constant-Q transformed (CQT) FFT spectrum. The CQT can be thought of as a series of logarithmically spaced filters having constant center frequency to bandwidth ratio, i.e.

$$\frac{f_k}{\Delta f_k} = Q \tag{17}$$

where Q is known as the transform's "quality factor". The main property of this transform is the log-like frequency scale where the consecutive musical notes are linearly spaced [23].

The CQT transform is applied to the FFT spectrum vectors computed from 23.2ms (512 samples) frames with 50% overlap in a way that there are 12 Constant-Q filters per octave resulting in a filter-bank of 89 filters which covers the whole bandwidth of 11025 Hz. The filter-bank outputs of 20 consecutive frames are further stacked into a 1780 (89x20)

dimensional super-vector which is used in the experiments. This is the same as to have a 20 frame time-frequency spectrum image. There is a overlap of 10 frames between such two consecutive spectrum images. This way, each 5 sec. music piece is represented by 41 spectrum images or super-vectors.

Table 1. Data sets used in the experiments.

Data set	5 sec. pieces	Total time (h)
IS-20	6 x 20	0.17
IS-50	6 x 50	0.42
IS-100	6 x 100	0.83

3.2. SVM Baseline

As a baseline classifier we use the conventional SVM. For each genre and each training data set we trained single SVM model in one-versus-all multi-class setting. Input vectors were scaled to fit the [0,1] range and the SVMs were trained to produce probabilistic outputs. For each 5 sec. test sample, logarithms of the SVM output for each of the 41 vectors were aggregated and used as score for classification. Table 2 shows the classification accuracy using both Linear and RBF kernel for each training data set. Clearly, for this case, the RBF kernel gives better performance.

Table 2. SVM Classification accuracy (%).

SVM kernel	Training data set		
	IS-20	IS-50	IS-100
Linear	48.7	53.5	53.8
RBF	55.7	62.4	64.0

3.3. GP Evaluation

For the experiments with Gaussian Process classifiers we used the GPML Toolbox package[1]. It provides wide variety of covariance and mean functions, several inference methods and likelihood functions. After some preliminary experimentation, we found that in terms of speed and performance the combination of logistic likelihood function and Laplace based Gaussian approximation inference method gives the best results.

As GP mean function we used zero mean since choosing any other available function either did not improve the performance or caused stability problems. The main factors which influence the GP performance are the form of the covariance function and the values of its parameters. We found that most suitable are the following covariance functions:

[1]http://www.gaussianprocess.org/gpml/code/matlab/doc/index.html

- Linear with parameter l

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)/l^2 \quad (18)$$

- Squared exponential with parameters σ and l

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{1}{2l^2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')) \quad (19)$$

- Rational quadratic with parameters σ, α and l

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2(1 + \frac{1}{2\alpha l^2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}'))^{-\alpha} \quad (20)$$

- Matérn with parameters σ and l

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= \sigma^2(1 + r)\exp(-r), \quad (21) \\ r &= \sqrt{\frac{3}{l^2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')} \end{aligned}$$

Training of the GP classification models, as explained in Sec.2.2, consists in estimating the covariance (and mean) function parameters. Since like SVMs the GPs are binary discriminative classifiers, training set-up is the same - one GP model per genre per dataset trained in one-versus-all mode.

In the case of GP classification, there are several ways of computing test samples scores. As with the SVM, the score of single 5 sec. test sample is an aggregation of the GP outputs for each of the 41 feature vectors. However, the GP outputs not only the probability $q(y_* = +1|\boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X})$ of input vector $\boldsymbol{x}_*$ having label $y_* = +1$, but the mean μ_* and variance σ_*^2 of y_* as well. We found that using means μ_* as scores, in average, gives better results than using log probabilities.

The GP classification performance for each of the above mentioned covariance functions is summarized in Table 3. As in the case of SVM, the linear covariance function is much worse than the non-linear ones. Among the non-linear covariances, the Exponential one seems to be slightly better than the others.

Table 3. Gaussian Process Classification accuracy (%).

Covariance kernel	Training data set		
	IS-20	IS-50	IS-100
Linear	50.5	54.2	54.6
Exponential	61.3	65.7	67.7
Rational	61.2	65.3	67.7
Matérn	60.9	65.0	67.6

3.4. SVM and GP model comparison

The way we use SVMs and GP models as binary classifiers is very similar. In addition, the performance of both of them is

greatly influenced by the choice of the kernel function. However, the GP models have the advantage of producing truly probabilistic outputs and ability to use prior with parameters learned from the data. This is most probably the reason for their superior performance. In Fig.1 and Fig.2 we compare the SVM and GP models results using the same type of kernels: Linear and Exponential. As can be seen, in both cases GP models are better and the performance gap is bigger when the Exponential kernel is used.

