Fast optimal bandwidth selection for kernel density estimation

Vikas Chandrakant Raykar and Ramani Duraiswami
Dept. of computer science and UMIACS, University of Maryland, CollegePark
{vikas,ramani}@cs.umd.edu

Abstract

We propose a computationally efficient $\epsilon-exact$ approximation algorithm for univariate Gaussian kernel based density derivative estimation that reduces the computational complexity from $O(MN)$ to linear $O(N+M)$. We apply the procedure to estimate the optimal bandwidth for kernel density estimation. We demonstrate the speedup achieved on this problem using the "solve-the-equation plug-in" method, and on exploratory projection pursuit techniques.

1 Introduction

Kernel density estimation techniques [10] are widely used in various inference procedures in machine learning, data mining, pattern recognition, and computer vision. Efficient use of these methods requires the optimal selection of the *bandwidth* of the kernel. A series of techniques have been proposed for data-driven bandwidth selection [4]. The most successful state of the art methods rely on the estimation of general integrated squared *density derivative functionals*. This is the most computationally intensive task with $O(N^2)$ cost, in addition to the $O(N^2)$ cost of computing the kernel density estimate. The core task is to *efficiently compute an estimate of the density derivative*. Currently the most practically successful approach, *solve-the-equation plug-in* method [9] involves the numerical solution of a non-linear equation. Iterative methods to solve this equation will involve repeated use of the density functional estimator for different bandwidths which adds much to the computational burden. Estimation of density derivatives is needed in various other applications like estimation of modes and inflexion points of densities [2] and estimation of the derivatives of the projection index in projection pursuit algorithms [5].

2 Optimal bandwidth selection

A univariate random variable X on $\mathbf{R}$ has a density p if, for all Borel sets A of $\mathbf{R}$, $\int_A p(x)dx = \Pr[x \in A]$. The task of density estimation is to estimate p from an i.i.d. sample $x_1, \ldots, x_N$ drawn from p. The estimate $\widehat{p} : \mathbf{R} \times (\mathbf{R})^N \rightarrow \mathbf{R}$ is called the *density estimate*. The most popular non-parametric method for density estimation is the *kernel density estimator* (KDE) [10]

$$\widehat{p}(x) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right), \tag{2.1}$$

where $K(u)$ is the *kernel function* and h is the *bandwidth*. The kernel $K(u)$ is required to satisfy the following two conditions:

$$K(u) \geq 0 \text{ and } \int_{\mathbf{R}} K(u)du = 1. \tag{2.2}$$

The most widely used kernel is the Gaussian of zero mean and unit variance. In this case the KDE can be written as

$$\widehat{p}(x) = \frac{1}{N\sqrt{2\pi h^2}}\sum_{i=1}^{N} e^{-(x-x_i)^2/2h^2}. \tag{2.3}$$

The computational cost of evaluating Eq. 2.3 at N points is $O(N^2)$, making it prohibitively expensive. The *Fast Gauss Transform* (FGT) [3] is an approximation algorithm that reduces the computational complexity to $O(N)$, at the expense of reduced precision. Yang et al. [11] presented an extension the *improved fast Gauss transform*(IFGT) that scaled well with dimensions. The main contribution of the current paper is the extension of the IFGT to accelerate the kernel *density derivative* estimate, and solve the *optimal bandwidth problem*.

The *integrated square error* (ISE) between the estimate $\widehat{p}(x)$ and the actual density $p(x)$ is given by $\text{ISE}(\widehat{p}, p) = \int_{\mathbf{R}}[\widehat{p}(x) - p(x)]^2 dx$. The ISE depends on a particular realization of N points. It can be averaged over these realizations to get the *mean integrated squared error* (MISE). An asymptotic large sample approximation for MISE, the AMISE, is usually derived via the Taylor's series. The A here is for asymptotic. Based on certain assumptions, the AMISE between the actual density and the estimate can be shown to be

$$\text{AMISE}(\widehat{p}, p) = \frac{1}{Nh}R(K) + \frac{1}{4}h^4\mu_2(K)^2 R(p^{''}), \tag{2.4}$$

