Online Discriminative Kernel Density Estimation

Matej Kristan, Aleš Leonardis

Faculty of Computer and Information Science, University of Ljubljana
{matej.kristan}, {ales.leonardis}@fri.uni-lj.si

Abstract

We propose a new method for online estimation of probabilistic discriminative models. The method is based on the recently proposed online Kernel Density Estimation (oKDE) framework which produces Gaussian mixture models and allows adaptation using only a single data point at a time. The oKDE builds reconstructive models from the data, and we extend it to take into account the interclass discrimination through a new distance function between the classifiers. We arrive at an online discriminative Kernel Density Estimators (odKDE). We compare the odKDE to oKDE, batch state-of-the-art KDEs and support vector machine (SVM) on a standard database. The odKDE achieves comparable classification performance to that of best batch KDEs and SVM, while allowing online adaptation, and produces models of lower complexity than the oKDE.

1 Introduction

Building discriminative models of some process from the observed data is a central task of many applications in machine learning. A popular approach to generating models is to estimate the probability density function (pdf) associated with the observed data. In this respect, *reconstructive* models such as the Gaussian mixture models, (GMM), (e.g., [5]) have been successfully applied in *batch* operation, i.e., in situations in which all the data is observed in advance. In contrast to reconstructive models, the discriminative models capitalize on the discriminative information, however, this may lead to decreased robustness [4]. A significant drawback of the purely batch methods is that their estimation becomes increasingly difficult when processing extremely large amounts of data. Furthermore, in real-world environments, all the data may not be available in advance, or we even want to observe some process for an indefinite duration, while continually providing

the best estimate of the model from the data observed so far. This generates the need for models that can be constructed in *online* operation.

Adapting the existing reconstructive GMM methods to work with online cases, in which only a single data-point may be observed at a time, is a nontrivial task. In contrast to the batch incremental models (e.g., [11]) who store and revisit all the data in multiple passes, the online models have to adapt from a (single) new data-point and then discard that data-point. The main difficulty is therefore that the online models have to maintain sufficient information to generalize well to the yet unobserved data and have to adjust their complexity without having access to all the observations (future as well as past). There have been various attempts to extend the *reconstructive* GMMs to online operation, however, these either imply strong spatio-temporal constraints on the data [1, 14], assume constraints on the shape of the target distribution [8] or require tuning of parameters to a specific application [15]. Recently, we have proposed a non-parametric approach called the *online Kernel Density Estimation* (oKDE) [10]. In contrast to the other approaches, the oKDE does not impose any of the above constraints but assumes only that the target pdf is sufficiently smooth and produces models with a high reconstructive performance.

While the purely reconstructive models may contain redundant information required for discrimination, the discriminative models disregard the reconstructive information required for online adaptation. Indeed, Fidler et al. [4] have shown that even in batch methods accounting for the reconstructive information leads to improved robustness of the discriminative models. Following their results, we adapt the oKDE framework to account for the discriminative power of the models along with the reconstructive, thus arriving at an *online discriminative Kernel Density Estimator*, which is the main contribution of the paper. The proposed method allows online adaptation of the discriminative models, by maintaining enough reconstructive power to efficiently adapt to new observations, and can be used to

develop online classifiers. The remainder of the paper is structured as follows. In Section 2 we briefly review the oKDE framework, in Section 3 we extend the oKDE to discriminative models, in Section 4 we evaluate the approach and the Section 5 concludes the paper.

2 The online Kernel Density Estimation

The online Kernel Density Estimation (oKDE) produces a generative model from the d-dimensional streaming data as an N-component Gaussian mixture model

$$p(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi_{\boldsymbol{\Sigma}_i}(\mathbf{x} - \mathbf{x}_i), \qquad (1)$$

where $\phi_{\boldsymbol{\Sigma}}(\mathbf{x} - \mu) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} e^{(-\frac{1}{2}(\mathbf{x}-\mu)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mu))}$ is a Gaussian kernel centered at μ with covariance matrix $\boldsymbol{\Sigma}$. We give here only a brief overview of the oKDE framework and refer the reader to [10] for more details.

