Hyperkernel Based Density Estimation

Ravi S. Ganti, Nikolaos Vasiloglou, Alexander Gray
Georgia Institute Of Technology
gmravi2003@gmail.com,agray@cc.gatech.edu,nvasil@ieee.org

Abstract

We focus on solving the problem of learning an optimal smoothing kernel for the unsupervised learning problem of kernel density estimation(KDE) by using hyperkernels. The optimal kernel is the one which minimizes the regularized negative leave-one-out-log likelihood score of the train set. We demonstrate that "fixed bandwidth" and "variable bandwidth" KDE are special cases of our algorithm.

1 Introduction

Density estimation [1] is one of the most fundamental unsupervised learning problems in machine learning, where given m points $x_1, x_2, \ldots x_m \in R^n$ sampled from an unknown distribution $\mathcal{P}$ and the task is to model the distribution $\mathcal{P}$. Kernel Density Estimation (KDE) [2] is the most popular non-parametric method for density estimation. KDE involves the use of a smoothing kernel which is a p.d.f. The density estimate at a point x is then estimated as the kernel contribution of all train points at x

$$\hat{f}(x) = \frac{1}{mh} \sum_{i=1}^{m} k\left(\frac{x - x_i}{h}\right) \tag{1}$$

Gaussian,and Epanechnikov kernels are examples of such smoothing kernels. The task in KDE is to estimate the bandwidth[h]. Some common approaches to estimating h are leave-one-out-likelihood cross validation and Mean Squared Error cross validation. Though kernel density estimators demonstrate strong universal consistency [3], they are severely biased at the boundaries, if the underlying density function is of finite support. This "boundary bias" affects the problem visually and also leads to slower rate of asymptotic convergence. Techniques to correct the boundary bias include variable bandwidth KDE where h is a function of the train point, fitting a local polynomial function [4] to the empirical distribution function, and then deriving an estimator of the p.d.f by differentiation. Learning the optimal smoothing kernel by solving a variational problem that minimizes the variance or MISE of the density estimates has been done in [5], by using a representation in terms of Legendre's polynomials. While all the above methods work well in the univariate setting, most of them are difficult to generalize to a higher dimensional setting. Learning a density estimator by fitting a non-parametric mixture model to the underlying data have been explored in [6, 7].

In this work we use a hyperkernel to induce a Hilbert space of kernel functions, and optimize over this space for the kernel function that minimizes the regularized negative log likelihood score. The advantage of this method is its scalability to higher dimensions by using a multiplicative hyperkernel, and the fact that the optimization problem is convex. We explain as to how "variable bandwidth" KDE can be seen as a special case of our algorithm and empirically verify that under certain settings our optimal kernel behaves exactly like "fixed bandwidth" KDE.

The paper is organized as follows: In Section 2 we introduce the idea of hyperkernels, which is then followed by our problem formulation in Section 3. We show experimental results in Secton 4 and finish up with conclusions and future work in Section 5.

2 Hyperkernels

We will begin by introducing the notion of kernels [8]. Given data $X = \{x_1, x_2, \ldots x_n\}$ sampled from a distribution $\mathcal{P}$ and a non-empty set $\mathcal{X}$, a kernel function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$ is a symmetric positive definite function. Kernels define a map from $\mathcal{X}$ into the Hilbert space of functions mapping $\mathcal{X}$ to $\mathcal{R}$ as $\phi : x \rightarrow K(x, .)$, with the dot product between two function $\langle \phi(x), \phi(y) \rangle$ defined as $K(x, y)$. Most kernel based algorithms solve an optimization problem of the form $\hat{f} = \arg\min_{f \in \mathcal{H}} R_{\text{emp}}(f, X) + \lambda \Omega(f)$ where R_{emp} is the empirical risk that depends only on the train set, $\lambda > 0$ and $\Omega(f)$ is a strictly monotonic penalty functional. By the representer theorem [9] the optimal solution is $\hat{f}(.) = \sum_{i=1}^{m} \alpha_i k(x_i, .)$. In most problems the kernel function is chosen apriori, and the optimal function $\hat{f}$ is learnt that minimizes some regularized empirical loss function. By making both the empirical loss function and the regularization convex the problem reduces to a convex optimization problem. [1]

