Fast optimal bandwidth selection for kernel density estimation

Vikas C. Raykar and Ramani Duraiswami
University of Maryland, CollegePark
{vikas,ramani}@cs.umd.edu

2006 SIAM Conference on Data Mining
Bethesda, April 20, 2006

Gist of the paper

- Bandwidth selection for kernel density estimation scales as $\mathcal{O}(N^2)$.

- We present a fast computational technique that scales as $\mathcal{O}(N)$.

- For $50,000$ points we obtained speedups in the range 65 to 105.

Density estimation

- Widely used in exploratory data analysis, machine learning, data mining, computer vision, and pattern recognition.

- A density p gives a principled way to compute probabilities on sets.

$$\Pr[x \in A] = \int_A p(x)dx.$$

- **Estimate the density from samples $x_1, \ldots, x_N$ drawn from p.**

Different methods for density estimation

- Parametric methods.
  - Assume a functional form for the density.
- Non-parametric methods.
  - *letting the data speak for themselves*
  - Histograms.
  - **Kernel density estimators.** [Most popular.]

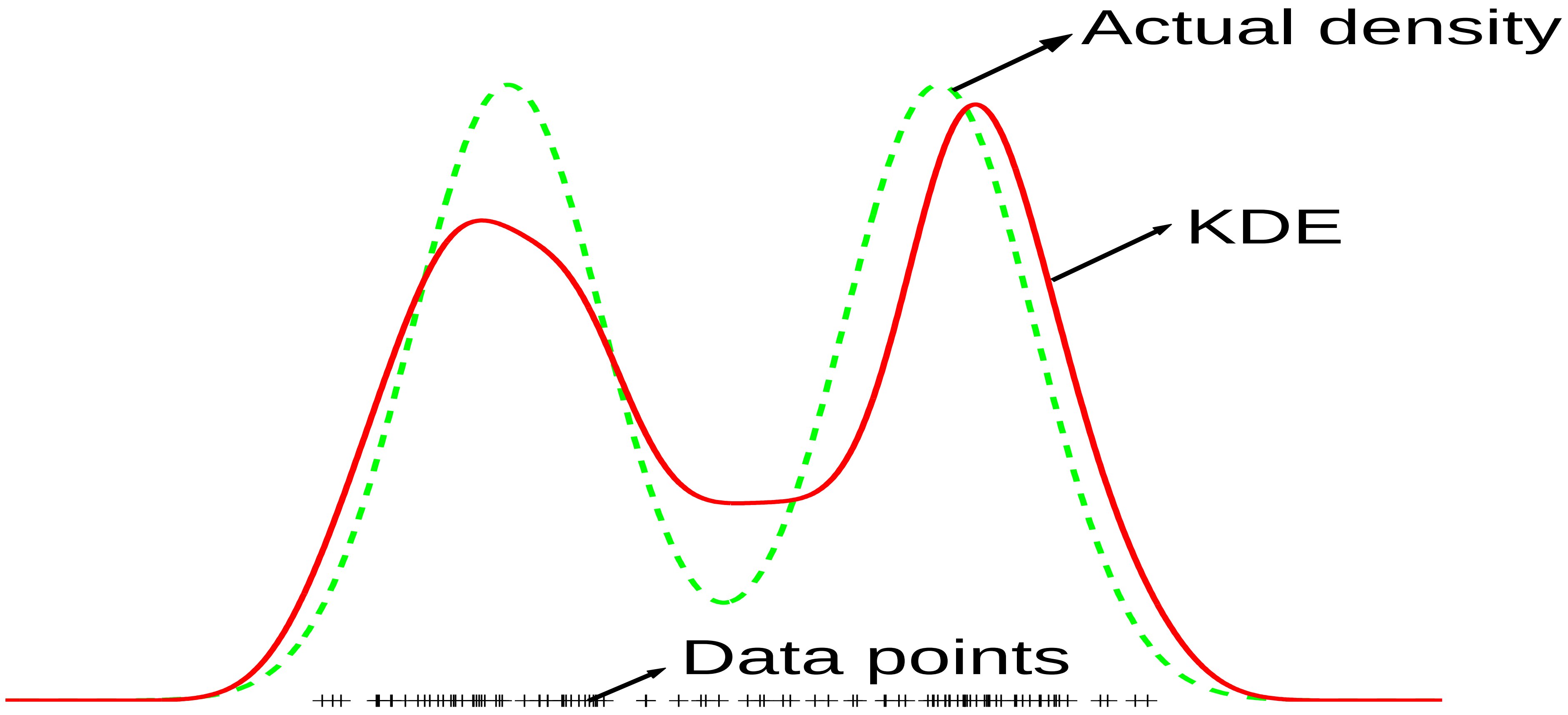
Kernel density estimate (KDE)

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

- The kernel function K is essentially spreading a probability mass of $1/N$ associated with each point about its neighborhood.

- The neighborhood size is essentially decided by the parameter h called the **bandwidth** of the kernel.

KDE illustration



Gaussian kernel

The most widely used kernel is the Gaussian of zero mean and unit variance.

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}.$$

For the Gaussian kernel the kernel density estimate can be written as

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

Computational complexity of KDE

- Essentially sum of N Gaussians.

- Computing KDE at M points scales as $\mathcal{O}(MN)$.

- Various approximate algorithms are proposed that reduce the computational complexity to $\mathcal{O}(M + N)$.

