Adaptive Kernel Regression for Image Processing and Reconstruction

Peyman Milanfar*
EE Department
University of California, Santa Cruz

*Joint work with Sina Farsiu, Hiro Takeda

AFOSR Sensing Program Review, June 21, 2007
Summary

• **Motivation:**
  – Existing methods for processing, reconstruction, and interpolation of spatial data are too strongly dependent upon underlying assumptions about signal and noise models.
  – Develop “universal” and robust methods based on adaptive nonparametric statistics for processing and reconstruction of image and video data.

• **Goal:**
  – Demonstrate the applicability of the Kernel Regression framework to a wide class of problems, producing algorithms competitive with state of the art.
Outline

• **Introduction and Motivation**
• Classic Kernel Regression
• Data-Adaptive Kernel Regression
• Experiments
• Conclusion
Data Sampling Scenarios

- Full samples: Denoising
- Incomplete samples: Denoising + interpolation
- Irregular samples: Denoising + reconstruction

Want a unified method for treating all these scenarios


Milanfar et al. EE Dept, UCSC
Outline

• Introduction and Motivation
• **Classic Kernel Regression**
• Data-Adaptive Kernel Regression
• Experiments
• Conclusion
Kernel Regression Framework

- The data model

\[ y_i = z(x_i) + \varepsilon_i, \quad i = 1, 2, \ldots, P \]

- The specific form of \( z(x) \) may remain unspecified.

Zero-mean, i.i.d noise (No other assump.)
Local Approximation in KR

• The data model

\[ y_i = z(x_i) + \varepsilon_i, \quad i = 1, 2, \ldots, P \]

• Local representation (N-terms Taylor expansion)

\[
z(x_i) = z(x) + \nabla z(x)^T (x_i - x) + \frac{1}{2!} (x_i - x)^T \{ \mathcal{H} z(x) \} (x_i - x) + \cdots
\]

\[
= \beta_0 + \beta_1^T (x_i - x) + \beta_2^T \text{vech} \left\{ (x_i - x) (x_i - x)^T \right\} + \cdots,
\]

• Note
  – We really only need to estimate the first unknown, \( \beta_0 \).
  – Other localized representations are also possible.
**KR Optimization Problem**

- The local representation is valid when $x_i$ and $x$ are close.
- We should give a higher weight to nearby samples

\[
\min_{\{\beta_n\}_{n=0}^N} \sum_{i=1}^P \left[ y_i - \beta_0 - \beta_1^T (x_i - x) - \beta_2^T \text{vech} \left\{ (x_i - x)(x_i - x)^T \right\} - \cdots \right] K_n(x_i - x)
\]

- The regression order
- N+1 terms
- This term gives the estimated pixel value at $x$
- The choice of the kernel function is open. e.g. Gaussian

→ Weighted Least Squares

Milanfar et al. EE Dept, UCSC
Classic Kernel Regression as a Locally Linear Filter

- The Net Effect of Classic $L_2$ KR

$$\hat{z}(x) = \hat{\beta}_0 = \sum_i W_{\mathcal{H}}(x_i, x, N) y_i$$

Equivalent kernel
Moving Beyond “Classical”

• Four factors to be considered
  – Kernel function ($K$):
  – The Regression order ($N$): Bias/Variance tradeoff
  – Basis Function: Different polynomial bases, wavelets etc., over-complete dictionaries
  – Smoothing matrix ($H$):
    • Data-adaptive choice will produce locally nonlinear filters with superior performance
Outline

• Introduction and Motivation
• Classic Kernel Regression
• Data-Adaptive Kernel Regression
• Experiments
• Conclusion
Data-Adapted Kernels

\[
\min_{\{\beta_n\}_{n=0}^{N}} \sum_{i=1}^{P} \left| y_i - \beta_0 - \beta_1^T (x_i - x) - \beta_2^T \text{vech} \left\{ (x_i - x) (x_i - x)^T \right\} - \cdots \right|^{m} K_h(x_i - x)
\]

\[
\min_{\{\beta_n\}_{n=0}^{N}} \sum_{i=1}^{P} \left| y_i - \beta_0 - \beta_1^T (x_i - x) - \beta_2^T \text{vech} \left\{ (x_i - x) (x_i - x)^T \right\} - \cdots \right| \cdot K_h(x_i - x, y_i - y)
\]