where, $R(g) = \int_{\mathbf{R}} g(x)^2 dx$, $\mu_2(g) = \int_{\mathbf{R}} x^2 g(x)dx$, and $p^{''}$ is the second derivative of the density p. The first term in Eq. 2.4 is the integrated variance and the second term is the integrated squared bias. The bias is proportional to h^4 whereas the variance is proportional to $1/Nh$, which leads to the well known *bias-variance tradeoff*. Based on the AMISE expression the optimal bandwidth h_{AMISE} can be obtained by differentiating Eq. 2.4 w.r.t. bandwidth h and setting it to zero.

$$h_{AMISE} = \left[\frac{R(K)}{\mu_2(K)^2 R(p^{''})N}\right]^{1/5}. \tag{2.5}$$

However this expression cannot be used directly since $R(p^{''})$ depends on the second derivative of the density p. In order to estimate $R(p^{''})$ we will need an estimate of the density derivative.

A simple estimator for the density derivative can be obtained by taking the derivative of the KDE $\widehat{p}(x)$ defined earlier [1]. The r^{th} density derivative estimate $\widehat{p}^{(r)}(x)$ can be written as

$$\widehat{p}^{(r)}(x) = \frac{1}{Nh^{r+1}} \sum_{i=1}^{N} K^{(r)} \left(\frac{x - x_i}{h} \right), \tag{2.6}$$

where $K^{(r)}$ is the r^{th} derivative of the kernel K. The r^{th} derivative of the Gaussian kernel $k(u)$ is given by $K^{(r)}(u) = (-1)^r H_r(u) K(u)$, where $H_r(u)$ is the r^{th} Hermite polynomial. Hence the density derivative estimate can be written as

$$\widehat{p}^{(r)}(x) = \frac{(-1)^r}{\sqrt{2\pi} N h^{r+1}} \sum_{i=1}^{N} H_r \left(\frac{x - x_i}{h} \right) e^{-(x-x_i)^2/2h^2}. \tag{2.7}$$

The computational complexity of evaluating the r^{th} derivative of the density estimate due to N points at M target locations is $O(rNM)$. Based on similar analysis the optimal bandwidth h^r_{AMISE} to estimate the r^{th} density derivative can be shown to be [10]

$$h^r_{AMISE} = \left[\frac{R(K^{(r)})(2r+1)}{\mu_2(K)^2 R(p^{(r+2)}) N} \right]^{1/2r+5}. \tag{2.8}$$

3 Estimation of density functionals

Rather than requiring the actual density derivative, methods for automatic bandwidth selection require the estimation of what are known as *density functionals*. The general integrated squared density derivative functional is defined as $R(p^{(s)}) = \int_{\mathbf{R}} \left[p^{(s)}(x) \right]^2 dx$. Using integration by parts, this can be written in the following form, $R(p^{(s)}) = (-1)^s \int_{\mathbf{R}} p^{(2s)}(x) p(x) dx$. More specifically for even s we are interested in estimating density functionals of the form,

$$\Phi_r = \int_{\mathbf{R}} p^{(r)}(x) p(x) dx = E\left[p^{(r)}(X) \right]. \tag{3.9}$$

An estimator for Φ_r is,

$$\widehat{\Phi}_r = \frac{1}{N} \sum_{i=1}^{N} \widehat{p}^{(r)}(x_i). \tag{3.10}$$

where $\widehat{p}^{(r)}(x_i)$ is the estimate of the r^{th} derivative of the density $p(x)$ at $x = x_i$. Using a kernel density derivative estimate for $\widehat{p}^{(r)}(x_i)$ (Eq. 2.6) we have

$$\widehat{\Phi}_r = \frac{1}{N^2 h^{r+1}} \sum_{i=1}^{N} \sum_{j=1}^{N} K^{(r)}\left(\frac{x_i - x_j}{h} \right). \tag{3.11}$$

It should be noted that computation of $\widehat{\Phi}_r$ is $O(rN^2)$ and hence can be very expensive if a direct algorithm is used. The optimal bandwidth for estimating the density functional is given by [10]

$$h^r_{\mathrm{AMSE}} = \left[\frac{-2K^{(r)}(0)}{\mu_2(K) \Phi_{r+2} N} \right]^{1/r+3}. \tag{3.12}$$

4 Solve-the-equation plug-in method

The most successful among all current bandwidth selection methods [4], both empirically and theoretically, is the *solve-the-equation plug-in* method [4]. We use the following version as described in [9].