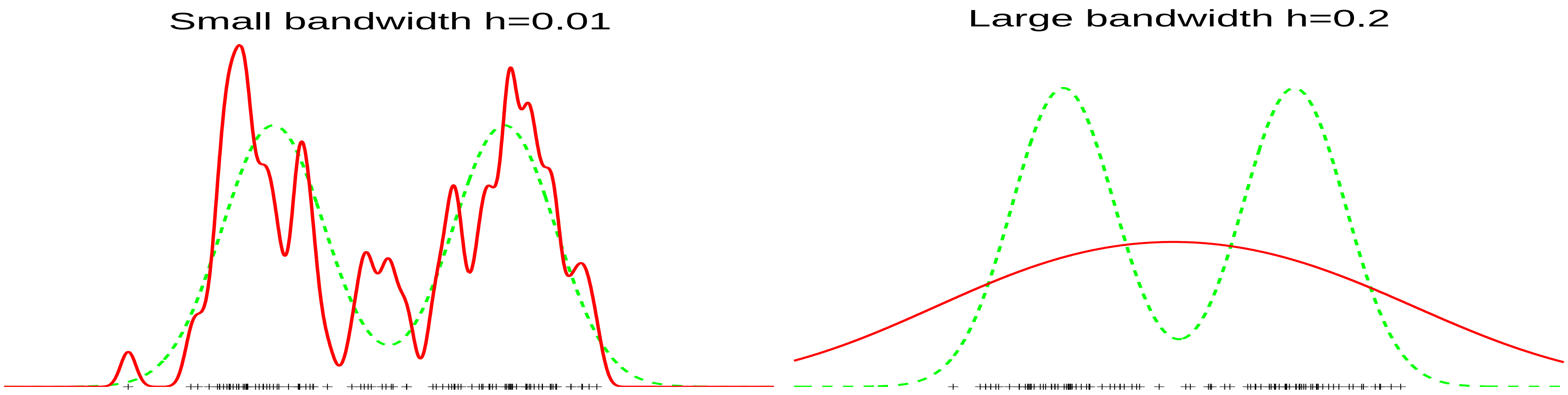
  [FFT, FGT, IFGT, dual-tree].

- This paper focuses on reducing the **computational complexity of finding the optimal bandwidth**, which scales as $\mathcal{O}(N^2)$.

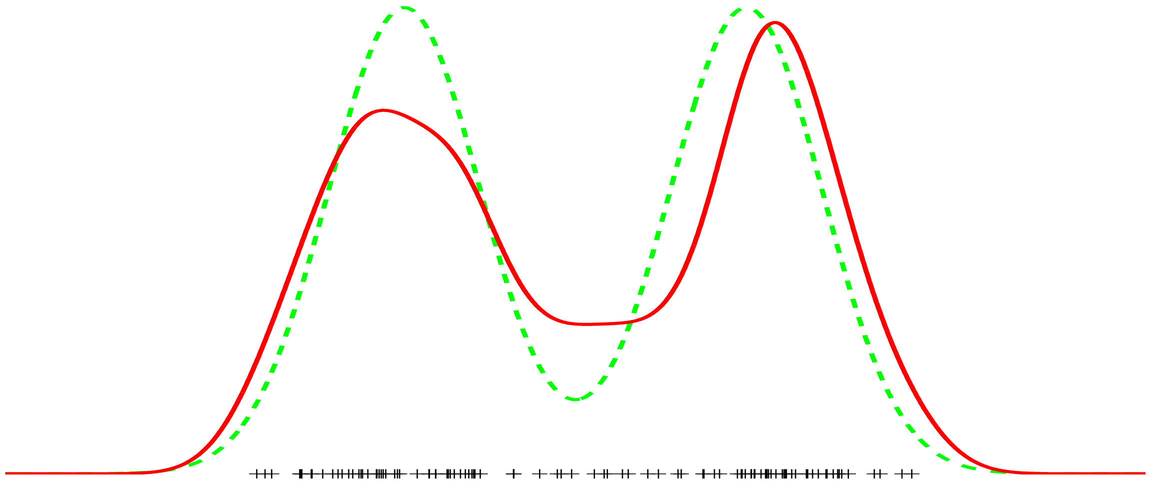
Role of bandwidth h

- As **h decreases towards 0**, the number of modes increases to the number of data points and the KDE is very noisy.

- As **h increases towards ∞**, the number of modes drops to 1, so that any interesting structure has been smeared away and the KDE just displays a unimodal pattern.



The bandwidth h has to be chosen optimally.



The sense in which the bandwidth is optimal has to be made precise.

The most widely used is the AMISE optimal bandwidth.

Performance measure

- Integrated squared error (ISE)

$$\text{ISE}(\widehat{p}, p) = \int_{\mathbf{R}} [\widehat{p}(x) - p(x)]^2 dx.$$

- Mean integrated squared error (MISE)

$$\text{MISE}(\widehat{p}, p) = E[\text{ISE}(\widehat{p}, p)] = E\left[\int_{\mathbf{R}} [\widehat{p}(x) - p(x)]^2 dx\right]$$

- A measure of the ‘average’ performance of the kernel density estimator, averaged over the support of the density and different realization of the points.

Asymptotic performance measure

- The dependence of the MISE on the bandwidth h is not very explicit.

- This makes it difficult to interpret the influence of the bandwidth on the performance of the estimator.

- An asymptotic large sample approximation for this expression is usually derived via the Taylor's series called as the AMISE, the A is for asymptotic.

