Chapter 3
Speech and Music Emotion Recognition Using Gaussian Processes

Konstantin Markov and Tomoko Matsui

Abstract Gaussian Processes (GPs) are Bayesian nonparametric models that are becoming more and more popular for their superior capabilities to capture highly nonlinear data relationships in various tasks ranging from classical regression and classification to dimension reduction, novelty detection and time series analysis. Here, we introduce Gaussian processes for the task of human emotions recognition from emotionally colored speech as well as estimation of emotions induced by listening to a piece of music. In both cases, first, specific features are extracted from the audio signal, and then corresponding GP-based models are learned. We consider both static and dynamic emotion recognition tasks, where the goal is to predict emotions as points in the emotional space or their time trajectory, respectively. Compared to the current state-of-the-art modeling approaches, in most cases, GPs show better performance.

3.1 Introduction

Emotions play an important role in human-to-human communication. Expressed both by speech and body language, they convey a lot of nonlinguistic information making human interaction inherently "natural." That is why it is important to study and model emotions in order to achieve as natural as possible human–computer communication. The first and foremost task is to accurately identify the emotional state of a person. This would benefit current speech recognition and translation systems, facilitate development of new human centric applications, and also help diagnose and prevent mental health disorders such as depression which exhibit specific emotional patterns.

K. Markov (✉)
The University of Aizu, Fukushima, Japan
e-mail: markov@u-aizu.ac.jp

T. Matsui
The Institute of Statistical Mathematics, Tokyo, Japan
e-mail: tmatsui@ism.ac.jp

G.W. Peters and T. Matsui (eds.), *Modern Methodology and Applications in Spatial-Temporal Modeling*, JSS Research Series in Statistics,
DOI 10.1007/978-4-431-55339-7_3

On the other hand, a lot of music data have become available recently either locally or over the Internet and in order for users to benefit from them, an efficient music information retrieval (MIR) technology is necessary. Although users are more likely to use genres or artists names when searching or categorizing music, the main power of music is in its ability to communicate and trigger emotions in listeners. Thus, determining computationally the emotional content of music is also an important task.

There are two approaches to represent emotions in computer systems: categorical and dimensional [3, 24]. Categorical approach involves finding emotional descriptors, usually adjectives, which can be arranged into groups. Given the perceptual nature of human emotion, it is difficult to come up with an intuitive and coherent set of adjectives and their specific grouping. Depending on the research objectives, the number of emotion categories and their names can vary greatly. A popular choice is the set of so-called "primary" emotions [6] which includes joy, sadness, fear, anger, surprise, and disgust. Other emotions can be produced by "mixing" primary emotions like colors in a color palette. To alleviate the challenge of ensuring consistent interpretation of emotion categories, some studies propose to describe emotions using continuous multidimensional metrics defined on low-dimensional spaces. Most widely accepted is the two-dimensional *Valence–Arousal* (V–A) affect space [45, 48] where emotions are represented by points in the V–A plane. Figure 3.1 shows the space where some regions are associated with distinct emotion categories. An extension to three-dimensional affect space which includes additional *Dominance* (D) axes has also been proposed [14]. It can be argued that emotions are not necessarily constant, but can vary within utterances or during the course of a song. This variation in time can be represented by a trajectory in the emotional space. Here, we assume that in the case of static emotions, the task is to automatically find the point in the V–A or V–A–D space which corresponds to the speaker affect state or emotion induced by a given music piece. For dynamic emotions, the task would be to estimate or track the emotion trajectory in the affect space.

An important problem in emotion recognition is how to extract features that efficiently and compactly characterize different emotions. One aspect of this problem is the analysis window used for feature extraction. A standard approach in audio signal

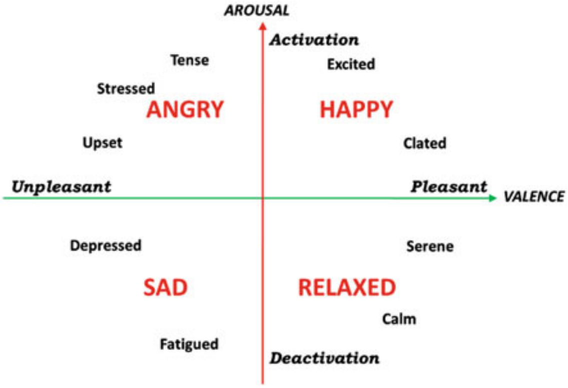


Fig. 3.1 Two-dimensional (Valence-Arousal) affective space of emotions. Different regions correspond to different categorical emotions

processing is to divide the signal into small intervals called frames from which local feature vectors are extracted. This is justified for quickly changing targets. Emotions, however, vary slowly and the analysis interval may be as long as few seconds. Common approach is to obtain some statistics such as mean, variance, etc. of the local features for each interval and stack them into one vector. This technique is well suited for the case of dynamic emotion recognition. For the static emotions case, analysis interval is usually extended to cover the whole utterance or song and global statistics are calculated.

There is a strong evidence that prosodic features such as pitch and energy are closely related to the emotional content of an utterance. Overall energy and its distribution across frequencies as well as duration of pauses are directly affected by the arousal state of the speaker [5]. Spectral-based features commonly used in speech recognition, i.e., MFCC and LPCC, have also shown good performance, though the log frequency power coefficients (LFPC) have been found to perform better [40]. When data from other modalities such as video are available, features extracted from facial expressions can be combined with the acoustic features which may lead to an improved recognition accuracy [22].

Prior studies focused on searching for emotion-specific music features have not found any dominant single one [64], so the most commonly used are those utilized in the other MIR tasks as well. Conventional features can be divided into "low-level" features including timbre (zero-crossing rate, spectral centroid, flux, roll-off, MFCC, and others) and temporal (amplitude modulation or autoregressive coefficients) features, as well as "mid-level" features, such as rhythm, pitch, and harmony [16]. On the other hand, it is also possible to apply unsupervised learning methods to find some "high level" representations of the "low-level" features, and then use them as a new type of features. This can be accomplished using non-negative matrix factorization (NMF), sparse coding [34], or deep neural networks (DNN) [29].

For categorical emotions, both speech and music emotion recognition tasks can be cast as a classification problem, so the same models can be used. This holds for the dimensional emotions as well since the task is actually a regression problem. Hidden markov models (HMM), Gaussian mixture models (GMM), support vector machine (SVM), and neural networks have been used to classify emotions [12]. Regression models, such as multiple linear regression (MLR), support vector regression (SVR), or Adaboost.RT, as well as multi-level least-squares or regression trees [3] have been successfully applied for dimensional emotion estimation. Model learning is usually supervised and requires labeled training data. Finding consistent emotion labels in terms of V–A or V–S–D values is even more challenging than obtaining category labels because emotion interpretation can be very subjective and varies among listeners. It requires data annotation by multiple experts which are expensive, time consuming, and labor intensive [1]. Especially, problematic is the collection of ground truth labels for time-continuous emotions, because the reaction lag of evaluators also needs to be taken into account [33].

Gaussian processes have been known as nonparametric Bayesian models for quite some time, but just recently have attracted attention of researchers from other fields than statistics and applied mathematics. After the work of Rasmussen and

Williams [44] which introduced GPs for the machine learning tasks of classification and regression, many researchers have utilized GPs in various practical applications. As SVMs, they are also based on kernel functions and Gram matrices, and can be used as their plug-in replacement. The advantage of GPs with respect to SVMs is that their predictions are truly probabilistic and that they provide a measure of the output uncertainty. Another big plus is the availability of algorithms for their hyperparameter learning. The downside is that the GP training complexity is $\mathcal{O}(n^3)$, which makes them difficult to use in large-scale tasks. Several sparse approximation methods have been proposed [7, 53], but this problem has not yet been fully solved and is a topic of an ongoing research.