Milanfar et al. EE Dept, UCSC
A Special Case: The “Separable” Kernel

- Consider the denoising problem with regularly sampled data

\[
K_{\text{bilat}}(x_i - x, y_i - y) = K_{h_s}(x_i - x) \cdot K_{h_r}(y_i - y)
\]

- Special case: \(N = 0, m=2\)

\[
\hat{z}_{\text{bilat}}(x) = \sum_{i=1}^{P} W_i(x, K_{h_s}, K_{h_r}, N=0) y_i
\]

\[
= \frac{1}{P} \sum_{i=1}^{P} K_{h_s}(x_i - x) K_{h_r}(y_i - y) y_i
\]

With Gaussian Kernels, this is just the Bilateral filter! (Tomasi, et al. ‘98)

\[N>0 \text{ can generalize the bilateral filter!}\]

Milanfar et al. EE Dept, UCSC
Bilateral Kernel is Unstable

Essentially noiseless case

Fairly Noisy case

Milanfar et al. EE Dept, UCSC
Steering Kernel Method

\[ K_{\text{steer}}(x_i - x, y_i - y) = K_{H_i}(x_i - x) \]

Local dominant orientation estimate based on local gradient covariance

\[ K_{H_i}(x_i - x) = \frac{\sqrt{\det(C_i)}}{2\pi h^2} \exp \left\{ -\frac{1}{2h^2}(x_i - x)^T C_i(x_i - x) \right\} \]

\[ H_i = h C_i^{-\frac{1}{2}} \]

Milanfar et al. EE Dept, UCSC
Better Approach: Parameterization

\[ C_i = \gamma_i U_{\theta_i} \Lambda_i U_{\theta_i}^T \]

- Scaling parameter
- Elongation matrix
- Rotation matrix
Steering Kernel is Stable!

- (a) Texture
- (b) Flat
- (c) Strong edge
- (d) Corner
- (e) Weak edge

Essentially noiseless case

Fairly Noisy case

Milanfar et al. EE Dept, UCSC
The Net Effect of Kernel Regressors in $L_2$

**Classic Kernel**: Locally Linear Filtering:

$$\hat{z}(x) = \hat{\beta}_0 = \sum_i W(x_i, x, N) y_i$$

Depends only on $x-x_i$

**Steering Kernel**: Locally Non-Linear Filtering:

$$\hat{z}(x) = \hat{\beta}_0 = \sum_i W(x_i, x, y_i, y, N) y_i$$

Depends on both $x-x_i$ and $y-y_i$
Iterated Application of Steering

- Steering kernel $K_{steer}(x_i - x_j)$

Given data

- Direction
- Strength

Orientation estimation

Steering kernel regression

Iteration!

Iterative Algorithm

**Initialize:**

Noisy data $y$ → Initial Gradient Est. → Smoothing Matrix Est. → $H^0_i$ → Kernel Reg. → $\hat{\beta}_1^0$ → $\hat{z}^0$

**Iterate:**

$\hat{\beta}_1^u$ → Smoothing Matrix Est. → $H^u_i$ → Kernel Reg. → $\hat{\beta}_1^{u+1}$ → $\hat{z}^{u+1}$

Milanfar et al. EE Dept, UCSC
Outline

• Introduction and Motivation
• Classic Kernel Regression
• Data-Adaptive Kernel Regression
• Experiments
• Conclusion
Gaussian Noise Removal (STD=25)

Original image  Noisy image, RMSE=24.87 (SNR=5.69dB)

Milanfar et al. EE Dept, UCSC
Gaussian Noise Removal (STD=25)

Portilla and Simoncelli, RMSE=6.64
Bayes Least Square-Gaussian Scale Mixture (2003)

Iterative steering kernel regression, N=2, RMSE=6.63

Kervrann, et al. RMSE=6.61
Optimal spatial adaptation (2006)

Buades, et al. Non-Local Means, RMSE=8.45
Gaussian Noise Removal (STD=25)

Portilla and Simoncelli, RMSE=6.64
Bayes Least Square-Gaussian Scale Mixture (2003)

Kervrann, RMSE=6.61
Optimal spatial adaptation (2005)