AMISE

The AMISE can be shown to be *

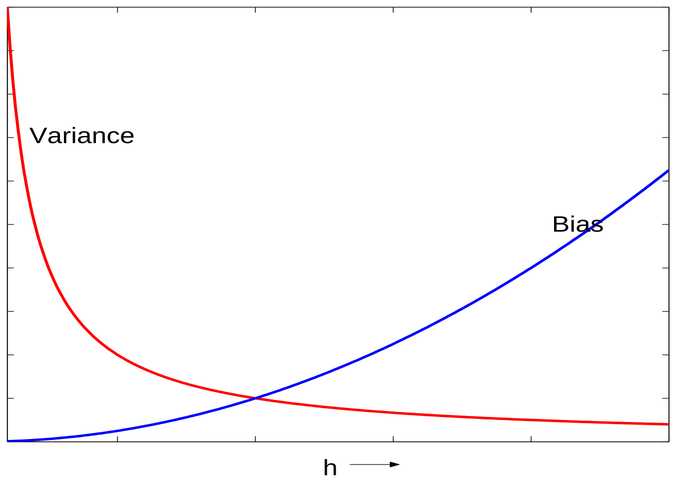
$$\begin{aligned} \text{AMISE}(\widehat{p}, p) &= \frac{1}{Nh} R(K) + \frac{1}{4} h^4 \mu_2(K)^2 R(p'') \\ &= \text{Variance} + (\text{bias})^2 \end{aligned}$$

where

$$R(g) = \int_{\mathbf{R}} g(x)^2 dx, \ , \ \mu_2(g) = \int_{\mathbf{R}} x^2 g(x) dx,$$

*Wand, M. P. and Jones, M. C. 1995. Kernel Smoothing. Chapman and Hall, London.

Bias-Variance tradeoff



- Variance is proportional to $1/h$.
- Bias is proportional to h^2.
- Optimal h is found by setting the first derivative of AMISE to zero.

AMISE optimal bandwidth

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

- This expression cannot be used directly since $R(p'')$ depends on the second derivative of the density p.

- Different strategies have been proposed to solve this problem.

- The most popular **plug-in methods** use an estimate of $R(p'')$ which in turn needs an estimate of p''.

- So for optimal bandwidth estimation we need **estimates of the density derivatives**.

Estimating density functionals

- This bandwidth for estimation of the density functional $R(p^{''})$ is quite different from the the bandwidth h used for the kernel density estimate.

- We can find an expression for the optimal bandwidth for the estimation of $R(p^{''})$.

- However this bandwidth will depend on an unknown density functional $R(p^{'''})$.

- This problem will continue since the optimal bandwidth for estimating $R(p^{(s)})$ will depend on $R(p^{(s+1)})$.

- In general **we need estimates of higher order derivatives also**.

Kernel density derivative estimation

- Take the derivative of the kernel density estimate.

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

- For the Gaussian kernel this takes the form

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

- $H_r(u)$ is the r^{th} Hermite polynomial.

Computational complexity of bandwidth estimation

- In order to estimate a density functional we need to evaluate the density derivative at N points.

- Hence computing a density functional is $\mathcal{O}(rN^2)$.

- The current most practically successful approach, **solve-the-equation plug-in** * method involves the numerical solution of a non-linear equation.

  - Iterative methods to solve this equation will involve repeated use of the density derivative functional estimator for different bandwidths which adds much further to the computational burden.

*Sheather, S. and Jones, M. 1991. A reliable data-based bandwidth selection method for kernel density estimation. Journal of Royal Statistical Society Series B 53, 683-690.

Fast $\epsilon-exact$ density derivative estimation

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

- The computational complexity is $\mathcal{O}(rNM)$.

- We will present an $\epsilon-exact$ approximation algorithm that reduces it to $\mathcal{O}(prN + npr^2M)$ where the constants p and n depends on the precision ϵ and the bandwidth h.

- For example for $N = M = 409,600$ points while the direct evaluation of the density derivative takes around 12.76 **hours** the fast evaluation requires only 65 **seconds** with an error of around $\epsilon = 10^{-12}$.

Notion of $\epsilon - exact$ approximation

For any given $\epsilon > 0$ the algorithm computes an approximation $\widehat{G}_r(y_j)$ such that

$$\left| \frac{\widehat{G}_r(y_j) - G_r(y_j)}{Q} \right| \leq \epsilon,$$

where $Q = \sum_{i=1}^{N} |q_i|$.

We call $\widehat{G}_r(y_j)$ an $\epsilon - exact$ approximation to $G_r(y_j)$.

- ϵ can be arbitrarily small.

- For machine precision accuracy there is no difference between the direct and the fast methods.

Algorithm

- The fast algorithm is based on separating the x_i and y_j.
- The Gaussian is factorized via Taylor's series.
- Only first few terms are retained.
- Need to derive good error bounds to decide how many terms to retain to achieve a desired error.
- The Hermite is factorized via the binomial theorem.

Factorization of the Gaussian

$$e^{-\|y_j-x_i\|^2/h_2^2} = \sum_{k=0}^{p-1} \frac{2^k}{k!} \left[e^{-\|x_i-x_*\|^2/h_2^2} \left(\frac{x_i - x_*}{h_2} \right)^k \right] \left[e^{-\|y_j-x_*\|^2/h_2^2} \left(\frac{y_j - x_*}{h_2} \right)^k \right] + error_p.$$

where,

$$error_p \leq \frac{2^p}{p!} \left(\frac{\|x_i - x_*\|}{h_2} \right)^p \left(\frac{\|y_j - x_*\|}{h_2} \right)^p e^{-(\|x_i-x_*\|-\|y_j-x_*\|)^2/h_2^2}.$$

Factorization of the Hermite

$$H_r\left(\frac{y_j - x_i}{h_1}\right) = \sum_{l=0}^{\lfloor r/2 \rfloor} \sum_{m=0}^{r-2l} a_{lm} \left(\frac{x_i - x_*}{h_1}\right)^m \left(\frac{y_j - x_*}{h_1}\right)^{r-2l-m}$$

where,

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

Ignore the error terms and regroup

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

where

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

- The coefficients B_{km} can be evaluated separately in $\mathcal{O}(prN)$.
- Evaluation of $\widehat{G}_r(y_j)$ at M points is $\mathcal{O}(pr^2M)$.
- Hence the computational complexity has reduced from the quadratic $\mathcal{O}(rNM)$ to the linear $\mathcal{O}(prN + pr^2M)$.