Evaluation of the emotion recognition systems is usually performed in terms classification accuracy for categorical emotions. In the case of dimensional emotions, Pearson correlation coefficient and/or root-mean-squared error measures (RMSE) are used and often applied for each affect dimension separately. However, recently there have been discussions about the usefulness of the correlation coefficient from practical point of view. The analysis given in [22] shows that in order to achieve high correlation, coarse trajectory estimation is enough, while close frame-wise matching of up to 90 % of the trajectory can still result in much lower correlation. There are also different opinions on how to treat cases when correlation coefficient is negative.

In the next section, various existing emotion recognition systems are reviewed and compared. Brief introduction of the Gaussian processes and their implementation in regression tasks is given in Sects. 3.3 and 3.4. GP regression models can be used for static emotion estimation in a straightforward way. During training, they learn the nonlinear mapping between the feature vectors and the corresponding affect dimensions values. Thus, separate GP models are trained for each arousal and valence (and Dominance) dimension. Dynamic emotion trajectories can be considered as a time series data, so methods from statistical time series analysis would be applicable to ensure that not only feature-emotion mapping, but also temporal evolution of emotions is taken into account. One such method is Bayesian filtering by state-space models (SSMs). It is briefly described in Sect. 3.5. A widely used SSM based on linear functions is the Kalman filter (KF) [18] which is explained in Sect. 3.6. Linearity assumptions of KF, however, are significant drawback. On the other hand, particle filters (PF) allow for nonlinear functions to be used such as GPs. Section 3.7 describes the PF basics and its implementation using Gaussian processes. How to build emotion recognition systems using GPs for both static and dynamic emotions and some evaluation results on speech and music data are presented in Sect. 3.8. The last section contains some discussion and conclusions.

3.2 Related Studies

There are many studies on speech emotion recognition and most of them take the categorical approach to emotion representation. Various types of classifiers have been used such as HMM, GMM, SVM, ANN, k-mean, and others. The most popular is a fully connected HMM using prosodic features [39, 52]. In [40], a discrete HMM with MFCC, LPCC, and LFPC vectors was used and up to 75.5 % accuracy was obtained over the set of “primary” emotions. For dimensional dynamic emotion recognition, however, there are just a few studies. This task has been facilitated by the audio-visual emotion challenge (AVEC) series of evaluations. The 2013 winner [38] uses MFCC and other spectral low-level descriptors as features and partial least-squares (PLS) regression. However, this approach fails to capture dynamics information. This problem is solved in [60] using long short-term RNN to capture the time dependencies in emotion trajectories.

In one of the earliest studies on music emotion recognition, features representing timbre, rhythm, and pitch were used in SVM-based system to classify music into 13 mood categories [30]. With 499 hand-labeled 30 s clips, an accuracy of 45 % was achieved. In 2007, music emotion classification was included in the MIR evaluation exchange (MIREX) benchmarks and the best performance of 61.5 % was again achieved using SVM classifier [56]. However, recent studies have suggested that regression approaches using continuous mood representation can perform better than categorical classifiers [63]. SVR was applied in [64] to map music clips, each represented by a single feature vector, into two-dimensional V–A space. After principal component analysis (PCA)-based feature dimensionality reduction, this system achieved R^2 scores of 0.58 and 0.28 for arousal and valence, respectively. Later, this approach was extended by representing perceived emotion of a clip as a probability distribution in the emotion plane [62]. It also is possible to combine categorical and continuous emotion representations by quantizing the V–A space and apply emotion cluster classification using SVM [51], or another regression model, trained for each cluster [11].

For dynamic emotions, one approach is to divide a piece of music into segments short enough to assume that emotion does not change within each segment, and then use standard classification techniques [32]. Another study [49] considers arousal and valence as latent states of a linear dynamical system and applies KF to recover emotion dynamics over time. However, KF is a linear system and has its limitations. There exist nonlinear SSMs such as the extended Kalman filter (EKF) and unscented Kalman filter (UKF), but they put certain constraints on the SSM state and measurement functions and often suffer from stability issues. Another approach is to consider the fact that for some time intervals, emotion depends on the past and future system inputs. This suggests that context-sensitive or recurrent models can be applied. One such model is the conditional random field (CRF), but for its direct implementation the emotion space needs to be discretized [50]. However, recently proposed CRF extension allows to overcome this drawback [20]. Another model which has gained popularity lately is the long short-term memory (LSTM) recurrent neural network. It has been successfully applied for dynamic music emotion recognition and has shown state-of-the-art performance [59, 61].

Although Gaussian processes have become popular in machine learning community and have been used in such tasks as object categorization in computer vision [23] or economics and environmental studies [46], there are still few GP applications in the field of signal processing. In one such application, GP regression model is applied to time-domain voice activity detection and speech enhancement [41]. In [31], using GP, researchers estimate speakers likability given recordings of their voices. Another recent study employs GPs for head-related transfer function (HRTF) estimation in acoustic scene analysis [26]. Finally, several extensions and new models based on GPs have been developed. For example, Gaussian process latent variable model (GP-LVM) was introduced for nonlinear dimensionality reduction [27], but have also been applied to image reconstruction [54] and human motion modeling [28]. Another promising extension is the Gaussian process dynamic model (GPDM) [58]. It is a nonlinear 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 [42]. Although the absolute classification accuracy of the GPDM was not high, in certain conditions, they outperformed the conventional hidden Markov model (HMM). In [19], GPDM is used as a model for nonparametric speech representation and speech synthesis.

Some previous studies [35–37] have shown that GPs can be a feasible alternative to SVMs both for music genre classification and static emotion recognition. For the varying emotion case, as mentioned earlier, a state-space models are well suited. A number of GP-based state-space models (GP-SSM) have been proposed recently. GP-BayesFilters [25] use GPs as nonlinear functions and derive GP particle filter, GP-EKF, and GP-UKF algorithms using Monte Carlo (MC) sampling. In [8, 9], an analytic filtering approximation algorithm is presented, but lacks an analytic approach to GP-SSM parameter learning. An attempt to derive such algorithm is done in [55] which, however, has some stability problems. A Particle Markov Chain Monte Carlo (PMCMC) training method is described in [15], but it suffers from slowly converging MC sampling techniques. The problem of training GP-based state-space models parameters can be made much easier if true observations of the latent state process are available. This way, the state dynamics parameters can be learned separately from the parameters of the measurement function. In the KF case, training of the corresponding matrices and noise variances can be done using multivariate linear regression. For the GP-SSM, similar approach is applicable. The difference is that since GP output is scalar and separate GP models have to be trained for each state or observation vector dimension. Models parameters can be obtained using GP regression model learning as explained in Sect. 3.4.

3.3 Gaussian Processes

Gaussian processes are used to describe distributions over functions. Formally, the GP is defined as a collection of random variables any finite number of which has a joint Gaussian distribution [44]. It is completely specified by its mean and covariance functions. For a real process $f(\boldsymbol{x})$, the mean function $m(\boldsymbol{x})$, and the covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ are defined as

$$\begin{aligned} m(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})] \\ k(\boldsymbol{x}, \boldsymbol{x}') &= \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))]. \end{aligned} \tag{3.1}$$

Thus, the GP can be written as

$$f(\boldsymbol{x}) \sim \mathscr{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')). \tag{3.2}$$

A GP prior over function $f(\boldsymbol{x})$ implies that for any finite number of inputs $\boldsymbol{X} = \{\boldsymbol{x}_i\} \in \mathbb{R}^d, i = 1, \ldots, n$, the vector of function values $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)]^T = [f_1, \ldots, f_n]^T$ has a multivariate Gaussian distribution

$$\boldsymbol{f} \sim \mathscr{N}(\boldsymbol{\mu}, \boldsymbol{K}) \tag{3.3}$$

where the mean $\boldsymbol{\mu}$ is often assumed to be zero. The covariance matrix $\boldsymbol{K}$ has the following form:

$$\boldsymbol{K} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ k(\boldsymbol{x}_2, \boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_2, \boldsymbol{x}_n) \\ \vdots & & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix} \tag{3.4}$$

and characterizes the correlation between different points in the process. For $k(\boldsymbol{x}, \boldsymbol{x}')$, any kernel function which produces symmetric and semi-definite covariance matrix can be used.