Iterative steering kernel regression, N=2, RMSE=6.63

Non-Local Means, RMSE=8.45

Milanfar et al. EE Dept, UCSC
Compression Artifact Removal (Deblocking)

Original pepper image

JPEG with quality of 10, RMSE=9.76
Compression Artifact Removal (Deblocking)

Portilla and Simoncelli, RMSE=8.80
Bayes Least Square-Gaussian Scale Mixture (2003)

Kervrann, RMSE=8.85
Optimal spatial adaptation (2005)

Iterative steering kernel regression, N=2, RMSE=8.48

Non-Local Mean, RMSE=8.64
Film Grain Reduction (Real Noise)

Real noisy image
Film Grain Reduction (Real Noise)

Portilla and Simoncelli, Bayes Least Square-Gaussian Scale Mixture, Denoising on YCrCb channels

Kervrann, Optimal spatial adaptation, Denoising on YCrCb channels

Iterative steering kernel regression, N=2, Denoising on YCrCb channels

Non-Local Mean, Denoising on YCrCb channels

Milanfar et al. EE Dept, UCSC
Comparison in Absolute Residual Image in the Y channel

Portilla and Simonceli, Bayes Least Square-Gaussian Scale Mixture, Denoising on YCrCb channels

Kervrann, Optimal spatial adaptation, Denoising on YCrCb channels

Iterative steering kernel regression, Denoising on YCrCb channels

Non-Local Mean, Denoising on YCrCb channels
Real Noise Reduction

Real noisy image
Real Noise Reduction

Portilla and Simoncelli, Bayes Least Square-Gaussian Scale Mixture, Denoising on YCrCb channels

Kervrann, Optimal spatial adaptation, Denoising on YCrCb channels

Iterative steering kernel regression, N=2, Denoising on YCrCb channels

Non-Local Mean, Denoising on YCrCb channels
What about SKR for *Interpolation*?

**The Problem**: At the location \( x \) where we wish to interpolate, there is no pixel value (yet)

**The Solution**: Produce a “pilot”, low-complexity, estimate of the pixel ? then apply the more sophisticated adaptive kernel techniques described earlier.
Illustration:

30% of Pixels Retained

Local Constant Kernel Estimate

Steering Kernel Estimate
Interpolation of Irregular Samples

Irregularly down-sampled image (Randomly delete 85% of pixels)

Iterative steering kernel regression, N=2, RMSE=8.48

Delaunay-spline smoother, RMSE=9.05

Classic kernel regression, N=2, RMSE=9.69
Deblurring

- Taking the PSF into account:

\[ y_i = z(x_i) + \varepsilon_i = g(x_i) \ast u(x_i) + \varepsilon_i \]

- The model in matrix form, we have

\[ \mathbf{Y} = \mathbf{Z} + \varepsilon = \mathbf{GU} + \varepsilon \]

\[
\begin{bmatrix}
  u(x_1) \\
  \vdots \\
  u(x_p)
\end{bmatrix}
\xrightarrow{\text{Shift}}
\begin{bmatrix}
  u(x_1 + v) \\
  \vdots \\
  u(x_p + v)
\end{bmatrix} = \begin{bmatrix}
  \vdots \\
  \vdots \\
  \vdots
\end{bmatrix}
\]

The Taylor expansion at \( x_1 \)

The Taylor expansion at \( x_p \)

Milanfar et al. EE Dept, UCSC
Deblurring Example
PSF: 19x19 uniform, BSNR: 40dB

The original image

The degraded image
RMSE = 29.67
Deblurring Example
PSF:19x19 uniform, BSNR:40dB

ForWaRD
RMSE = 14.37

LPA-ICI
RMSE = 13.36

Data-adaptive kernel reg.
RMSE = 14.21

Milanfar et al. EE Dept, UCSC
Super-resolution Example

Resolution enhancement from video frames captured by a commercial webcam (3COM Model No.3719)
Conclusion

• Adaptive kernel regression is a powerful tool for a wide class of problems.

• It provides results competitive with state of the art in
  – denoising,
  – interpolation,
  – deblurring,
  – resolution enhancement.

• Don’t try this at home.

Other examples and software are available at http://www.soe.ucsc.edu/~htakeda