Other tricks

- Space subdivision.

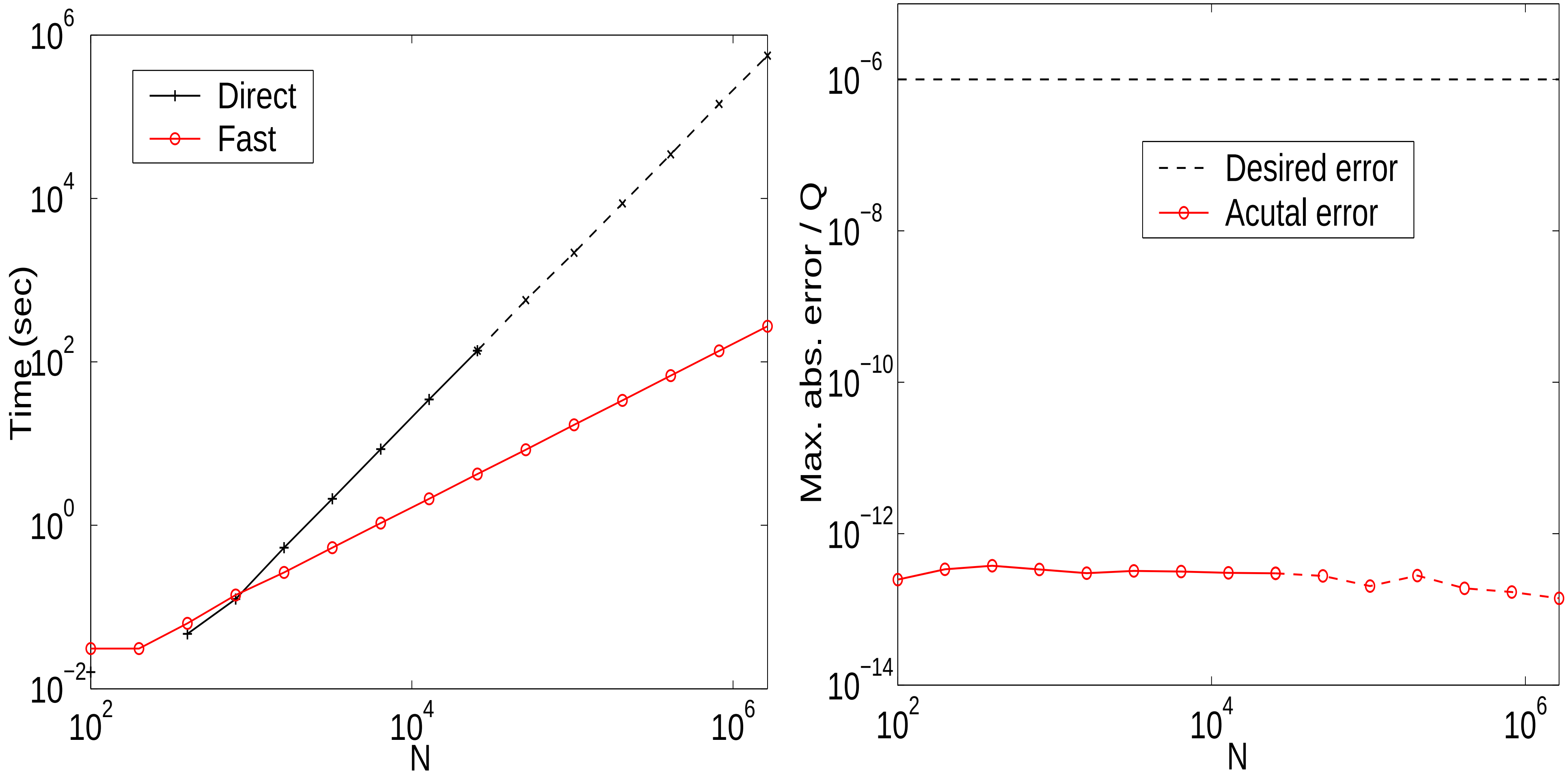
- Rapid decay of the Gaussian.

- Choosing p based on tight error bounds.