While kernel functions can be seen as measuring the similarity between two points, hyperkernel functions [10, 11] measure the similarity between 2 pairs of points. Let $\underline{\mathcal{X}} = \mathcal{X}^2$. A hyperkernel $\underline{\mathrm{K}}$ on $\underline{\mathcal{X}}$ is defined as follows:

Definition 1. *$\underline{K} : \underline{\mathcal{X}} \times \underline{\mathcal{X}} \rightarrow R$ is called a hyperkernel on $\underline{\mathcal{X}}$ iff $\underline{\mathrm{K}}$ is symmetric positive definite on $\underline{\mathcal{X}}$, and $\underline{K}_{\underline{x}}$ which is the function obtained by keeping one pair of arguments of the hyperkernel fixed is also symmetric positive definite.*

Analogous to the representer theorem for kernels we have the representer theorem for hyperkernels, which is as follows:

Theorem 1. *[10] Let $\mathcal{X}$ and X be as defined above and K be an unknown kernel function. Then the minimizer of the function $R_{reg}(K, X) = R_{emp}(k, X) + \lambda ||K||^2_{\underline{\mathcal{H}}}$, where $\underline{\mathcal{H}}$ is the Hilbert space induced by the hyperkernel $\underline{K}$, and R_{emp} is some empirical loss function over the dataset X is given by $k(x, x') = \sum_{\substack{p=1 \\ q=1}}^{m} \beta_{p,q} \underline{K}((x, x'), (x_p, x_q))$.*

3 Problem Formulation

We learn the density function by first learning a kernel function that minimizes the regularized leave-one-out-likelihood score over the train dataset, and then deriving a KDE estimator using the optimal kernel learnt. We minimize the function

$$L_{\text{reg}}(k, X) = \underbrace{-log\left(\prod_{i=1}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} k(x_i, x_j)\right)}_{\text{empirical loss}} + \underbrace{\frac{\lambda}{2} ||k||^2_{\underline{\mathcal{H}}}}_{\text{regularization term}}$$

Applying the representer theorem we have

$$\|k\|^2_{\underline{\mathcal{H}}} = \sum_{\substack{i=1 \\ j=1}}^{m} \sum_{p=1}^{m} \sum_{q=1}^{m} \beta_{i,j} \beta_{p,q} \underline{K}((x_i, x_j), (x_p, x_q)) = \beta^T \underline{K} \beta$$

Since $\underline{\mathrm{K}}$ is a symmetric positive definite kernel, $||k||^2_{\mathcal{H}}$ is convex in β. Now consider the empirical loss term. By the representer theorem we have

$$L_{\text{emp}}(k, X) = -\sum_{i=1}^{m} log\left(\sum_{\substack{j=1 \\ j \neq i}}^{m} \left(\sum_{p=1}^{m} \left(\sum_{q=1}^{m} \beta_{p,q} \underline{K}((x_i, x_j), (x_p, x_q))\right)\right)\right)$$

which is convex in β. Again by the representer theorem we have

$$\hat{f}(x) = \frac{1}{m} \sum_{i=1}^{m} k(x, x_i) = \frac{1}{m} \sum_{i=1}^{m} \sum_{p=1}^{M} \sum_{q=1}^{m} \beta_{p,q} \underline{K}((x, x_i), (x_p, x_q))$$

Hence we solve the following convex optimization problem

$$\min_{\beta} -\sum_{i=1}^{m} log\left(\sum_{j=1}^{m} \left(\sum_{p=1}^{m} \left(\sum_{q=1}^{m} \beta_{p,q} \underline{K}((x_i, x_j), (x_p, x_q))\right)\right)\right) + \lambda \beta^T \underline{K} \beta \quad (2)$$

[1] It is important to note that positive definite kernels are different from the smoothing kernels that we used in equation 1. For example an epanechnikov kernel is a smoothing kernel that is not positive definite

subject to

$$\beta \geq 0 \tag{3}$$

$$\frac{1}{m}\sum_{j=1}^{m}\sum_{p=1}^{m}\sum_{q=1}^{m}\beta_{p,q}\int_x \underline{K}\left((x,x_j),(x_p,x_q)\right)\,\mathrm{d}x = 1 \tag{4}$$

where $\beta \in \mathcal{R}^{m^2\times 1}$, $\underline{K} \geq 0, \lambda > 0, \underline{K} \in S_+^{m^2}$. Constraint 4 enusres that the density function learnt integrates to one and the constraint 3 along with $\underline{K} \geq 0$ ensures that the density function is positive everywhere.