The AMISE optimal bandwidth is the solution to the equation

$$h = \left[\frac{R(K)}{\mu_2(K)^2 \widehat{\Phi}_4[\gamma(h)] N} \right]^{1/5}, \tag{4.13}$$

where $\widehat{\Phi}_4[\gamma(h)]$ is an estimate of $\Phi_4 = R(p^{''})$ using the pilot bandwidth $\gamma(h)$, which depends on h. The bandwidth is chosen such that it minimizes the asymptotic MSE for the estimation of Φ_4 and is

$$g_{\mathrm{MSE}} = \left[\frac{-2K^{(4)}(0)}{\mu_2(K) \Phi_6 N} \right]^{1/7}. \tag{4.14}$$

Substituting for N, g_{MSE} can be written as a function of h as follows

$$g_{\mathrm{MSE}} = \left[\frac{-2K^{(4)}(0) \mu_2(K) \Phi_4}{R(K) \Phi_6} \right]^{1/7} h^{5/7}_{\mathrm{AMISE}}. \tag{4.15}$$

This suggests that we set

$$\gamma(h) = \left[\frac{-2K^{(4)}(0) \mu_2(K) \widehat{\Phi}_4(g_1)}{R(K) \widehat{\Phi}_6(g_2)} \right]^{1/7} h^{5/7}, \tag{4.16}$$

where $\widehat{\Phi}_4(g_1)$ and $\widehat{\Phi}_6(g_2)$ are estimates of Φ_4 and Φ_6 using bandwidths g_1 and g_2 respectively. The bandwidths g_1 and g_2 are chosen such that it minimizes the asymptotic MSE.

$$g_1 = \left[\frac{-6}{\sqrt{2\pi} \widehat{\Phi}_6 N} \right]^{1/7} \quad g_2 = \left[\frac{30}{\sqrt{2\pi} \widehat{\Phi}_8 N} \right]^{1/9} \tag{4.17}$$

where $\widehat{\Phi}_6$ and $\widehat{\Phi}_8$ are estimators for Φ_6 and Φ_8 respectively. We can use a similar strategy for estimation of Φ_6 and Φ_8. However this problem will continue since the optimal bandwidth for estimating Φ_r will depend on Φ_{r+2}. The usual strategy is to estimate a Φ_r at some stage, using a quick and simple estimate of bandwidth chosen with reference to a parametric family, usually a normal density. It has been observed that as the number of stages increases, the variance of the bandwidth increases. The most common choice is to use only two stages. If p is a normal density with variance σ^2 then for

even r we can compute Φ_r exactly [10]. An estimator of Φ_r will use an estimate $\widehat{\sigma}^2$ of the variance. Based on this we can estimate Φ_6 and Φ_8 as

$$\widehat{\Phi}_6 = \frac{-15}{16\sqrt{\pi}}\widehat{\sigma}^{-7},\ \widehat{\Phi}_8 = \frac{105}{32\sqrt{\pi}}\widehat{\sigma}^{-9}. \tag{4.18}$$

The two stage solve-the-equation method using the Gaussian kernel can be summarized as follows. (1) Compute an estimate $\widehat{\sigma}$ of the standard deviation. (2) Estimate the density functionals Φ_6 and Φ_8 using the normal scale rule (Eq. 4.18). (3) Estimate the density functionals Φ_4 and Φ_6 using Eq. 3.11 with the optimal bandwidth g_1 and g_2 (Eq. 4.17). (4) The bandwidth is the solution to the nonlinear Eq. 4.13 which can be solved using any numerical routine like the Newton-Raphson method. The main computational bottleneck is the estimation of Φ which is of $O(N^2)$.