3.4 Gaussian Process Regression

Given input data vectors $\boldsymbol{X} = \{\boldsymbol{x}_i\}, i = 1, \ldots, n$ and their corresponding target values $\boldsymbol{y} = \{y_i\}$, in the simplest regression task, y and $\boldsymbol{x}$ are related as

$$y = f(\boldsymbol{x}) + \varepsilon \tag{3.5}$$

where the latent function $f(\boldsymbol{x})$ is unknown and ε is often assumed to be a zero mean Gaussian noise, i.e., $\varepsilon \sim \mathscr{N}(0, \sigma_n^2)$. Putting a GP prior over $f(\boldsymbol{x})$ allows us to marginalize it out, which means that we do not need to specify its form and parameters. This makes our model very flexible and powerful since $f(\boldsymbol{x})$ can be any nonlinear function of unlimited complexity.

In practice, targets y_i are assumed to be conditionally independent given f_i, so that the likelihood can be factorized as

$$p(\boldsymbol{y}|\boldsymbol{f}) = \prod_1^n p(y_i|f_i) \tag{3.6}$$

where $p(y_i|f_i) = \mathscr{N}(y_i|f_i, \sigma_n^2)$, according to our observation model Eq. (3.5). Since $\boldsymbol{f}$ has normal distribution, i.e., $\boldsymbol{f}|\boldsymbol{X} \sim \mathscr{N}(\mathbf{0}, \boldsymbol{K})$, it follows that $\boldsymbol{y}$ is also a Gaussian random vector

$$p(\boldsymbol{y}|\boldsymbol{X}) = \mathscr{N}(\boldsymbol{y}|\mathbf{0}, \boldsymbol{K} + \sigma_n^2\boldsymbol{I}). \tag{3.7}$$

Given some new (test) input $\boldsymbol{x}_*$, we can now estimate the unknown target y_* and, more importantly, its distribution. Graphically, the relationship between all involved variables can be represented as shown in Fig. 3.2. To find y_*, we first obtain the joint probability of training targets $\boldsymbol{y}$ and $f_* = f(\boldsymbol{x}_*)$, which is Gaussian

$$p(\boldsymbol{y}, f_*|\boldsymbol{x}_*, \boldsymbol{X}) = \mathscr{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{K} + \sigma_n^2\boldsymbol{I} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^T & k(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix}\right) \tag{3.8}$$

where $\boldsymbol{k}_*^T = [k(\boldsymbol{x}_1, \boldsymbol{x}_*), \ldots, k(\boldsymbol{x}_n, \boldsymbol{x}_*)]$. Then, from this distribution, it is easy to obtain the conditional $p(f_*|\boldsymbol{y}, \boldsymbol{x}_*, \boldsymbol{X})$, which is also Gaussian

$$p(f_*|\boldsymbol{y}, \boldsymbol{x}_*, \boldsymbol{X}) = \mathscr{N}(f_*|\mu_{f_*}, \sigma_{f_*}^2) \tag{3.9}$$

with mean and variance

$$\mu_{f_*} = \boldsymbol{k}_*^T(\boldsymbol{K} + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{y}, \tag{3.10}$$

$$\sigma_{f_*}^2 = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^T(\boldsymbol{K} + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{k}_* \tag{3.11}$$

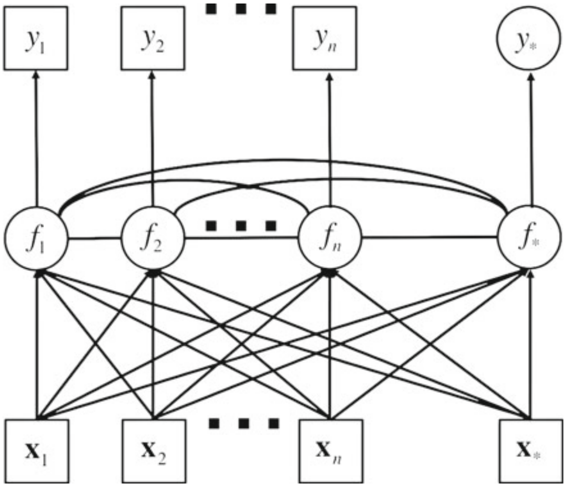


Fig. 3.2 Graphical representation of observable $\boldsymbol{x}$, y, (enclosed in *squares*), latent f, and unobservable y_* (enclosed in *circles*) variable relationships in Gaussian process-based regression task

It is worth noting that the mean μ_{f_*} is a linear combination of the observed targets $\boldsymbol{y}$. It can also be viewed as a linear combination of the kernel functions $k(\boldsymbol{x}_*, \boldsymbol{x}_i)$. On the other hand, the variance $\sigma^2_{f_*}$ depends only on inputs $\boldsymbol{X}$.

To find out the predictive distribution of y_*, we marginalize out f_*

$$\begin{aligned} p(y_*|\boldsymbol{y}, \boldsymbol{x}_*, \boldsymbol{X}) &= \int p(y_*|f_*)p(f_*|\boldsymbol{y}, \boldsymbol{x}_*, \boldsymbol{X})df_* \\ &= \mathcal{N}(y_*|\mu_{y_*}, \sigma^2_{y_*}) \end{aligned} \tag{3.12}$$

where it is easy to show that for homoscedastic likelihood, as in our case, the predictive mean and variance are [43]

$$\mu_{y_*} = \mu_{f_*}, \text{ and} \tag{3.13}$$

$$\sigma^2_{y_*} = \sigma^2_{f_*} + \sigma^2_n. \tag{3.14}$$

Making this mean our predicted target, $y_{\text{pred}} = \mu_{y_*}$ will minimize the risk for a squared loss function $(y_{\text{true}} - y_{\text{pred}})^2$. The variance $\sigma^2_{y_*}$, on the other hand, shows the model uncertainty about y_{pred}.