Numerical Experiments

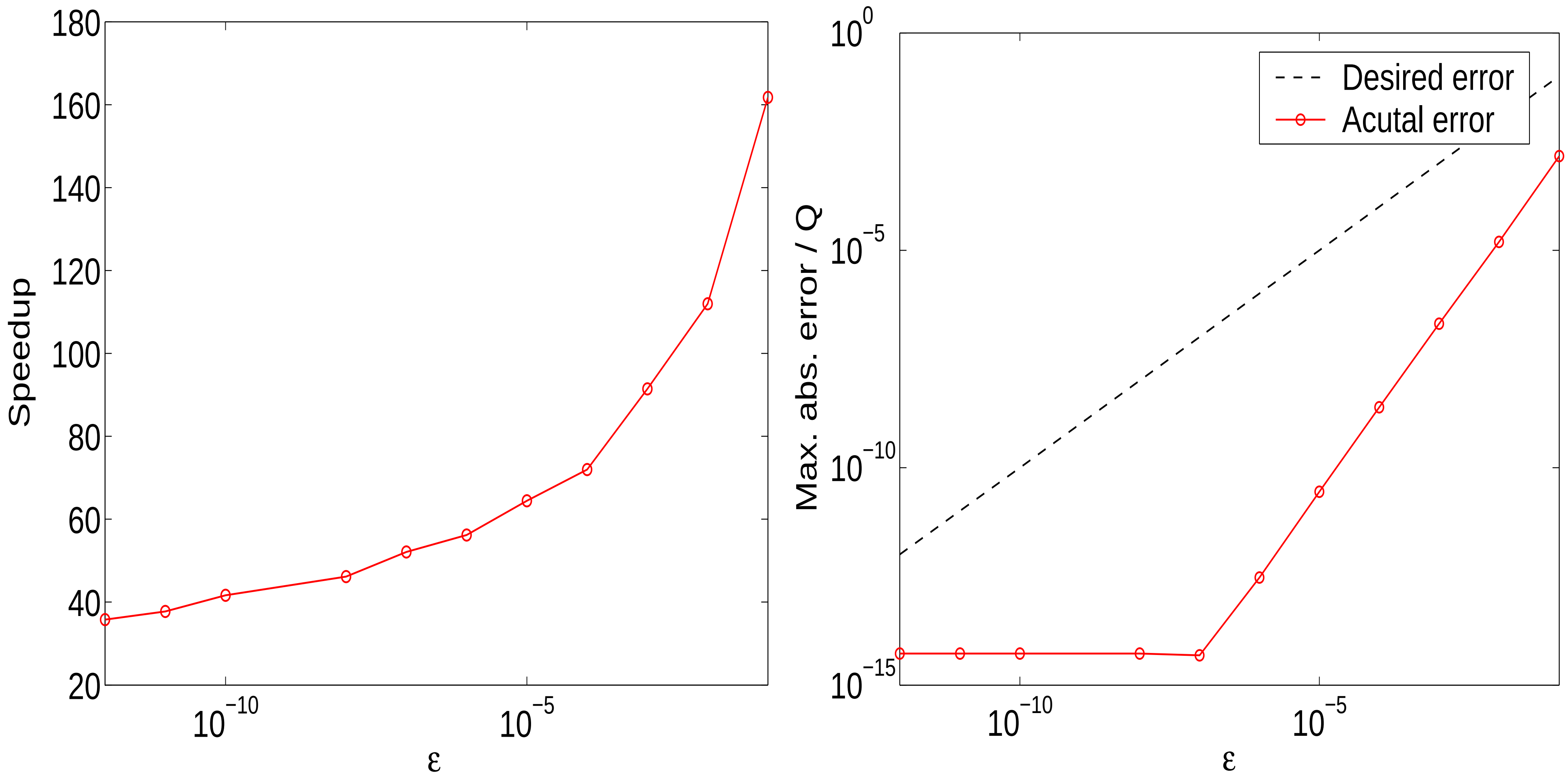
- Algorithm programmed in C++ with MATLAB bindings.
- Experiments run on 2.4 GHz processor with 2 GB RAM.
- Source and target points uniformly distributed in the unit interval.

As a function of N [$M = N$ $h = 0.1$ $r = 4$]



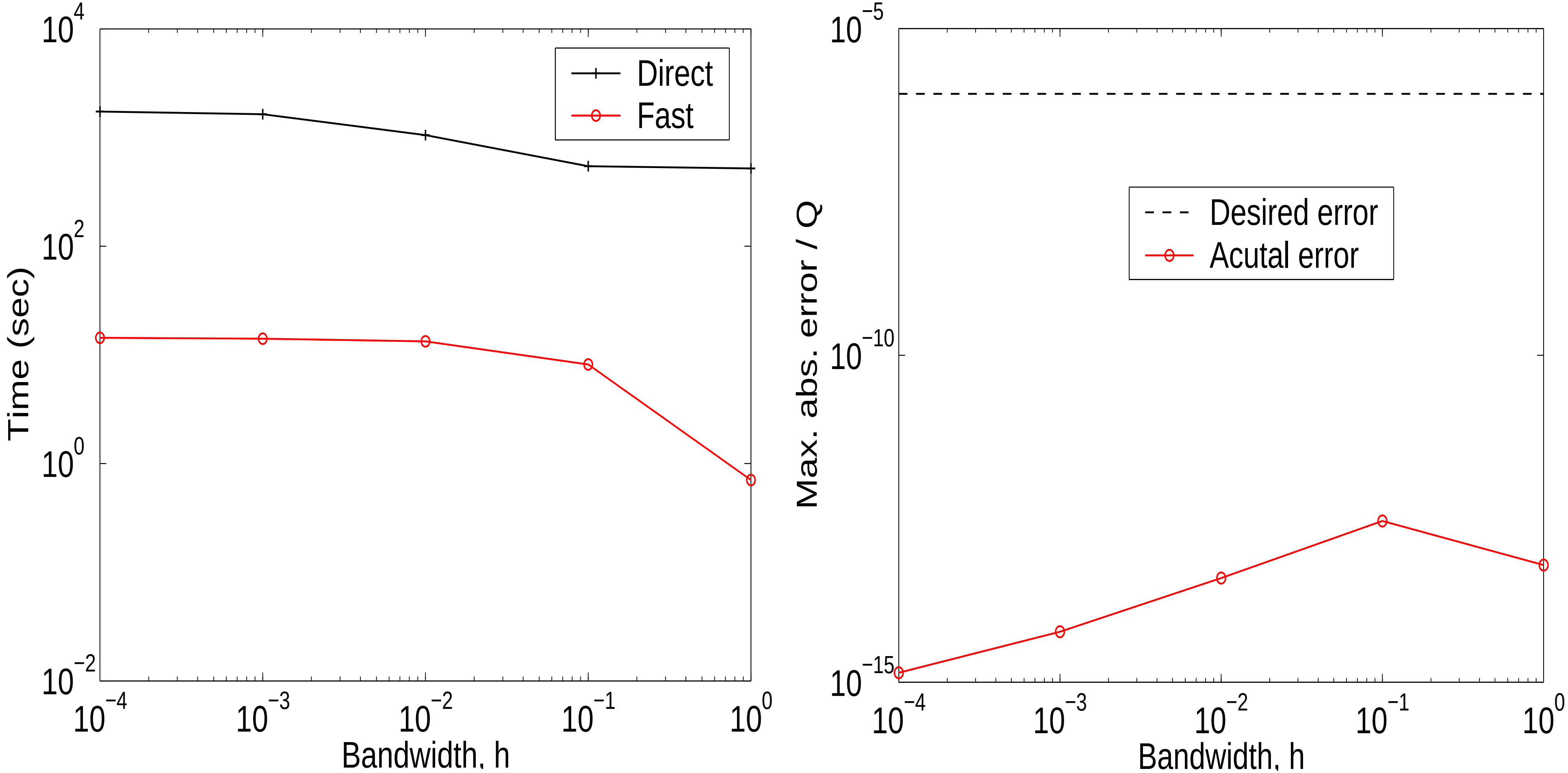
Linear in N.