5 Fast density derivative estimation

To estimate the density derivative at M *target points*, $\{y_j \in \mathbf{R}\}_{j=1}^{M}$, we need to evaluate sums such as

$$G_r(y_j) = \sum_{i=1}^{N} q_i H_r\left(\frac{y_j - x_i}{h_1}\right) e^{-(y_j-x_i)^2/h_2^2} j = 1, \ldots, M, \tag{5.19}$$

where $\{q_i \in \mathbf{R}\}_{i=1}^{N}$ will be referred to as the *source weights*, $h_1 \in \mathbf{R}^+$ is the bandwidth of the Gaussian and $h_2 \in \mathbf{R}^+$ is the bandwidth of the Hermite polynomial. The computational complexity of evaluating Eq. 5.19 is $O(rNM)$. For any given $\epsilon > 0$ the algorithm computes an approximation $\widehat{G}_r(y_j)$ such that

$$\left|\hat{G}_r(y_j) - G_r(y_j)\right| \leq Q\epsilon, \tag{5.20}$$

where $Q = \sum_{i=1}^{N} |q_i|$. We call $\widehat{G}_r(y_j)$ an $\epsilon - exact$ approximation to $G_r(y_j)$. We describe the algorithm briefly. More details can be found in [8].

For any point $x_* \in \mathbf{R}$ the Gaussian can be written as,

$$\begin{aligned} e^{-\|y_j-x_i\|^2/h_2^2} &= e^{-\|x_i-x_*\|^2/h_2^2} e^{-\|y_j-x_*\|^2/h_2^2} \\ &\quad e^{2(x_i-x_*)(y_j-x_*)/h_2^2}. \end{aligned} \tag{5.21}$$

In Eq. 5.21 for the third exponential $e^{2(y_j-x_*)(x_i-x_*)/h^2}$ the source and target are entangled. This entanglement is separated using the Taylor's series expansion as follows.

$$\begin{aligned} e^{2(x-x_*)(y-x_*)/h^2} &= \sum_{k=0}^{p-1} \frac{2^k}{k!}\left(\frac{x-x_*}{h}\right)^k \left(\frac{y-x_*}{h}\right)^k \\ &\quad + \ error, \end{aligned} \tag{5.22}$$

Using this the Gaussian can now be factorized as

$$\begin{aligned} e^{-\|y_j-x_i\|^2/h_2^2} &= \sum_{k=0}^{p-1} \frac{2^k}{k!}\left[e^{-\|x_i-x_*\|^2/h_2^2}\left(\frac{x_i-x_*}{h_2}\right)^k\right] \\ &\quad \left[e^{-\|y_j-x_*\|^2/h_2^2}\left(\frac{y_j-x_*}{h_2}\right)^k\right] + err. \end{aligned} \tag{5.23}$$

The r^{th} Hermite polynomial can be factorized as [10]

$$\begin{aligned} H_r\left(\frac{y_j-x_i}{h_1}\right) &= \sum_{l=0}^{\lfloor r/2 \rfloor} \sum_{m=0}^{r-2l} a_{lm}\left(\frac{x_i-x_*}{h_1}\right)^m \\ &\quad \left(\frac{y_j-x_*}{h_1}\right)^{r-2l-m}, \text{ where} \end{aligned} \tag{5.24}$$

$$a_{lm} = \frac{(-1)^{l+m} r!}{2^l l! m! (r-2l-m)!}. \tag{5.25}$$

Using Eq. 5.23 and 5.24, $G_r(y_j)$ after ignoring the error terms can be approximated as

$$\begin{aligned} \widehat{G}_r(y_j) &= \sum_{k=0}^{p-1} \sum_{l=0}^{\lfloor r/2 \rfloor} \sum_{m=0}^{r-2l} a_{lm} B_{km} e^{-\|y_j-x_*\|^2/h_2^2} \\ &\quad \left(\frac{y_j-x_*}{h_2}\right)^k \left(\frac{y_j-x_*}{h_1}\right)^{r-2l-m}, \text{ where} \end{aligned}$$