Parameter learning

Until now, we have considered fixed covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$, but in general, it is parameterized by some parameter vector $\boldsymbol{\theta}$. This introduces *hyper-parameters* to 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 evaluation of the following posterior:

$$p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y}|\boldsymbol{X})} = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{3.15}$$

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

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{\theta})d\boldsymbol{f}. \tag{3.16}$$

However, the evaluation of the integral in Eq. (3.15) can be difficult and as an approximation we may directly maximize Eq. (3.16) w.r.t. the hyperparameters $\boldsymbol{\theta}$. This is known as maximum likelihood II (ML-II) type hyperparameter estimation. Since both the GP prior $\boldsymbol{f}|\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K})$ and the likelihood $\boldsymbol{y}|\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma^2_n \boldsymbol{I})$ are Gaussians, the logarithm of Eq. (3.16) can be obtained analytically

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^T \boldsymbol{K}_y^{-1} \boldsymbol{y} - \frac{1}{2}\log|\boldsymbol{K}_y| - \frac{n}{2}\log 2\pi \tag{3.17}$$

where $\boldsymbol{K}_y = \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}$ is the covariance matrix of the noisy targets $\boldsymbol{y}$. Hyperparameters $\boldsymbol{\theta} = \{\sigma_n^2, \boldsymbol{\theta}_k\}$ include the noise variance and parameters of the kernel function. Those which maximize Eq. (3.17) can be found using gradient-based optimization method. Partial derivatives for each θ_i are found from

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}{\partial \theta_i} = &- \frac{1}{2}\boldsymbol{y}^T \boldsymbol{K}_y^{-1} \frac{\partial \boldsymbol{K}_y}{\partial \theta_i} \boldsymbol{K}_y^{-1} \boldsymbol{y} \\ &- \frac{1}{2}\mathtt{tr}(\boldsymbol{K}_y^{-1} \frac{\partial \boldsymbol{K}_y}{\partial \theta_i})\end{aligned} \tag{3.18}$$

where for $\theta_i = \sigma_n^2$ we have

$$\frac{\partial \boldsymbol{K}_y}{\partial \sigma_n^2} = \sigma_n^2 \boldsymbol{I}. \tag{3.19}$$

Usually, kernel function parameters are all positive, which would require constrained optimization. In practice, this problem is easily solved by optimizing with respect to the logarithm of the parameters, so simple unconstrained optimization algorithms can be used.

3.5 State-Space Models

There are many ways to define a state-space model. Here, we consider an SSM given by

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}) + \boldsymbol{u}_{t-1}, \quad \boldsymbol{x}_t \in \mathscr{R}^d, \tag{3.20}$$

$$\boldsymbol{y}_t = g(\boldsymbol{x}_t) + \boldsymbol{v}_t \qquad \boldsymbol{y}_t \in \mathscr{R}^e, \tag{3.21}$$

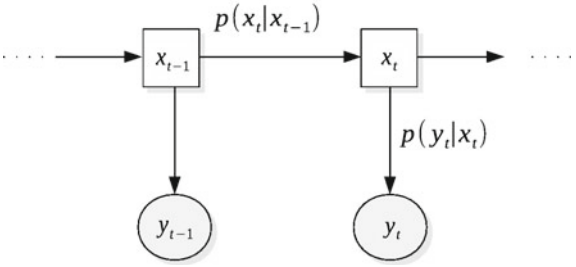
where $f()$ and $g()$ are the unknown functions governing temporal state dynamics and state-to-measurement mapping, respectively. System and observation noises, $\boldsymbol{u}_t \sim \mathscr{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_u)$ and $\boldsymbol{v}_t \sim \mathscr{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_v)$, are both Gaussian with uncorrelated dimensions. The same SSM can be written in terms of probability distributions as

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathscr{N}(\boldsymbol{x}_t; f(\boldsymbol{x}_{t-1}), \boldsymbol{\Sigma}_u), \tag{3.22}$$

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathscr{N}(\boldsymbol{y}_t; g(\boldsymbol{x}_t), \boldsymbol{\Sigma}_v). \tag{3.23}$$

Figure 3.3 shows the SSM as a graphical model with arrows denoting dependencies between variables. The initial state $\boldsymbol{x}_0$ is assumed to have known Gaussian distribution $p(\boldsymbol{x}_0) = \mathscr{N}(\boldsymbol{\mu}_0^x, \boldsymbol{\Sigma}_0^x)$. For a sequence of T measurements, the task of filtering is to find approximations to the posterior distribution $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$, where for any sequence $\{z_n\}_{n>0}$ and any $i < j$, $z_{i:j} = z_i, \ldots, z_j$. Often, the task is defined as to find the marginal distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ [2].

Fig. 3.3 Graphical representation of state-space model. States $\boldsymbol{x}_t$ are continuous latent variables and measurements $\boldsymbol{y}_t$ are observable vectors. *Arrows* show the probabilistic relationship between variables



Following a Bayesian approach, the distribution of interest can be decomposed as follows:

$$p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t})}{p(\boldsymbol{y}_{1:t})} \tag{3.24}$$

$$= \frac{p(\boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t, \boldsymbol{y}_{1:t-1})} \tag{3.25}$$

$$= p(\boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1})\frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \tag{3.26}$$

where

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)d\boldsymbol{x}_{t-1:t}. \tag{3.27}$$

This allows $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$ to be obtained recursively starting from $p(\boldsymbol{x}_0|\boldsymbol{y}_0) = p(\boldsymbol{x}_0)$ and moving forward one step at a time. Similarly, for the marginal distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, we can find that

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \tag{3.28}$$

where

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}. \tag{3.29}$$

Commonly, Eqs. (3.28) and (3.29) are referred to update and prediction steps. However, most particle filtering methods do not use these steps, but numerically approximate Eq. (3.26) [10].

As a by-product of the sequential filtering distribution estimation, the marginal likelihood $p(\boldsymbol{y}_{1:t})$ can be easily obtained from

$$p(\boldsymbol{y}_{1:t}) = p(y_1)\prod_{k=2}^{t} p(\boldsymbol{y}_k|\boldsymbol{y}_{1:k-1}) \tag{3.30}$$

When we apply an SSM for continuous emotion recognition, states $\boldsymbol{x}_t$ would represent the unknown affect vector in the V–A(–D) space, and $\boldsymbol{y}_t$ would correspond to feature vectors extracted from the audio signal. When observations of the state variable are available during training, $f()$ and $g()$ can be learned independently which makes the SSM parameter estimation simpler.

3.6 Kalman Filter

As we already mentioned, when state dynamics and measurement functions are linear, such as $f(\boldsymbol{x}) = \boldsymbol{F}\boldsymbol{x}$ and $g(\boldsymbol{x}) = \boldsymbol{G}\boldsymbol{x}$ with matrix parameters $\boldsymbol{F}$ and $\boldsymbol{G}$, an analytic solution can be easily obtained [47]. It can be shown that all distributions of interest are Gaussian:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \mathscr{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_t^p, \boldsymbol{\Sigma}_t^p) \tag{3.31}$$

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \mathscr{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{3.32}$$

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \mathscr{N}(\boldsymbol{y}_t; \boldsymbol{G}\boldsymbol{\mu}_t^p, \boldsymbol{S}_t) \tag{3.33}$$

with means and covariances which can be computed from the prediction step

$$\boldsymbol{\mu}_t^p = \boldsymbol{F}\boldsymbol{\mu}_{t-1}, \tag{3.34}$$

$$\boldsymbol{\Sigma}_t^p = \boldsymbol{F}\boldsymbol{\Sigma}_{t-1}\boldsymbol{F}^T + \boldsymbol{\Sigma}_u, \tag{3.35}$$

and the update step

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_t^p + \boldsymbol{K}_t(\boldsymbol{y}_t - \boldsymbol{G}\boldsymbol{\mu}_t^p), \tag{3.36}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_t^p - \boldsymbol{K}_t\boldsymbol{S}_t\boldsymbol{K}_t^T, \tag{3.37}$$

$$\boldsymbol{S}_t = \boldsymbol{G}\boldsymbol{\Sigma}_t^p\boldsymbol{G}^T + \boldsymbol{\Sigma}_v, \tag{3.38}$$

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_t^p\boldsymbol{G}^T\boldsymbol{S}_t^{-1}. \tag{3.39}$$

This is an optimal filtering solution given that linearity assumption holds and that noises are indeed Gaussian. In practice, however, most often neither is true.

In general, when there are no ground truth observations of the latent state variables, estimation of $\boldsymbol{F}$ and $\boldsymbol{G}$ as well as the noise variances $\boldsymbol{\Sigma}_u$ and $\boldsymbol{\Sigma}_v$ can be done using likelihood maximization via expectation–maximization algorithm [18]. However, when they are available, simple multivariate linear regression can be used to obtain the necessary parameters.