Broadly speaking, the oKDE proceeds in two steps:
Update: Starting with the GMM from the previous time-step $p_{t-1}(\mathbf{x})$ and the new observation $\mathbf{x}_t$, the oKDE augments the $p_{t-1}(\mathbf{x})$ by a Gaussian kernel $\phi_{\boldsymbol{\Sigma}_t}(\mathbf{x} - \mathbf{x}_t)$ centered at $\mathbf{x}_t$, it automatically calculates the optimal covariance $\boldsymbol{\Sigma}_t$ for that kernel and accordingly readjusts the covariances of the existing kernels using the online plug-in rule. If required, it also refines the GMM by splitting up some of the components.
Compress: To maintain a low complexity (i.e., low number of components) the GMM is simplified from time to time. The oKDE generates a binary tree among the components and iteratively merges pairs of components until some threshold of a cost function is exceeded. In [10], the cost function represents the distance between the distribution before and after the compression, which penalizes the reconstruction errors.

3 Discriminative KDE

We pose the online discriminative learning as a task to estimate from a stream of data a set of K discriminative classes, each class c_i described by a Gaussian mixture model $p(\mathbf{x}|c_i)$ and a prior probability $p(c_i)$. In principle, we could use the oKDE to construct each of these classes, but, due to its reconstructive nature, the produced models will be likely redundant for classification. Indeed, we require the models to contain just so much of the information to prevent a degraded classification. Recall that the oKDE simplifies the models under a certain cost function which measures the reconstruction error induced by compression [10]. This

means that by redefining this cost function to rather take into account the classification error, the compression step in the oKDE will lead to models that *reduce their complexity* while *retaining their discriminative power*.

Assume that we want to compress the c_i-th class mixture model $p(\mathbf{x}|c_i)$ into $p_{\mathrm{cmp}}(c_i|\mathbf{x})$, while minimizing the induced classification errors. First we have to rewrite this model into a *classification model*. We consider the class c_i as a *positive example* class C^+, described by a mixture model $p(C^+|\mathbf{x}) = p(c_i|\mathbf{x})$. Then we collect *all the other classes* to form a single *negative example* class C^-, $p(C^-|\mathbf{x}) = \sum_{j \backslash i} p(c_j|\mathbf{x})$. The posterior over the resulting two-class model is then defined as

$$p(C|\mathbf{x}) = \delta_{C^+}(C)p(C^+|\mathbf{x}) + \delta_{C^-}(C)p(C^-|\mathbf{x}), \quad (2)$$

where $\delta_{C^*}(C)$ is a Dirac function centered at C^*. The compressed counterpart of the posterior (2), is obtained by setting $p_{\mathrm{cmp}}(C^+|\mathbf{x}) = p_{\mathrm{cmp}}(c_i|\mathbf{x})$:

$$p_{\mathrm{cmp}}(C|\mathbf{x}) = \delta_{C^+}(C)p_{\mathrm{cmp}}(C^+|\mathbf{x}) + \delta_{C^-}(C)p(C^-|\mathbf{x}). \quad (3)$$

From the classification point of view we can say that $p(\mathbf{x}|c_i)$ can be compressed into $p_{\mathrm{cmp}}(\mathbf{x}|c_i)$ as long as the distance between the corresponding posteriors $p(C|\mathbf{x})$ and $p_{\mathrm{cmp}}(C|\mathbf{x})$, does not change significantly. We therefore require a distance measure between the posterior before and after compression.