Precision Vs Speedup [$M = N = 50,000$ $h = 0.1$ $r = 4$]



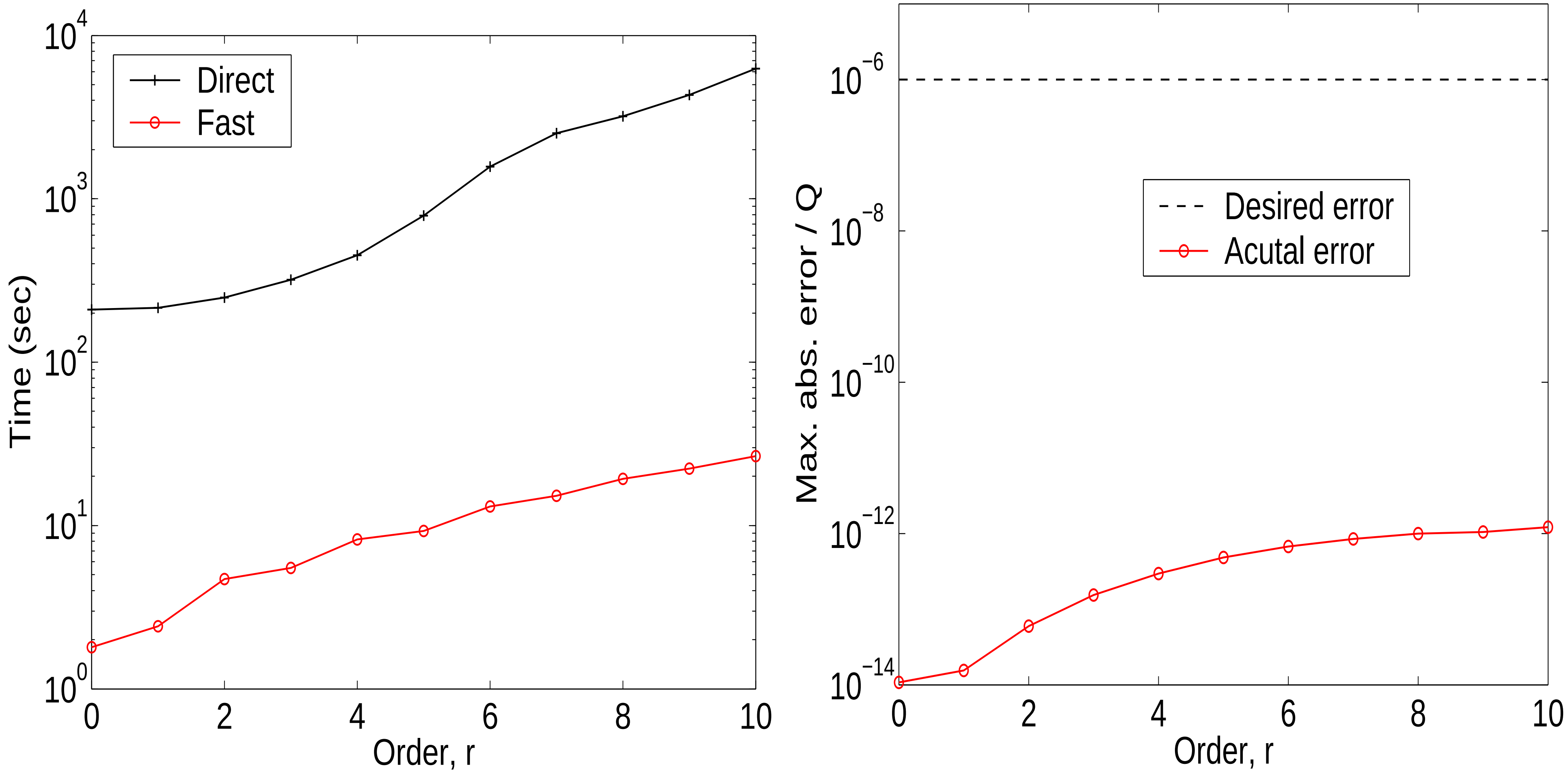
Better speedup for reduced precision.

As a function of bandwidth h [$M = N = 50,000$ $r = 4$]



Better speedups for large bandwidths.

As a function of r $[M = N = 50,000 \ h = 0.1]$



Speedup for bandwidth estimation

- Used the solve-the-equation plug-in method of Jones at.al (1996) *.

- We demonstrate the speedup achieved on the mixture of normal densities used by Marron and Wand (1992).