3.7 Particle Filters

Using nonlinear functions for $f()$ and $g()$ would greatly increase the expressiveness of the state-space model, but introduces two problems—what kind of nonlinearity is suitable for the task at hand and how to estimate its parameters. Gaussian processes allow to eliminate the first problem and, when state observations are available, provide solution to the second.

However, filtering with SSM when $f()$ and $g()$ are described by GPs is not straightforward. There are just a few studies on this problem and no common and efficient algorithm exists yet. Here, we utilize a particle filter-based approximation similar to the one proposed in [25].

Particle filters are a class of Monte Carlo algorithms which are based on sampling methods for density function approximations. Thus, the filtering distribution of interest can be approximated by

$$p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(\boldsymbol{x}_{1:t} - \boldsymbol{x}^i_{1:t}) \tag{3.40}$$

where samples, called particles, $\boldsymbol{x}^i_{1:t}$, $i = 1, \ldots, N$ are independently drawn from the distribution. However, in practice, often it is impossible to generate samples directly from $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$. The importance sampling (IS) method solves this problem by introducing the so-called *importance distribution*, $q()$, from which samples can be easily obtained, i.e.,

$$\boldsymbol{x}^i_{1:t} \sim q(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}) \tag{3.41}$$

and then we get the approximation as

$$p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} w^i_t\delta(\boldsymbol{x}_{1:t} - \boldsymbol{x}^i_{1:t}) \tag{3.42}$$

where

$$w^i_t \propto \frac{p(\boldsymbol{x}^i_{1:t}|\boldsymbol{y}_{1:t})}{q(\boldsymbol{x}^i_{1:t}|\boldsymbol{y}_{1:t})}. \tag{3.43}$$

For sequential distribution approximation, it would be useful to have an importance density which can be factorized as

$$q(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t})q(\boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1}). \tag{3.44}$$

This way, taking into account Eq. (3.26), the weights become

$$w_t^i \propto \frac{p(\boldsymbol{x}_{1:t-1}^i|\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{1:t-1}^i, \boldsymbol{y}_{1:t})q(\boldsymbol{x}_{1:t-1}^i|\boldsymbol{y}_{1:t-1})}, \tag{3.45}$$

$$= w_{t-1}^i \frac{p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{1:t-1}^i, \boldsymbol{y}_{1:t})}. \tag{3.46}$$

Often, it is convenient to simplify the importance distribution from the denominator to $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{1:t})$ which makes it possible to keep only the current samples $\boldsymbol{x}_t^i$ instead of the whole histories $\boldsymbol{x}_{1:t}^i$. Thus, the sequential importance sampling (SIS) algorithm involves iteration of two main steps: sampling from the importance distribution, $\boldsymbol{x}_t^i \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \boldsymbol{y}_{1:t})$ and weights update according to Eq. (3.46). However, the SIS algorithm suffers from the so-called "degeneracy" problem where after several iterations, all but few or even single particle will have negligible weights. A common solution is to "resample" with replacement N samples from the $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ approximated by the pool of particles so that $Pr(\boldsymbol{x}_t^{i*} = \boldsymbol{x}_t^j) = w_t^j$ and then reset the weights to $1/N$.

In many cases, it is convenient to choose the importance distribution to be the SSM's dynamic model

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}). \tag{3.47}$$

Then, assuming that "resampling" is performed at each step, the weights become simply

$$w_t^i \propto p(\boldsymbol{y}_t|\boldsymbol{x}_t). \tag{3.48}$$

This particular particle filter setting is known as bootstrap filter [17]. In the next section, we describe the bootstrap filter algorithm when Gaussian processes are used as SSM dynamics and measurements models.

3.7.1 Particle Filter with GP

In order to implement a bootstrap filter, it is necessary to be able to sample from $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and to calculate $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$. They, according to Eqs. (3.22) and (3.23) are Gaussians so it is easy to do it. Means of these distributions are obtained from the GPs output and variances $\boldsymbol{\Sigma}_u$ and $\boldsymbol{\Sigma}_v$ are learned during GP parameter estimation (see Sect. 3.4). One feature of the GP is that its output is actually a Gaussian distribution, and therefore, the output variance will have to be added to the corresponding dimension of $\boldsymbol{\Sigma}_u$ or $\boldsymbol{\Sigma}_v$.

Algorithm 1 provides the steps of the GP particle filter. It is assumed that GP parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$ for each target dimension are already obtained.

Algorithm 1 GP Particle filter

```
Input: $N, T, \boldsymbol{y}_{1:T}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \boldsymbol{\mu}_0^x, \boldsymbol{\Sigma}_0^x$,    Output: $\hat{\boldsymbol{x}}_{1:T}$
1. for $i = 1, \ldots, N$
2.     $\boldsymbol{x}_0^i \sim \mathcal{N}(\boldsymbol{\mu}_0^x, \boldsymbol{\Sigma}_0^x)$    $\Rightarrow$ initialize particle $i$
3.     $w_0^i = 1/N$    $\Rightarrow$ initialize weight $i$
4. end
5. for $t = 1, \ldots, T$
6.     Resample particles $\boldsymbol{x}_t^i$ according to weights $w_t^i$
7.     for $i = 1, \ldots, N$
8.         $\boldsymbol{f}_t^i, \boldsymbol{\Sigma}_{x,t}^i = GP(\boldsymbol{x}_{t-1}^i | \boldsymbol{\theta}_x)$
9.         $\boldsymbol{x}_t^i \sim \mathcal{N}(\boldsymbol{f}_t^i, \boldsymbol{\Sigma}_{x,t}^i + \boldsymbol{\Sigma}_u)$    $\Rightarrow$ propagate particle $i$
10.        $\boldsymbol{g}_t^i, \boldsymbol{\Sigma}_{y,t}^i = GP(\boldsymbol{x}_t^i | \boldsymbol{\theta}_y)$
11.        $w_t^i = \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{g}_t^i, \boldsymbol{\Sigma}_{y,t}^i + \boldsymbol{\Sigma}_v)$    $\Rightarrow$ update weight $i$
12.    end
13.    $w_t^i = w_t^i / \sum_i w_t^i$    $\Rightarrow$ normalize weights
14.    $\hat{\boldsymbol{x}}_t = \sum_i w_t^i \boldsymbol{x}_t^i$    $\Rightarrow$ estimated mean of $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t})$
15. end
16. return $\hat{\boldsymbol{x}}_{1:T}$
```

The computational complexity of this algorithm is $\mathcal{O}(NT(d+e)n^2)$, where n is the number of training vectors, because for each particle at each time t algorithm evaluates GP d times in step 8 and e times in step 10.

3.8 System Evaluation

Although from a practical point of view it might be better to have an emotion recognition system which has a categorical output, i.e., recognizes emotions in terms of textual descriptors, here, we assume that the task is to estimate the V–A(–D) point or trajectory in the affect space as accurately as possible. After that, categorical emotions can be easily obtained by affect space clustering. As a performance evaluation measures, we adopt the Pearson correlation coefficient (R) and the root-mean-square error (RMSE) which are widely used in regression tasks.

For the systems implementation, where possible, we used open-source software packages such as the GPML toolbox [43] for Gaussian processes models and the EKF/UKF toolbox [21] for Kalman filtering.