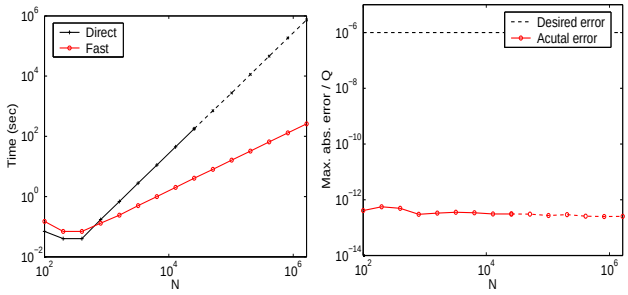
$$B_{km} = \frac{2^k}{k!} \sum_{i=1}^{N} q_i e^{-\|x_i-x_*\|^2/h_2^2} \left(\frac{x_i-x_*}{h_2}\right)^k \left(\frac{x_i-x_*}{h_1}\right)^m.$$

Thus far, we have used the Taylor's series expansion about a certain point x_*. However if we use the same x_* for all the points we typically would require very high truncation number p since the Taylor's series gives good approximation only in a small open interval around x_*. We uniformly sub-divide the space into K intervals of length $2r_x$. The N source points are assigned into K clusters, S_n for $n = 1, \ldots, K$ with c_n being the center of each cluster. The aggregated coefficients are now computed for each cluster and the total contribution from all the clusters is summed up. Since the Gaussian decays very rapidly a further speedup is achieved if we ignore all the sources belonging to a cluster if the cluster is greater than a certain distance from the target point, i.e., $\|y_j - c_n\| > r_y$. Substituting $h_1 = h$ and $h_2 = \sqrt{2}h$ the final algorithm can be written as

$$\begin{aligned} \widehat{G}_r(y_j) &= \sum_{\|y_j-c_n\| \leq r_y} \sum_{k=0}^{p-1} \sum_{l=0}^{\lfloor r/2 \rfloor} \sum_{m=0}^{r-2l} a_{lm} B_{km}^n \\ &\quad e^{-\|y_j-c_n\|^2/2h^2} \left(\frac{y_j-c_n}{h}\right)^{k+r-2l-m} \end{aligned} \tag{5.26}$$

$$B_{km}^n = \frac{1}{k!} \sum_{x_i \in S_n} q_i e^{-\|x_i-c_n\|^2/2h^2} \left(\frac{x_i-c_n}{h}\right)^{k+m}.$$

5.1 Computational and space complexity Computing the coefficients B_{km}^n for all the clusters is $O(prN)$. Evaluation of $\widehat{G}_r(y_j)$ at M points is $O(npr^2M)$, where n if the maximum number of neighbor clusters which influence y_j. Hence the total computational complexity is $O(prN + npr^2M)$. For each cluster we need to store all the pr coefficients. Hence the storage needed is of $O(prK + N + M)$.



(a) (b)

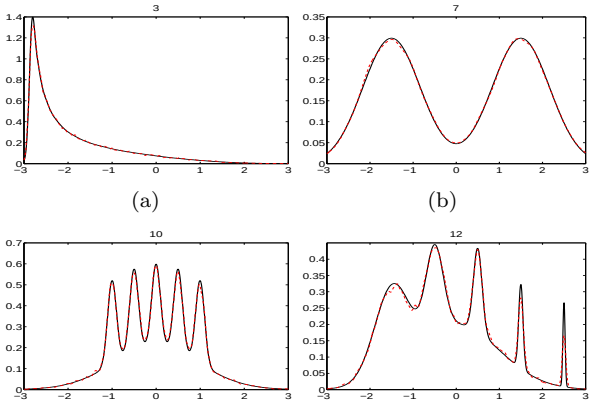
Figure 1: (a) Running time in seconds and (b) maximum absolute error relative to Q for the direct and the fast methods as a function of N. $N = M$ source and the target points were uniformly distributed $[0, 1]$. [$h = 0.1$, $r = 4$, and $\epsilon = 10^{-6}$.]