4 Experiments

For our experiments we used the Gaussian hyperkernel to solve the optimization problem shown in equation 2. Gaussian hyperkernels [12] have the closed form $K\left((x_1,x_1'),(x_2,x_2')\right) = G\left(x_1,x_1',\sigma\sqrt{2}\right)G\left(x_2,x_2',\sigma\sqrt{2}\right)G\left(\overline{x}_1,\overline{x}_2,\sqrt{\sigma^2+\sigma_h^2}\right)$ where $\bar{x_i} = \frac{x_i+x_i'}{2}$ and $G(x,y,\sigma) = \left(\sqrt{2\pi\sigma^2}\right)^{\frac{-d}{2}} e^{-\frac{(x-y)^2}{2\sigma^2}}$, d is the dimensionality of the data. While σ has the usual interpretation of bandwidth as in Gaussian kernel, $\sqrt{\sigma^2+\sigma_h^2}$ can be seen as the bandwidth of a spatially varying Gaussian scaling factor. The learnt density function $\hat{f}$ takes the form

$$\hat{f}(x) = \frac{1}{m}\sum_{i=1}^{m} G\left(x,x_i,\sigma\sqrt{2}\right) \underbrace{\sum_{p=1}^{m}\sum_{q=1}^{m}\beta_{p,q}G\left(x_p,x_q,\sigma\sqrt{2}\right)G\left(\frac{x+x_i}{2},\frac{x_p+x_q}{2},\sqrt{\sigma^2+\sigma_h^2}\right)}_{\text{scaling}(x_i,x)}$$

Our density estimate at a point x can be seen as a sum of scaled Gaussian kernel contributions of different train points at the point x. The scaling depends both on the train point x_i and the test point x. Hence our density estimation procedure can be seen as "variable bandwidth" KDE with the bandwidth varying both with the test point and the train points. A special case is when $\sigma_h \to \infty$. In this setting the scaling is independent of x, x_i. In fact we have empirically confirmed that the scaling is equal to one and hence all we are doing is "fixed bandwidth" KDE with $h = \sigma\sqrt{2}$

We tested our formulation with one-dimensional datasets proposed in [13] which are all mixtures of gaussians with varying levels of smoothness. Due to space constraints we present results on a few datasets. For all our experiments the size of the train and the test sets is 16 and we pick the parameters $\sigma_h, \sigma, \lambda$ by cross-validation. The optimization problem 2 was solved using SeDuMi [14]. We report the likelihood, and the RMSE of the test set as outputted by our algorithm and compare it to "fixed bandwidth" KDE with a Gaussian kernel, whose bandwidth is selected by likelihood cross-validation

Distribution Functional Form	Parameters	NLL-HKDE NLL- KDE	RMSE-HKDE RMSE-KDE
Strongly Skewed $\sum_{l=0}^{7}\frac{1}{8}N\left(3\left(\left(\frac{2}{3}\right)^l-1\right),\left(\frac{4}{9}\right)^l\right)$	$\sigma = 0.30$ h=0.44	6.88 7.22	0.3057 0.4357
Claw Density $\frac{1}{2}N(0,1)+\sum_{l=0}^{4}\frac{1}{10}N\left(\frac{l}{2}-1,0.01\right)$	$\sigma = 0.46$ h=0.65	7.92 8.97	0.1843 0.1942
Trimodal $\frac{9}{20}N\left(-\frac{6}{5},\frac{9}{25}\right)+\frac{9}{20}N\left(\frac{6}{5},\frac{9}{25}\right)+\frac{1}{10}N\left(0,\frac{1}{16}\right)$	$\sigma = 1$ h=0.85	10.44 10.97	0.2144 0.2144