As explained in Sect. 3.4, GP covariance function parameters can be estimated via optimization procedures, but the type of the covariance function as well as the mean function which can be other than zero are system parameters to be set heuristically. The most common choices for covariance function include

- Linear (Lin) with parameter l

$$k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + 1)/l^2 \tag{3.49}$$

- Squared exponential (Exp) with parameters σ and l

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp(-\frac{1}{2l^2}(\boldsymbol{x} - \boldsymbol{x}')^T (\boldsymbol{x} - \boldsymbol{x}')) \tag{3.50}$$

- Matérn (Mat) of degree 3 with parameters σ and l

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x}') &= \sigma^2 (1 + r) \exp(-r), \\ r &= \sqrt{\frac{3}{l^2}(\boldsymbol{x} - \boldsymbol{x}')^T (\boldsymbol{x} - \boldsymbol{x}')} \end{aligned} \tag{3.51}$$

As for the mean function, previous experimental studies [36] showed that constant mean may be a better choice.

3.8.1 Speech Emotion Estimation Experiments

The database used in these experiments has been released as part of the Audio/Visual Emotion Challenge and Workshop (AVEC 2014) [57]. It consists of recordings from 84 subjects. There are 100 recordings for training and as many for testing. Duration ranges from 6 to 248 s. Each recording is annotated using three affective dimensions: arousal, valence, and dominance. The AVEC 2014 database includes speech features extracted using the openSMILE toolkit [13]. The feature set consists of 32 energy and spectral related low-level descriptors (LLD) and 6 voicing related LLDs. These features are aggregated in windows of 3 s with 1 s overlap and various statistics such as mean, standard deviation, flatness, skewness, and kurtosis are calculated for each window.

Since the original feature dimension is too high, two subsets of features were used. The first one includes only the LLD means. In the second one, LLDs delta coefficients (ΔLLDs) are included as well. Table 3.1 compares the performance of two GP-based particle filter systems with the KF. Results are given as average over all affect dimensions (V–A–D) and all 100 test samples. We have to note that for considerable number of test samples, the correlation coefficient showed negative values resulting in reduced total average.[1]

[1] These results are not directly comparable with the official AVEC'2014 results because they have been computed using the absolute R value which boosts them to the 0.5–0.6 range. We, however, believe that this approach masks system errors which are the reason for negative R values.

Table 3.1 Comparison between Kalman filter and GP-based particle filters using Linear (Lin) and squared exponential (Exp) covariance functions

Feature set		KF		GP-PF (Lin)		GP-PF (Exp)	
	# dims	R	$RMSE$	R	$RMSE$	R	$RMSE$
LLD	38	0.0350	0.1598	0.1219	0.1303	0.1417	0.0850
LLD+ΔLLD	76	0.0881	0.1691	0.1631	0.1430	0.1642	0.0890

As can be expected, the GP-based particle filter systems outperform the KF significantly. They are able to better capture the complex relationship between acoustic features and emotion representation. Increased data dimension improves the correlation measure R, but also worsens to some extend the root-mean-square error.

3.8.2 Music Emotion Estimation Experiments

For the music emotion estimation experiments, the "MediaEval'2014" database [1] was used. It consists of 1744 clips (each 45 s long) taken at random locations from 1744 different songs. They belong to various genres which can be grouped into the following eight groups: Blues, Electronic, Rock, Classical, Folk, Jazz, Country, and Pop. For training, we selected randomly 500 clips making sure that they are uniformly distributed across genre groups. In a similar way, another 500 clips were selected for testing. Each clip has a static arousal and valence annotation with score on a 9-point scale. Dynamic V–A annotations at 2 Hz rate are also available.

As feature vectors we adopted the features released by the "MediaEval'2014" organizers which include loudness, roughness, hcdf, spectral flux, and zero-crossing rate calculated at the same 2 Hz rate.

Static emotion

In order to obtain a single vector representation of each clip, two level statistics of the original feature vectors were computed. First, mean and standard deviation were taken from sliding windows of 6 vectors, which corresponds to 3 s of signal. Then, same statistics were calculated from the widow level data over the whole clip. Thus, the total dimension of the feature vectors is 20.

For the static emotion estimation case, the "MediaEval'2014" evaluation procedure was followed. It includes the R^2 as well as the RMSE measures. R^2 is commonly used to describe the goodness of fit of a statistical model and is defined as

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{3.52}$$

Table 3.2 Performance comparison between GP and SVM regression-based emotion estimation systems in terms of R^2 and RMSE measures

System	Arousal		Valence		Average	
	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$
SVR (Lin)	0.6801	0.1014	0.3612	0.1002	0.5207	0.1008
SVR (Rbf)	0.6869	0.0997	0.3713	0.0996	0.5291	0.0997
GP (Lin)	0.6747	0.1013	0.3604	0.1013	0.5176	0.1013
GP (Exp)	0.6986	0.0972	0.3594	0.1002	0.5290	0.0987
GP (Mat)	0.6973	0.0969	0.3536	0.1007	0.5255	0.0988

Table 3.3 Dynamic motion emotion recognition results using Kalman filter (KF) and GP-based particle filter (GP-PF) with several different covariance functions

System	Arousal		Valence		Average	
	R	$RMSE$	R	$RMSE$	R	$RMSE$
KF	0.1309	0.2862	0.0864	0.3048	0.1087	0.2955
GP-PF (Lin)	0.2504	0.2184	0.1328	0.2863	0.1916	0.2524
GP-PF (Exp)	0.2753	0.2166	0.1361	0.2718	0.2057	0.2442
GP-PF (Mat)	0.2821	0.2215	0.1295	0.2809	0.2058	0.2512

where y_i are the reference values, $\overline{y}$ is their mean, and $\hat{y}_i$ are the corresponding estimates. R^2 takes values in the range [0, 1][2] with $R^2 = 1$ meaning a perfect data fit.

For comparison, an SVM regression-based system with linear (Lin) and RGB (Rbf) kernel functions was built using the LIBSVM toolkit [4]. The cost parameter was optimized manually using a grid search. The other parameters were set to their defaults. Table 3.2 shows the GPR and SVR results for arousal and valence separately as well as the average score.

There is negligible difference in the GPR and SVR results, especially when exponential covariance and kernel functions are used which is the best case for both models. This to some extend confirms some previous results on the same task [36], but with different features, that GPR shows same or better performance than SVR. On the other hand, selected features may be too simple reveal the full potential of the models.

Dynamic emotion

For dynamic music emotion recognition, the original feature vectors were used to learn the GP parameters for the GP-PF system with various covariance functions. For comparison, KF system was also trained. Table 3.3 summarizes the emotion estimation results using these two systems. As in the speech emotion case, the GP-PF clearly outperforms the KF in both correlation and root-mean-squared error measures.

[2] In practice, it can take values outside this range, which would indicate estimation failure.