3.1 Distance between two classifiers

We define the distance between the posterior $p(C|\mathbf{x})$ and its compression $p_{\mathrm{cmp}}(C|\mathbf{x})$, given some value of $\mathbf{x}$, using the Hellinger distance [13],

$$D^2(p, p_{\mathrm{cmp}}|\mathbf{x}) \overset{\Delta}{=} \frac{1}{2} \int_C (p(C|\mathbf{x})^{\frac{1}{2}} - p_{\mathrm{cmp}}(C|\mathbf{x})^{\frac{1}{2}})^2$$

$$= \frac{1}{2} \sum_{C \in [C^+, C^-]} (p(C|\mathbf{x})^{\frac{1}{2}} - p_{\mathrm{cmp}}(C|\mathbf{x})^{\frac{1}{2}})^2. \quad (4)$$

Integrating (4) over the relevant feature space $\mathbf{x}$ gives the *expected* Hellinger distance

$$\hat{D}^2(p, p_{\mathrm{cmp}}) = \int D^2(p, p_{\mathrm{cmp}}|\mathbf{x})p_0(\mathbf{x})d\mathbf{x}, \quad (5)$$

where the expectation is calculated over the distribution $p_0(\mathbf{x}) = p(\mathbf{x}|C^+)p(C^+) + p(\mathbf{x}|C^-)p(C^-)$, with the priors $P(C^+) = p(c_i)$ and $P(C^-) = \sum_{j \backslash i} p(c_j)$. In our case, $p_0(\mathbf{x})$ can be written in the form of a Gaussian mixture model $p_0(\mathbf{x}) = \sum_{i=1}^{M} w_i \phi_{\boldsymbol{\Sigma}_i}(\mathbf{x} - \mathbf{x}_i)$ and (5) becomes

$$\hat{D}^2(p, p_{\mathrm{cmp}}) = \sum_{i=1}^{M} w_i \int D^2(p, p_{\mathrm{cmp}}|\mathbf{x})\phi_{\boldsymbol{\Sigma}_i}(\mathbf{x} - \mathbf{x}_i)d\mathbf{x}. \quad (6)$$

Note that while $D(\cdot, \cdot | \mathbf{x})$ is a metric, constrained to the interval $[0, 1]$, it is a nonlinear function of $\mathbf{x}$, and the integrals in (6) cannot be evaluated analytically. However, they can be numerically approximated using the *unscented transform*, which has been proposed by [9] for calculating nonlinear transformations of Gaussian variables. Similarly to a Monte Carlo integration, the *unscented transform* relies on evaluating integrals using carefully placed points, called *the sigma points*, over the support of the integral. Therefore, (6) is approximated as

$$\hat{D}^2(p, p_{\text{cmp}}) \approx \sum_{i=1}^{M} w_i \sum_{j=0}^{2d+1} D^2(p, p_{\text{cmp}} |^{(j)} \mathcal{X}_i)^{(j)} \mathcal{W}_i,$$
(7)

where $\{^{(j)}\mathcal{X}_i, {}^{(j)}\mathcal{W}_i\}_{j=0:d}$ are the weighted sets of sigma points corresponding to the i-th Gaussian $\phi_{\boldsymbol{\Sigma}_i}(\mathbf{x} - \mathbf{x}_i)$, and are defined as

$$
\begin{aligned}
{}^{(0)}\mathcal{X}_i &= \mathbf{x}_i \; ; \; {}^{(0)}\mathcal{W}_i = \frac{\kappa}{1+\kappa} \\
{}^{(j)}\mathcal{X}_i &= \mathbf{x}_i + s_j \sqrt{1+\kappa} (\sqrt{d\boldsymbol{\Sigma}_i})_j \; ; \\
{}^{(j)}\mathcal{W}_i &= \frac{\kappa}{2(1+\kappa)} \; ; \; s_j = \left\{ \begin{array}{lcr} 1 & ; & j \le d \\ -1 & ; & otherwise \end{array} \right.
\end{aligned}
$$
(8)

with $\kappa = \max([0, m - d])$, and $(\sqrt{\boldsymbol{\Sigma}_i})_j$ is the j-th column of the matrix square root of $\boldsymbol{\Sigma}_i$. Specifically, let $\mathbf{U}\mathbf{D}\mathbf{U}^T$ be a singular value decomposition of covariance matrix $\boldsymbol{\Sigma}$, such that $\mathbf{U} = \{U_1, \dots, U_d\}$ and $\mathbf{D} = \text{diag}\{\lambda_1, \dots, \lambda_d\}$, then $(\sqrt{\Sigma})_k = \sqrt{\lambda_k} U_k$. In line with the discussion on the properties of the unscented transform in [9], we set the parameter m to $m = 3$.