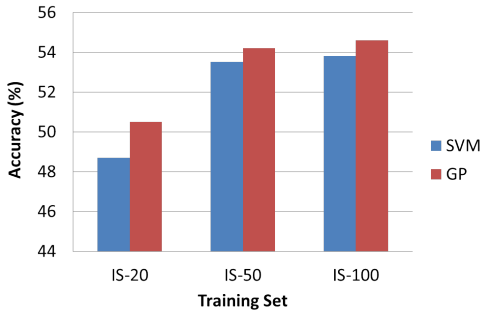

Fig. 1. SVM versus GP classification performance using Linear kernels.

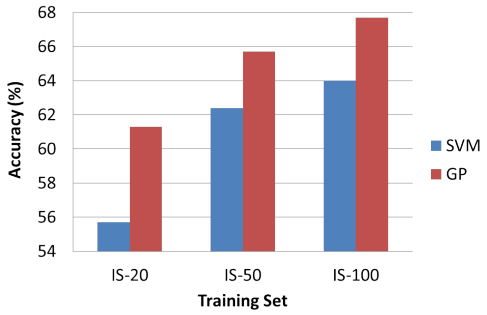

Fig. 2. SVM versus GP classification performance using Exponential kernels.

4. CONCLUSION

In this paper, we presented and described a music genre classification system where each music genre class is represented by a non-parametric Gaussian Process classification model. The implementation of the GPs for classification is similar to that of SVMs since they are too discriminative binary classifiers. Thus, we used an SVM based system as a baseline for performance comparison.

The evaluation experiments carried out using the ISMIR 2004 music database showed that GP models outperform the SVM when the same class of kernels functions are used, i.e. Linear or Exponential. This can be due to the fact that GP models output true probabilities and that covariance kernel function parameters can be learned from the training data.

GPs are not only good static classifiers, but also can be extended to model and discriminate temporal sequences, to represent non-linear mappings between two continuous spaces as well as to study non-linear dynamical systems. This gives opportunities for GPs to be used more widely in music and speech processing research fields.

5. REFERENCES

[1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. ASLP*, vol. 10, no. 5, pp. 293–302, 2002.

[2] R. Typke, F. Wiering, and R. C. Veltkamp, "A survey of music information retrieval systems," in *Proc. International Conference on Music Information Retrieval*, 2005, pp. 153–160.

[3] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[4] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*, Springer, New York, 2006.

[5] K. Chang, J.-S. Jang, and C. Iliopoulos, "Music genre classification via compressive sampling," in *Proc. ISMIR*, 2010, pp. 387–392.

[6] K. Markov and T. Matsui, "High level feature extraction for the self-taught learning algorithm," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, pp. 6, 2013.

[7] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in *Proc. ISMIR*, 2011.

[8] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning. The MIT Press, 2006.

[9] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *The Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[10] M. Titsias and N. Lawrence, "Bayesian gaussian process latent variable model," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.

[11] N. Lawrence and A. Moore, "Hierarchical gaussian process latent variable models," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 481–488.

[12] J. Wang, D. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, 2008.

[13] H. Park, S. Yun, S. Park, J. Kim, and C. Yoo, "Phoneme classification using constrained variational gaussian process dynamical system," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 2015–2023.

[14] G. Henter, M. Frean, and W. Kleijn, "Gaussian process dynamical models for nonparametric speech representation and synthesis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4505–4508.

[15] M. Deisenroth, M. Huber, and U. Hanebeck, "Analytic moment-based gaussian process filtering," in *Proc. 26th Annual International Conference on Machine Learning*, 2009, ICML '09, pp. 225–232.

[16] R. Turner, M. Deisenroth, and C. Rasmussen, "State-space inference and learning with gaussian processes," in *Proc. 13th Internatioanl Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 868–875.

[17] S. Park and S. Choi, "Gaussian process regression for voice activity detection and speech enhancement," in *Proc. IEEE International Joint Conference on Neural Networks (IJCNN)*, 2008, pp. 2879–2882.

[18] M. Kuss and C. Rasmussen, "Assessing appropriate inference for binary Gaussian Process classification," *Journal of Machine Learning Research*, vol. 6, pp. 1679–1704, 2005.

[19] H. Nickisch and C. Rasmussen, "Approximations for binary Gaussian Process classification," *Journal of Machine Learning Research*, pp. 2035–2078, 2008.

[20] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell, "Gaussian processes for object categorization," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 169–188, 2010.

[21] J. Platt, "Probabilities for SV Machines," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, Eds., pp. 61–74. MIT Press, 2000.

[22] P. Cano, E. Gomes, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack, "ISMIR 2004 audio description contest," Tech. Rep. MTG-TR-2006-02, Universitat Pompeu Fabra, 2006.

[23] C. Schoerkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *Proc. 7th. Sound and Music Computing Conference*, 2010.