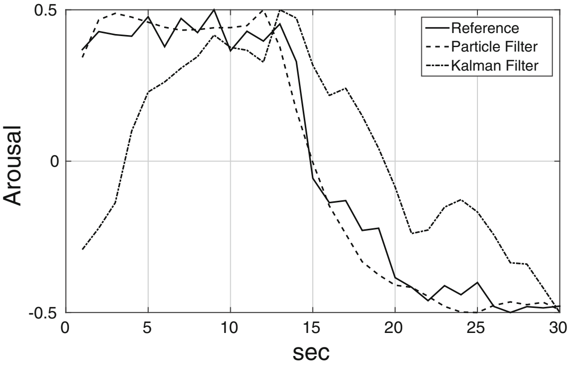


Fig. 3.4 Example of successful estimation of the arousal trajectory. The *solid curve* shows the reference arousal change and the other two are the GP particle filter and KF estimates with correlation coefficient of 0.988 and 0.698, respectively. All *curves* are scaled to fit in the [–0.5, 0.5] range

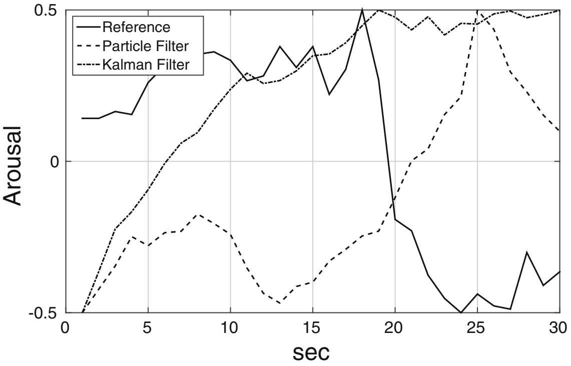


Fig. 3.5 Example of failed arousal trajectory estimation. The GP particle filter result ($R = -0.869$) exhibits opposite behavior, i.e., in contrast to the reference, at the beginning it is low and then goes up, while Kalman filter ($R = -0.460$) fails to capture the change and increases gradually

Examples of successful and failed estimation of the arousal trajectory are presented in Figs. 3.4 and 3.5, respectively. In each figure, there are three curves corresponding to the reference trajectory and the estimated trajectories from the GP particle and Kalman filters. As can be seen, even in the failed case, GP-PF was able to capture the change in the trajectory, although in the opposite direction.

3.9 Discussion and Conclusions

In this chapter, we introduced the Gaussian processes for the task of speech and music emotion recognition. For static emotion, i.e., when single point in the affect space has to be estimated for one utterance or music clip, GP regression can be used. Compared to the current state-of-the-art SVM regression, GPs perform on par or better than SVM as other studies have also shown.

The GP and SVM have many common characteristics. They are both nonparametric, kernel-based models, and their implementation and usage as regressors is very similar. However, GPs are probabilistic Bayesian predictors which in contrast to SVM produce Gaussian distributions as their output. Another GP advantage is the possibility of parameter learning from the training data. On the other hand, SVM provides a sparse solution, i.e., only “support” vectors are used for the inference, which can be a plus when working with large amount of data.

Although the same regression approach can be applied to the case of dynamic emotion recognition, capturing the characteristics of the emotion evolution in time greatly benefits the estimation performance. Thus, state-space models are well suited for such cases. The Kalman filter is a widely used linear state-space model which has been thoroughly studied and is fast and efficient model when data relationships are close to linear. When these relationships are highly nonlinear, however, the KF performance drops significantly. Nonlinear extensions, such as EKF or UKF, lessen the linearity restrictions; however, they require some prior knowledge about the form of the nonlinear functions and often suffer from stability issues.

The main advantage of Gaussian processes is that they do not require any knowledge or assumptions about the data relationships. As shown in Sect. 3.4, the mapping function $f()$ is marginalized out during the inference and can be any function with unlimited degree of nonlinearity. This leads to an improved system performance and as the above evaluations show, can be as much as two times better than the one of a linear system. Compared to other powerful nonlinear models such as Continuous Conditional Random Fields [20] or LSTM neural networks [61], the GP-based system has the advantage of being nonparametric. Thus, there is no need to choose explicit nonlinear (feature) functions as in the case of CRF or to train huge number of parameters (weights) for the NNs. Another advantage is the fully probabilistic nature of the GPs, which allows meaningful interpretation of their outputs. However, as with all nonparametric models, GPs scale poorly and for large tasks are computationally expensive.

Gaussian processes quickly penetrate many research fields and application areas which are currently dominated by the support vector machines or neural networks and show impressive performance on par or often better than the state-of-the-art approaches. Of course, there are some issues with GPs which need further improvement such as high computational complexity and storage requirements, but the current active research on GP theory will hopefully solve these problems in the near future.

References

1. Aljanaki, A., Yang, Y.H., Soleymani, M.: Emotion in music task at MediaEval 2014. In: MediaEval 2014 Workshop. Barcelona, Spain (2014)
2. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans. Sig. Process. **50**(2), 174–188 (2002)
3. Barthed, M., Fazekas, G., Sandler, M.: Multidisciplinary perspectives on musicemotion recognition: implications for content and context-based models. In: Proceedings of the 9th Symposium on Computer Music Modeling and Retrieval (CMMR), pp. 492–507 (2012)

4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27:1–27:27 (2011). http://www.csie.ntu.edu.tw/~cjlin/libsvm
5. Cowie, R., Cornelius, R.R.: Describing the emotional states that are expressed in speech. Speech Commun. **40**(1), 5–32 (2003)
6. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J.: Emotion recognition in human-computer interaction. IEEE Sig. Process. Mag. **18**(1), 32–80 (2001)
7. Csat, L., Opper, M.: Sparse on-line gaussian processes. Neural Comput. **14**(3), 641–668 (2002)
8. Deisenroth, M., Huber, M., Hanebeck, U.: Analytic moment-based gaussian process filtering. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, pp. 225–232 (2009)
9. Deisenroth, M., Turner, R., Huber, M., Hanebeck, U., Rasmussen, C.: Robust filtering and smoothing with gaussian processes. IEEE Trans. Autom. Control **57**(7), 1865–1871 (2012)
10. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: fifteen years later. Handb. nonlinear Filtering **12**, 656–704 (2009)
11. Eerola, T., Lartillot, O., Toiviainen, P.: Prediction of multidimensional emotional ratings in music from audio using multivariate regression models. In: ISMIR, pp. 621–626 (2009)
12. El Ayadi, M., Kamel, M.S., Karray, F.: Survey on speech emotion recognition: features, classification schemes, and databases. Pattern Recognit. **44**(3), 572–587 (2011)
13. Eyben, F., Wöllmer, M., Schuller, B.: Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the International Conference on Multimedia, pp. 1459–1462. ACM (2010)
14. Fontaine, J.R., Scherer, K.R., Roesch, E.B., Ellsworth, P.C.: The world of emotions is not two-dimensional. Psychol. Sci. **18**(12), 1050–1057 (2007)
15. Frigola, R., Lindsten, F., Schon, T., Rasmussen, C.: Bayesian inference and learning in gaussian process state-space models with particle MCMC. In: Advances in Neural Information Processing Systems, pp. 3156–3164 (2013)
16. Fu, Z., Lu, G., Ting, K.M., Zhang, D.: A survey of audio-based music classification and annotation. IEEE Trans. Multimedia **13**(2), 303–319 (2011)
17. Gordon, N.J., Salmond, D.J., Smith, A.F.: Novel approach to nonlinear/non-gaussian bayesian state estimation. IEEE Proc. Radar Sig. Process. **140**, 107–113 (1993)
18. Haykin, S. (ed.): Kalman Filtering and Neural Networks. Wiley (2001)
19. Henter, G., Frean, M., Kleijn, W.: Gaussian process dynamical models for nonparametric speech representation and synthesis. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4505–4508 (2012)
20. Imbrasaite, V., Baltrusaitis, T., Robinson, P.: Emotion tracking in music using continuous conditional random fields and relative feature representation. In: 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6 (2013). doi:10.1109/ICMEW.2013.6618357
21. Jouni, H., Simo, S.: Optimal filtering with kalman filters and smoothers. manual for matlab toolbox ekf/ukf. Helsinki University of Technology, Department of Biomedical Engineering and Computational Science (2008)
22. Kächele, M., Schels, M., Schwenker, F.: Inferring depression and affect from application dependent meta knowledge. In: Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge, AVEC '14, pp. 41–48. ACM (2014)
23. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Gaussian processes for object categorization. Int. J. Comput. Vis. **88**(2), 169–188 (2010)
24. Kim, E., Schmidt, E., Mingeco, R., Morton, B., Richardson, P., Scott J. Spec, J., Turnbull, D.: Music emotion recognition: a state of the art review. In: Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), pp. 255–266 (2010)
25. Ko, J., Fox, D.: GP-Bayes filters: bayesian filtering using gaussian process prediction and observation models. Auton. Robots **27**(1), 75–90 (2009)
26. Komatsu, T., Nishino, T., Peters, G., Matsui, T., Takeda, K.: Modeling head-related transfer functions via spatial-temporal gaussian process. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 301–305 (2013)