3.2 The online discriminative KDE

The distance function $\hat{D}(p, p_{\text{cmp}})$ from Section 3.1 penalizes any change of the classification posterior which is induced through the compression of the mixture models. Small values, i.e., $\hat{D}(p, p_{\text{cmp}}) \approx 0$, mean that the classification does not change, while $\hat{D}(p, p_{\text{cmp}}) = 1$ implies a complete change. In an online operation, we can therefore simply adapt the oKDE (Section 2) to build and compress a mixture model for each class separately from the observed data. To penalize the loss of discrimination during the compression, we can use the distance measure $\hat{D}(p, p_{\text{cmp}})$ as the compression cost function. We arrive at an online discriminative KDE (odKDE). In practice, we let the odKDE compress the model until the cost function exceeds some (small) threshold $\hat{D}_{\text{th}}$.

During learning, at time-step $t - 1$, we have a set of K models $\{p_{t-1}(\mathbf{x}|c_i), p_{t-1}(c_i)\}_{i=1:K}$. For simplicity assume that at time-step t, K new observations

$\{\mathbf{z}_i\}_{i=1:K}$, one per each class[1], arrive and the models are updated into $\{p_t(\mathbf{x}|c_i), p_t(c_i)\}_{i=1:K}$. A single time-step iteration of the approach is outlined in Algorithm 1. In the classification phase a new observation $\mathbf{z}$ is classified into a class $\hat{c}$ by applying the Bayesian rule

$$\hat{c} = \arg\max_{c_i} p(\mathbf{z}|c_i)p(c_i).$$
(9)

Algorithm 1 : The online discriminative KDE

Input:
  $\{p_{t-1}(\mathbf{x}|c_i), p_{t-1}(c_i)\}_{i=1:K} \dots$ the input models.
  $\{\mathbf{z}_i\}_{i=1:K} \dots$ observations (one per each class)
Output:
  $\{p_t(\mathbf{x}|c_i), p_t(c_i)\}_{i=1:K} \dots$ the output models.
Procedure:
  1: **for** $i = 1 : K$ **do**
  2:     Update $p_{t-1}(\mathbf{x}|c_i)$ with $\mathbf{z}_i$ into $\tilde{p}_t(\mathbf{x}|c_i)$ using the original update step of oKDE (Section 2).
  3:     Update the prior $p_t(c_i)$.
  4: **end for**
  5: **for** $i = 1 : K$ **do**
  6:     Construct the two-class classification model (2) by treating $\tilde{p}_t(\mathbf{x}|c_i)$ as a positive example and the rest $K - 1$ models as the negative example.
  7:     Compress $\tilde{p}_t(\mathbf{x}|c_i)$ into $p_t(\mathbf{x}|c_i)$ by pairwise merging components such that $\hat{D}(\tilde{p}_t, p_t) \le \hat{D}_{\text{th}}$.
  8: **end for**

---

4 Experiments

We have compared the classification performance of the odKDE with the online reconstructive KDE, oKDE [10], and three state-of-the-art batch KDEs: the cross-validation (CV) KDE [12], the reduced-set density estimator [6] (RSDE) initialized by the CV, and the Hall's KDE [7] (Hall). For the baseline classification, we have applied a multiclass support vector machine (SVM) with an RBF kernel [3]. The methods were compared on a set of public classification problems [2] (Table 1). In all experiments, the distance parameter in the odKDE was set to $\hat{D}_{\text{th}} = 0.005$.

The odKDE and oKDE were initialized for each class using the first 10 samples and the rest were added one at a time; this experiment was repeated via four-fold cross validation and for three random data orderings. From results in Table 1 we see that the odKDE generated models with comparable classification performance as the oKDE, but generally with a significantly lower complexity. Although the odKDE was

---

[1]Note that, in general, we can also handle cases in which observations come only from a subset of classes at a time.