5.2 Choosing the parameters Given any $\epsilon > 0$, we want to choose the following parameters, K (the number of intervals), r_y (the cut off radius for each cluster), and p (the truncation number) such that for any target point y_j, $\left|\hat{G}_r(y_j) - G_r(y_j)\right| \leq Q\epsilon$. We give the final results for the choice of the parameters. The detailed derivations can be seen in the technical report [8]. The number of clusters K is chosen such that $r_x = h/2$. The cutoff radius r_y is given by $r_y = r_x + 2h\sqrt{\ln(\sqrt{r!}/\epsilon)}$. The truncation number p is chosen such that $\Delta\big|_{[b=\min(b_*, r_y),\, a=rx]} \leq \epsilon$, where, $\Delta = \frac{\sqrt{r!}}{p!}\left(\frac{ab}{h^2}\right)^p e^{-(a-b)^2/4h^2}$, and $b_* = \frac{a+\sqrt{a^2+8ph^2}}{2}$.

5.3 Numerical experiments The algorithm was programmed in C++ and was run on a 1.6 GHz Pentium M processor with 512Mb of RAM. Figure 1 shows the running time and the maximum absolute error relative to Q for both the direct and the fast methods as a function of $N = M$. We see that the running time of the fast method grows linearly as the number of sources and targets increases, while that of the direct evaluation grows quadratically.

6 Speedup achieved for bandwidth estimation

We demonstrate the speedup achieved on the mixture of normal densities used by Marron and Wand [6]. The family of normal mixture densities is extremely rich and, in fact any density can be approximated arbitrarily well by a member of this family. Fig. 2 shows a sample of four different densities out of the fifteen densities which were used by the authors in [6] as typical representatives of the densities likely to be encountered in real data situations. We sampled $N = 50,000$ points from each density. The AMISE optimal bandwidth was estimated using both the direct methods and the proposed fast method. Table 1 shows the speedup achieved and the



(a) (b) (c) (d)

Figure 2: Four normal mixture densities from Marron and Wand [6].The solid line shows the actual density and the dotted line is the estimated density using the optimal bandwidth.

absolute relative error. We also used the Adult database from the UCI machine learning repository [7]. The database extracted from the census bureau database contains 32,561 training instances with 14 attributes per instance. Table 2 shows the speedup achieved and the absolute relative error for two of the continuous attributes.

7 Projection Pursuit

Projection Pursuit (PP) is an exploratory technique for visualizing and analyzing large multivariate datasets [5]. The idea of PP is to search for projections from high- to low-dimensional space that are most *interesting*. The PP algorithm for finding the most interesting one-dimensional subspace is as follows. First project each data point onto the direction vector $a \in \mathbf{R}^d$, i.e., $z_i = a^T x_i$. Compute the univariate nonparametric kernel density estimate, $\widehat{p}$, of the projected points z_i. Compute the projection index $I(a)$ based on the density estimate. Locally optimize over the the choice of a, to get the *most interesting* projection of the data. Repeat from a new initial projection to get a different view. The projection index is designed to reveal specific structure in the data, like clusters, outliers, or smooth manifolds. The entropy index based on Rényi's order-1 entropy is given by $I(a) = \int p(z) \log p(z) dz$. The density of zero mean and unit variance which uniquely minimizes this is the standard normal density. Thus the projection index finds the direction which is most non-normal. In practice we need to use an estimate $\widehat{p}$ of the the true density p, for example the KDE using the Gaussian kernel. Thus we have an estimate of the entropy index as follows

$$\widehat{I}(a) = \int \log \widehat{p}(z) p(z) dz = \frac{1}{N} \sum_{i=1}^{N} \log \widehat{p}(a^T x_i). \quad (7.27)$$

Table 1: The running time in seconds for the direct and the fast methods for four normal mixture densities of Marron and Wand [6] (See Fig. 2). The absolute relative error is defined as $|h_{direct} - h_{fast}/h_{direct}|$. For the fast method we used $\epsilon = 10^{-3}$.