27. Lawrence, N.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. J. Mach. Learn. Res. **6**, 1783–1816 (2005)
28. Lawrence, N., Moore, A.: Hierarchical gaussian process latent variable models. In: Proceedings of the 24th International Conference on Machine Learning, pp. 481–488. ACM (2007)
29. Lee, H., Pham, P., Largman, Y., Ng, A.Y.: Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (eds.) Advances in Neural Information Processing Systems, vol. 22, pp. 1096–1104 (2009)
30. Li, T., Ogihara, M.: Detecting emotion in music. ISMIR **3**, 239–240 (2003)
31. Lu, D., Sha, F.: Predicting likability of speakers with gaussian processes. In: Proceedings of the 13th Annual Conference of the International Speech Communication Association (2012)
32. Lu, L., Liu, D., Zhang, H.J.: Automatic mood detection and tracking of music audio signals. IEEE Trans. Audio, Speech, Lang. Process. **14**(1), 5–18 (2006)
33. Mariooryad, S., Busso, C.: Correcting time-continuous emotional labels by modeling the reaction lag of evaluators. IEEE Trans. Affect. Comput. (2014). doi:10.1109/TAFFC.2014.2334294
34. Markov, K., Matsui, T.: High level feature extraction for the self-taught learning algorithm. EURASIP J. Audio, Speech, Music Process. **2013**(1), 6 (2013)
35. Markov, K., Matsui, T.: Music genre classification using gaussian process models. In: Proceedings of the IEEE Workshop on Machine Learning for Signal Processing (MLSP) (2013)
36. Markov, K., Matsui, T.: Music genre and emotion recognition using gaussian processes. IEEE Access **2**, 688–697 (2014)
37. Markov, K., Iwata, M., Matsui, T.: Music emotion recognition using gaussian processes. In: Proceedings of the ACM Multimedia 2013 Workshop on Crowdsourcing for Multimedia, CrowdMM. ACM, ACM, Barcelona, Spain (2013)
38. Meng, H., Huang, D., Wang, H., Yang, H., AI-Shuraifi, M., Wang, Y.: Depression recognition based on dynamic facial and vocal expression features using partial least square regression. In: Proceedings of the 3rd ACM International Workshop on Audio/Visual Emotion Challenge, AVEC '13, pp. 21–30. ACM (2013)
39. Nogueiras, A., Moreno, A., Bonafonte, A., Mariño, J.B.: Speech emotion recognition using hidden markov models. In: INTERSPEECH, pp. 2679–2682 (2001)
40. Nwe, T.L., Foo, S.W., De Silva, L.C.: Speech emotion recognition using hidden markov models. Speech Commun. **41**(4), 603–623 (2003)
41. Park, S., Choi, S.: Gaussian process regression for voice activity detection and speech enhancement. In: Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), pp. 2879–2882 (2008)
42. Park, H., Yun, S., Park, S., Kim, J., Yoo, C.: Phoneme classification using constrained variational gaussian process dynamical system. Adv. Neural Inf. Process. Syst. **25**, 2015–2023 (2012)
43. Rasmussen, C., Nickisch, H.: Gaussian processes for machine learning (GPML) toolbox. J. Mach. Learn. Res. **11**, 3011–3015 (2010)
44. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning. The MIT Press, Cambridge (2006)
45. Russell, J.: A circumplex model of affect. J. Pers. Soc. Psychol. **39**(6), 1161–1178 (1980)
46. Saatçi, Y., Turner, R., Rasmussen, C.: Gaussian process change point models. In: Proceedings 27th Annual International Conference on Machine Learning, pp. 927–934 (2010)
47. Särkkä, S.: Bayesian filtering and smoothing, vol. 3. Cambridge University Press (2013)
48. Scherer, K.R.: What are emotions? and how can they be measured? Soc. Sci. Inf. **44**(4), 695–729 (2005). doi:10.1177/0539018405058216
49. Schmidt, E., Kim, Y.: Prediction of time-varying musical mood distributions using kalman filtering. In: 2010 Ninth International Conference on Machine Learning and Applications (ICMLA), pp. 655–660 (2010)
50. Schmidt, E.M., Kim, Y.E.: Modeling musical emotion dynamics with conditional random fields. In: ISMIR, pp. 777–782 (2011)
51. Schmidt, E.M., Turnbull, D., Kim, Y.E.: Feature selection for content-based, time-varying musical emotion regression. In: Proceedings of the International Conference on Multimedia Information Retrieval, pp. 267–274. ACM (2010)

52. Schuller, B., Rigoll, G., Lang, M.: Hidden markov model-based speech emotion recognition. In: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03), vol. 2, pp. II–1. IEEE (2003)
53. Snelson, E., Ghahramani, Z.: Sparse gaussian processes using pseudo-inputs. In: Advances in Neural Information Processing Systems, pp. 1257–1264. MIT press, Cambridge (2006)
54. Titsias, M., Lawrence, N.: Bayesian gaussian process latent variable model. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (2010)
55. Turner, R., Deisenroth, M., Rasmussen, C.: State-space inference and learning with gaussian processes. In: Proceedings of the 13th Internatioanl Conference on Artificial Intelligence and Statistics (AISTATS), pp. 868–875 (2010)
56. Tzanetakis, G.: Marsyas submissions to mirex 2007. Music Information Retrieval Evaluation eXchange (MIREX) (2007)
57. Valstar, M., Schuller, B., Smith, K., Almaev, T., Eyben, F., Krajewski, J., Cowie, R., Pantic, M.: AVEC 2014 – 3D dimensional affect and depression recognition challenge. In: Proceedings 4th ACM International Workshop on Audio/visual Emotion Challenge (2014)
58. Wang, J., Fleet, D., Hertzmann, A.: Gaussian process dynamical models for human motion. IEEE Trans.Pattern Anal. Mach. Intell. **30**(2), 283–298 (2008)
59. Weninger, F., Eyben, F., Schuller, B.: On-line continuous-time music mood regression with deep recurrent neural networks. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5412–5416 (2014). doi:10.1109/ICASSP.2014.6854637
60. Wollmer, M., Eyben, F., Reiter, S., Schuller, B., Cox, C., Douglas-Cowie, E., Cowie, R.: Abandoning emotion classes-towards continuous emotion recognition with modelling of long-range dependencies. Proc. INTERSPEECH **2008**, 597–600 (2008)
61. Wollmer, M., Kaiser, M., Eyben, F., Schuller, B., Rigoll, G.: LSTM-modeling of continuous emotions in an audiovisual affect recognition framework. Image Vis. Comput. **31**(2), 153–163 (2013)
62. Yang, Y.H., Chen, H.: Prediction of the distribution of perceived music emotions using discrete samples. IEEE Trans. Audio, Speech, Lang. Proces. **19**(7), 2184–2196 (2011)
63. Yang, Y.H., Chen, H.: Machine recognition of music emotion: a review. ACM Trans. Intell. Syst. Technol. **3**(3), 40:1–40:30 (2012)
64. Yang, Y.H., Lin, Y.C., Su, Y.F., Chen, H.: A regression approach to music emotion recognition. IEEE Trans. Audio, Speech, Lang. Proces. **16**(2), 448–457 (2008)