Table 1. Average classification results. The number of samples in each dataset, the dimensionality and the number of classes are denoted by N_S, N_D and N_C, respectively.

dataset	N_S	N_D	N_C	Recognition accuracy[%] (Number of components per class)					
				odKDE	oKDE	CV	RSDE	Hall	SVM
Iris	150	4	3	97%(6.5)	97%(31)	96%(38)	97%(9)	97%(38)	96%(16)
Pima	768	8	2	71%(107)	70%(162)	72%(288)	63%(15)	67%(288)	78%(162)
Wine	178	13	3	97%(2.5)	99%(45)	92%(45)	83%(12)	99%(45)	98%(22)
Letter	20000	16	26	94%(16)	95%(222)	96%(613)	55%(26)	95%(613)	96%(320)

learnt only by observing a *single* example at a time, the resulting models exhibit classification performance similar to the best batch KDE approaches and the SVM, who optimized their structure having access to *all* the data. By further inspection of the results (not shown here due to space constraints), we have observed that the odKDE models reached the upper bound on their complexity relatively early on during learning (e.g., after 200 samples for the *Letter* dataset), while their classification performance further increased through model refinement. This makes the odKDE a very appropriate tool for online operation since it produces compressed models with good classification performance, while at the same time maintains sufficient reconstructive information to allow online refinements of the models from new observations.

5 Conclusion

We have proposed an approach for online estimation of discriminative models by adapting the framework of online Kernel Density Estimation. We have defined a distance measure which measures the discrimination of models and used this measure in the oKDE as a cost function for compression. Results demonstrate that the proposed odKDE produces comparable classification performance to the state-of-the-art, and produces models of significantly lower complexity while allowing online adaptation. This makes the approach ideal for online estimation of classifiers from streaming data. In our future work we will study how different distance measures and data orderings influence the performance of the proposed method.

Acknowledgment

References

[1] O. Arandjelovic and R. Cipolla. Incremental learning of temporally-coherent gaussian mixture models. In *British Machine Vision Conference*, pages 759–768, 2005.

[2] A. Asuncion and D. Newman. UCI machine learning repository, 2007.

[3] C. C. Chang and C. J. Lin. *LIBSVM: A library for support vector machines*, 2001.

[4] S. Fidler, D. Skočaj, and A. Leonardis. Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):337–350, March 2006.

[5] M. A. F. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):381–396, 2002.

[6] M. Girolami and C. He. Probability density estimation from optimally condensed data samples. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(10):1253–1264, 2003.

[7] P. Hall, S. J. Sheater, M. C. Jones, and J. S. Marron. On optimal data-based bandwidth selection in kernel density estimation. *Biometrika*, 78(2):263–269, 1991.

[8] B. Han, D. Comaniciu, Y. Zhu, and L. S. Davis. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1186–1197, 2008.

[9] S. Julier and J. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Department of Engineering Science, University of Oxford, 1996.

[10] M. Kristan and A. Leonardis. Multivariate online kernel density estimation. In *Computer Vision Winter Workshop*, page accepted, 2010.

[11] D. Mansjur and B. Juang. Incremental learning of mixture models for simultaneous estimation of class distribution and inter-class decision boundaries. In *International Conference on Pattern Recognition*, pages 1–4, 2008.

[12] J. M. L. Murillo and A. A. Rodriguez. Algorithms for gaussian bandwidth selection in kernel density estimators. In *Neural Inf. Proc. Systems*, 2008.

[13] D. E. Pollard. *A user's guide to measure theoretic probability*. Cambridge University Press, 2002.

[14] M. Song and H. Wang. Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering. In *SPIE: Intelligent Computing: Theory and Applications*, pages 174–183, 2005.

[15] W. F. Szewczyk. Time-evolving adaptive mixtures. Technical report, National Security Agency, 2005.