Density	h_{direct}	h_{fast}	T_{direct} (sec)	T_{fast} (sec)	Speedup	Abs. Relative Error
(a)	0.020543	0.020543	8523.26	101.62	83.87	1.53e-006
(b)	0.092240	0.092240	5918.19	88.61	66.79	6.34e-006
(c)	0.024326	0.024326	7186.07	106.17	67.69	1.84e-006
(d)	0.032492	0.032493	8310.90	119.02	69.83	3.83e-006

Table 2: Optimal bandwidth estimation for two continuous attributes for the Adult database [7].

Attribute	h_{direct}	h_{fast}	T_{direct} (sec)	T_{fast} (sec)	Speedup	Abs. Relative Error
Age	0.860846	0.860856	4679.03	66.42	70.45	1.17e-005
fnlwgt	4099.564359	4099.581141	4637.09	68.83	67.37	4.09e-006

The entropy index $\widehat{I}(a)$ has to be optimized over the d-dimensional vector a subject to the constraint that $\|a\| = 1$. The optimization function will require the gradient of the objective function. For the index defined above the gradient can be written as $\frac{d}{da}[\widehat{I}(a)] = \frac{1}{N}\sum_{i=1}^{N} \frac{\widehat{p}'(a^T x_i)}{\widehat{p}(a^T x_i)} x_i$. For the PP the computational burden is greatly reduced if we use the proposed fast method. The computational burden is reduced in the following three instances. (1) Computation of the kernel density estimate, (2) estimation of the optimal bandwidth, and (3) computation of the first derivative of the kernel density estimate, which is required in the optimization procedure. Fig. 3 shows an example of the PP algorithm to segment an image based on color.

8 Conclusions

We proposed an fast $\epsilon - exact$ algorithm for kernel density derivative estimation which reduced the computational complexity from $O(N^2)$ to $O(N)$. We demonstrated the speedup achieved for optimal bandwidth estimation both on simulated as well as real data. A extended version of this paper is available as a technical report [8].

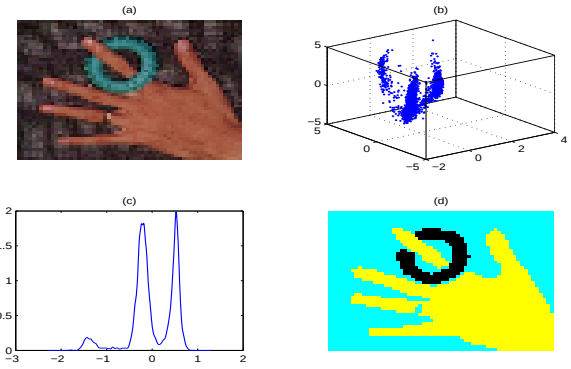


Figure 3: (a) The original image. (b) The centered and scaled RGB space. Each pixel in the image is a point in the RGB space. (c) KDE of the projection of the pixels on the most interesting direction found by projection pursuit. (d) The assignment of the pixels to the three modes in the KDE.

References

[1] P. K. Bhattacharya. Estimation of a probability density function and its derivatives. *Sankhya, Series A*, 29:373–382, 1967.

[2] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Info. Theory*, 21(1):32–40, 1975.

[3] L. Greengard and J. Strain. The fast Gauss transform. *SIAM J. Sci. Stat. Comput.*, 12(1):79–94, 1991.

[4] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *J. Amer. Stat. Assoc.*, 91(433):401–407, March 1996.

[5] M. C. Jones and R. Sibson. What is projection pursuit? *J. R. Statist. Soc. A*, 150:1–36, 1987.

[6] J. S. Marron and M. P. Wand. Exact mean integrated squared error. *The Ann. of Stat.*, 20(2):712–736, 1992.

[7] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

[8] V. C. Raykar and R. Duraiswami. Very fast optimal bandwidth selection for univariate kernel density estimation. Technical Report CS-TR-4774, University of Maryland, CollegePark, 2005.

[9] S.J. Sheather and M.C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *J. Royal Statist. Soc. B*, 53:683–690, 1991.

[10] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, 1995.

[11] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. In *IEEE Int. Conf. on Computer Vision*, pages 464–471, 2003.