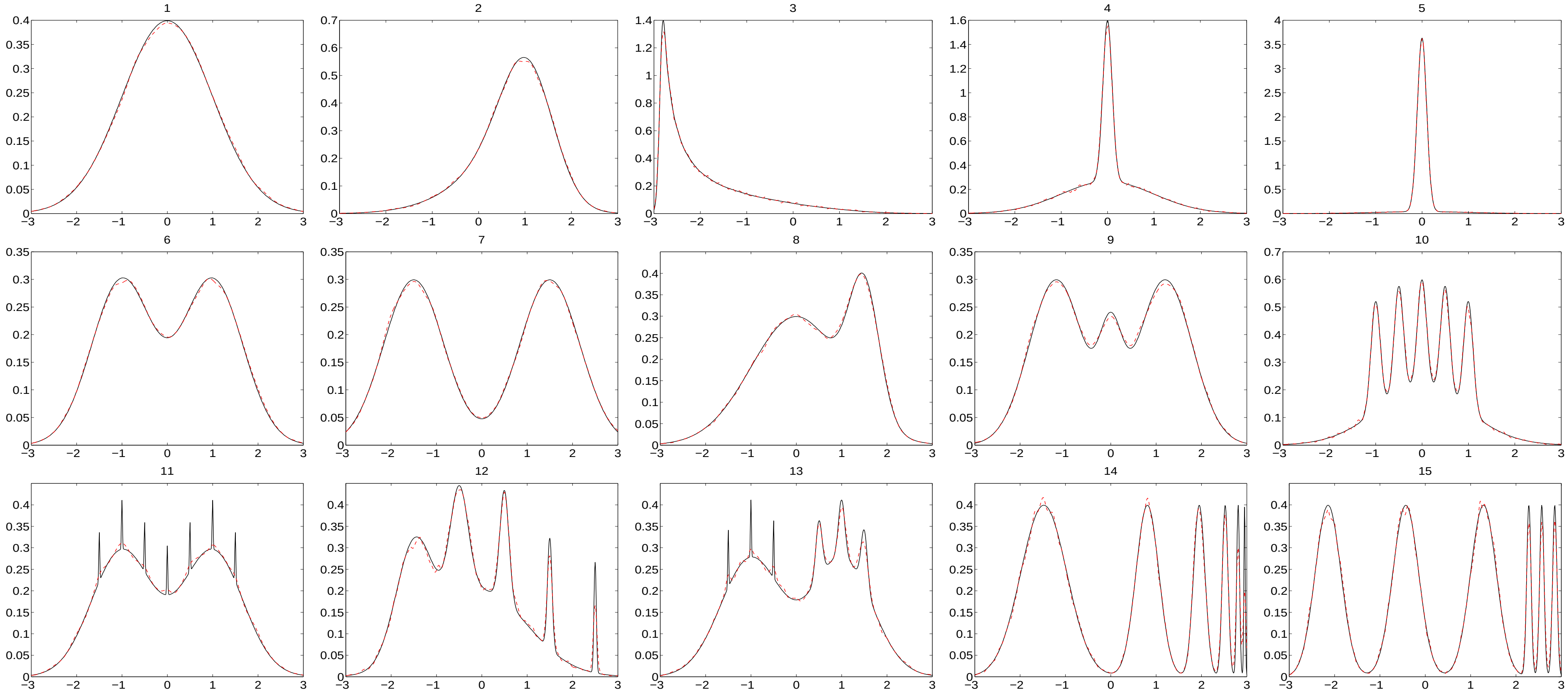
  - A typical representative of the densities likely to be encountered in real data situations.

- The absolute relative error is defined as $\frac{|h_{direct} - h_{fast}|}{h_{direct}}$.

- For $50,000$ points we obtained speedups in the range 65 to 105 with the absolute relative error of the order 10^{-5} to 10^{-7}.

*Sheather, S. and Jones, M. 1991. A reliable data-based bandwidth selection method for kernel density estimation. Journal of Royal Statistical Society Series B 53, 683-690.

Marron Wand normal mixtures *



*Marron, J. S. and Wand, M. P. 1992. Exact mean integrated squared error. The Annals of Statistics 20, 2, 712-736.

Speedup for Marron Wand normal mixtures

	h_{direct}	h_{fast}	T_{direct} (sec)	T_{fast} (sec)	Speedup	Rel. Err.
1	0.122213	0.122215	4182.29	64.28	65.06	1.37e-005
2	0.082591	0.082592	5061.42	77.30	65.48	1.38e-005
3	0.020543	0.020543	8523.26	101.62	83.87	1.53e-006
4	0.020621	0.020621	7825.72	105.88	73.91	1.81e-006
5	0.012881	0.012881	6543.52	91.11	71.82	5.34e-006
6	0.098301	0.098303	5023.06	76.18	65.93	1.62e-005
7	0.092240	0.092240	5918.19	88.61	66.79	6.34e-006
8	0.074698	0.074699	5912.97	90.74	65.16	1.40e-005
9	0.081301	0.081302	6440.66	89.91	71.63	1.17e-005
10	0.024326	0.024326	7186.07	106.17	67.69	1.84e-006
11	0.086831	0.086832	5912.23	90.45	65.36	1.71e-005
12	0.032492	0.032493	8310.90	119.02	69.83	3.83e-006
13	0.045797	0.045797	6824.59	104.79	65.13	4.41e-006
14	0.027573	0.027573	10485.48	111.54	94.01	1.18e-006
15	0.023096	0.023096	11797.34	112.57	104.80	7.05e-007

Projection pursuit

The idea of projection pursuit is to search for projections from high- to low-dimensional space that are most *interesting* *.

1. Given N data points in a d dimensional space project each data point onto the direction vector $a \in \mathbf{R}^d$, i.e., $z_i = a^T x_i$.

2. Compute the univariate nonparametric kernel density estimate, $\widehat{p}$, of the projected points z_i.

3. Compute the projection index $I(a)$ based on the density estimate.

4. Locally optimize over the the choice of a, to get the *most interesting* projection of the data.

*Huber, P. J. 1985. Projection pursuit. The Annals of Statistics 13, 435-475.