Table 1: Table compaing "fixed bandwidth" KDE and KDE using optimal hyperkernel for different test densities.NLL means Negative Log likelihood, HKDE means Hyperkernel density estimation. For all three datasets optimal σ_h and λ come to 0.05,0.001 respectively

5 Conclusions and Future Work

We presented a learning algorithm for optimal smoothing kernel selection for density estimation. By using hyperkernels and minimizing the penalized negative leave-one-out likelihood score we are able to learn an optimal kernel and hence a density. We present experimental results on small univariate

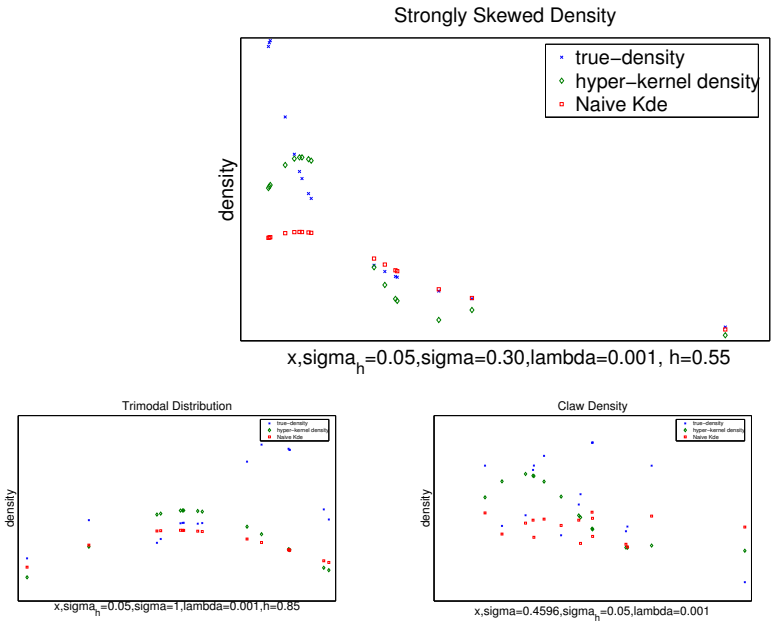

Figure 1: Comparison of hyperkernel density function with ”fixed bandwidth” KDE(which is shown as ”Naive” KDE) using Gaussian kernels for different datasets

datasets and compare it to ”fixed bandwidth” KDE. Future work will involve implementing a custom optimizer to solve both large scale and multivariate density estimation problems, and experimenting with other loss functions.

References

[1] BW Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman Hall/CRC, 1986.

[2] E. Parzen. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[3] Luc Devroye, L. Gyrfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[4] M. Lejeune and P. Sarda. Smooth estimators of distribution and density functions. *Comput. Statist. Data Anal*, 1992.

[5] T. Gasser, H. G. Mller, and V. Mammitzsch. Kernels for nonparametric curve estimation. *Journal of the Royal Statistical Society. Series B. Methodological*, 1985.

[6] Le Song, Xinhua Zhang, Arthur Gretton, Bernhard Schlkopf, Alex Smola, and J. Skolnick. Tailoring density estimation via reproducing kernel moment matching. In *ICML*, 2008.

[7] V. N. Vapnik and S. Mukherjee. Support vector method for multivariate density estimation. *Advances in Neural Information Processing Systems*, pages 659–665, 1999.

[8] B. Schlkopf and A. J. Smola. *Learning with kernels*. The MIT Press, 2002.

[9] G.S.Kimeldorf and G.Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 1971.

[10] CS Ong, A. Smola, and RC Williamson. Learning with Hyperkernels. *Journal of Machine Learning Research, submitted*, 2003.

[11] H. Ouyang and A. Gray. Learning dissimilarities by ranking: from SDP to QP. In *Proceedings of the ICML*. ACM New York, NY, USA, 2008.

[12] R. Kondor and T. Jebara. Gaussian and Wishart Hyperkernels. *Asvance In NIPS*, 19, 2007.

[13] JS Marron and MP Wand. Exact mean integrated squared error. *The Annals of Statistics*, pages 712–736, 1992.

[14] J. F. Sturm. Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 1999.