Projection index

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.

Speedup

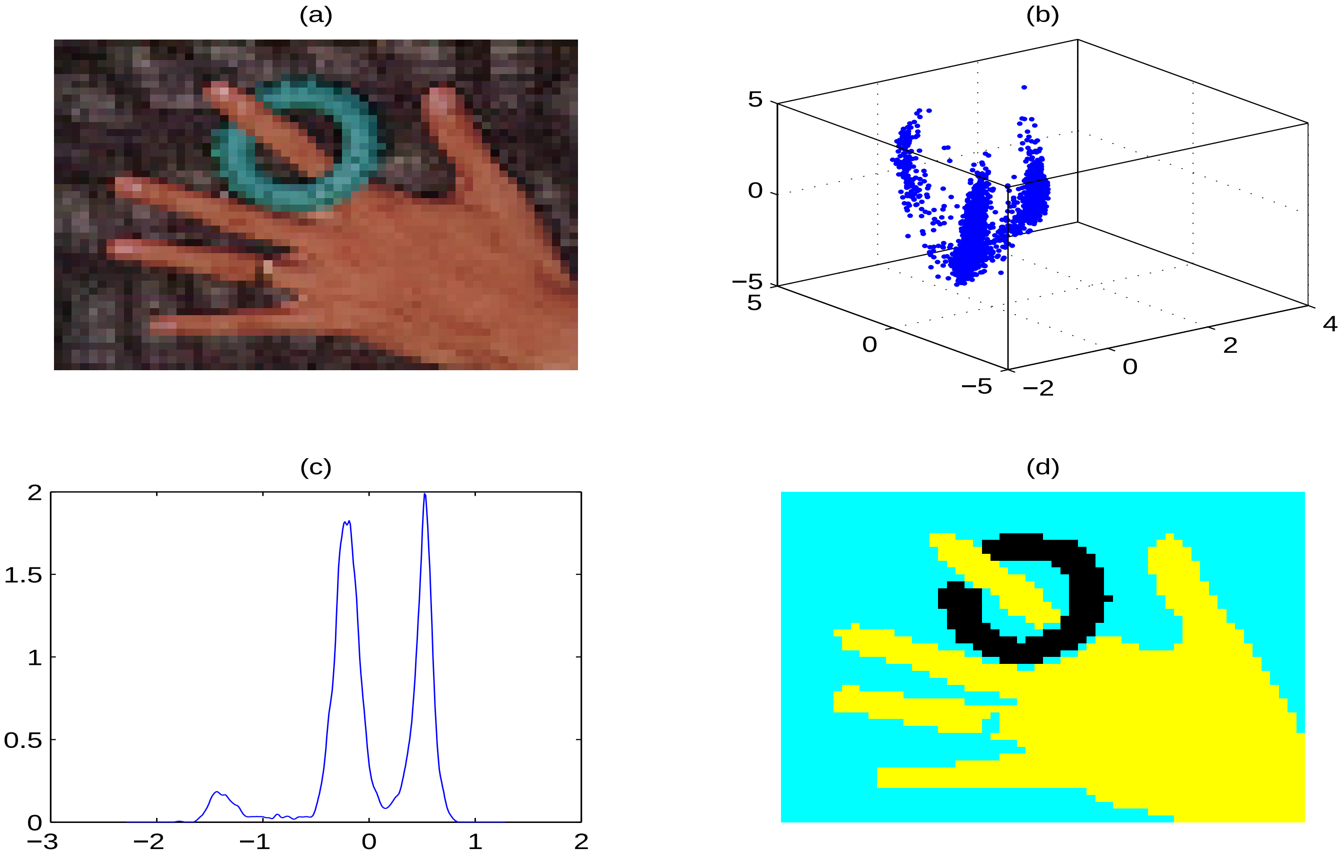
The computational burden is reduced in the following three instances.

1. Computation of the kernel density estimate (i.e. use the fast method with $r = 0$).

2. Estimation of the optimal bandwidth.

3. Computation of the first derivative of the kernel density estimate, which is required in the optimization procedure.

Projection pursuit on a image



The entire procedure took 15 minutes while that using the direct method takes around 7.5 hours.

Conclusions

- 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.

- We demonstrated how to potentially speedup the projection pursuit algorithm.

Software

- The code is available for academic use.
- www.cs.umd.edu/~vikas
- A detailed version of this paper is available as a TR *.

*Very fast optimal bandwidth selection for univariate kernel density estimation. Vikas C. Raykar and R. Duraiswami, CS-TR-4774, Department of computer science, University of Maryland, Collegepark.

Related work

- FFT *, FGT †, IFGT ‡, dual-tree §. All the above methods are designed to specifically accelerate the KDE.

- The main contribution of this paper is to accelerate the kernel density derivative estimate with an emphasis to solve the optimal bandwidth problem. The case of KDE arises as a special case of $r = 0$, i.e., the zero order density derivative.

[*] Silverman, B. W. 1982. Algorithm AS 176: Kernel density estimation using the fast Fourier transform. Journal of Royal Statistical society Series C: Applied statistics 31, 1, 93-99.

[†] Greengard, L. and Strain, J. 1991. The fast Gauss transform. SIAM Journal of Scientic and Statistical Computing 12, 1, 79-94.

[‡] Yang, C., Duraiswami, R., Gumerov, N., and Davis, L. 2003. Improved fast Gauss transform and efficient kernel density estimation. In IEEE International Conference on Computer Vision. 464-471.

[§] Gray, A. G. and Moore, A. W. 2003. Nonparametric density estimation: Toward computational tractability. In SIAM International conference on Data Mining.