BLIND IMAGE DECONVOLUTION USING SUPPORT VECTOR REGRESSION

Dalong Li, Russell M. Mersereau

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
{dalong,rmm}@ece.gatech.edu

Steven Simske

Imaging Systems Laboratory
Hewlett-Packard Laboratories
Fort Collins, CO 80528
steven.simske@hp.com

ABSTRACT

This paper describes an algorithm for the restoration of a noisy blurred image based on support vector regression. The blind image deconvolution was formulated as a machine learning problem. From the training set, the mapping between the noisy blurred image and the original image are learned by support vector regression (SVR). With the acquired mapping, the degraded image can be restored. Our approach was experimentally compared with the adaptive Lucy-Richardson maximum likelihood (ML) algorithm. In terms of ISNR (Improvement of Signal to Noise Ratio), SVR outperformes ML in blind deblurring tests in which the types of blurs, point spread function (PSF) support, and noise energy are all unknown.

1. INTRODUCTION

The primary objective of image restoration is to restore the visual quality of a degraded image. It has important applications including photographic deblurring, remote sensing and medical imaging. Blurring may be caused by degradations in the imaging process such as lens defocusing, atmospheric turbulence, object motion, or diffraction.

In many imaging applications, an observed discrete image $g(x, y)$ can be approximated as the sum of a two-dimensional convolution of the true image $f(x, y)$ with a linear shift invariant blur, also known as the point-spread-function (PSF), $h(x, y)$, and additive noise $\eta(x, y)$. That is

$$\begin{aligned} g(x,y) &= f(x,y) * h(x,y) + \eta(x,y) \\ &= \sum_{(n,m)} f(n,m)h(x-n, y-m) + \eta(x,y) \end{aligned}$$

$$x, y, n, m \in Z$$

in which * denotes the two-dimensional linear convolution operation, and Z is the set of the integers and η denotes the additive noise. The problem of recovering the true image $f(x, y)$ from the degraded image $g(x, y)$ is called image deconvolution or image restoration. Classical restorations require complete knowledge of the blur and a statistical description of the noise to be known prior to restoration. However, it is often impossible to determine these parameters a priori. This is due to a variety of practical constraints such as the difficulty of characterizing atmospheric turbulence in aerial imaging or the potential health hazard of using a stronger incident beam to improve the image quality in X-ray imaging. In these situations, blind image deconvolutions are essential to recover visual information.

Many blind restoration algorithms have been proposed in the past, as surveyed in [1]. The iterative blind deconvolution algorithm was first proposed by Ayers and Dainty in 1988 [2]. The double iteration algorithm developed by Holmes et al [3], was based on the EM algorithm. The EM algorithm is an efficient image restoration algorithm and has been widely used in many applications under different names such as “Lucy” and “Richardson” (LR), etc. [4, 5]. The original Lucy-Richardson algorithm is a nonlinear algorithm derived from Bayesian considerations, and based on the knowledge of the point spread function (PSF). This algorithm ensures the positivity of the image and the conservation of its total energy. Some recent work [6, 7] includes a variational approximation for Bayesian blind image deconvolution, which assumes that the PSF is partially known. Thus, a probabilistic law relating the observations and the quantities to be estimated can be formulated.

Blurs can be quite different mathematically in terms of their coefficients and the PSF support and different applications might dictate that the noise energy might be quite different. However, they all share some common characteristics. For example, they are all effectively low pass filters. Our approach for blind image deblurring seeks to extract the commonality behind the seemly diverse blurs. Once we can acquire that knowledge, we might be able to establish a common usable framework that can handle different blurs with different PSF supports. In this way, the blur identification is side-stepped. We use support vector regression [8] to obtain an optimized mapping that takes an ensemble of degraded images to the true images in the training phase; then in the test phase, the learned mapping is used to perform the blind image deconvolution.

The paper is organized as follows. In Section 2, linear regression is first reviewed followed by a discussion of support vector regression and the details of our algorithm. In Section 3, the implementations and the numerical experiments of our approach are reported. The performance is evaluated using SNR. Conclusions are discussed in Section 4.

2. SVR BASED IMAGE DEBLURRING

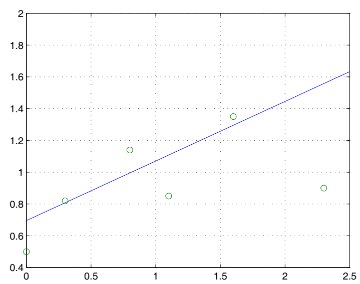


Fig. 1. Linear regression.

Given training data $(X_1; y_1), \ldots, (X_l; y_l)$ in Fig. 1, where X_i are input vectors and y_i are the associated output value of X_i, traditional linear regression finds a linear function $W^T X + b$ such that $(W; b)$ is an optimal solution of

$$\min_{w,b} \sum_{i=1}^{l} (y_i - (W^T X_i + b))^2 \tag{1}$$

In other words, $W^T X + b$ approximates training data by minimizing the sum of square errors. Usually n, the dimension of X is less than l, the number of the points. Otherwise, a line passes through all the points and overfitting occurs.

If the data is nonlinearly distributed, a linear function will not work. In these cases, we can map the data to a higher dimensional space by a function $\phi(x)$ as in support vector classification. To avoid overfitting, some modifications are adopted. Support vector regression solves the following modified optimization problem:

$$\min_{W,b,\xi,\xi^*} \frac{1}{2} W^T W + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \tag{2}$$

$$\text{subject to} \quad \begin{aligned} & y_i - (W^T \phi(X_i) + b) \leq \epsilon + \xi_i, \\ & (W^T \phi(X_i) + b) - y_i \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, i = 1, \ldots, l \end{aligned}$$

ξ_i is the upper training error (ξ_i^* is the lower training error) subject to the $\epsilon-$insensitive tube $|y - (W^T \phi(X) + b)| \leq \epsilon$, and ϵ is a threshold.

For blind image deconvolution, the $2n + 1$ by $2m + 1$ neighborhood of the blurred image pixel $g(x, y)$ is:

$$\begin{pmatrix} g(x-n, y-m) & \ldots & g(x-n, y+m) \\ \ldots & g(x, y) & \ldots \\ g(x+n, y-m) & \ldots & g(x+n, y+m) \end{pmatrix}.$$

The matrix is converted either by row or by column into a vector. For example, $(g(x-n, y-m), \ldots, g(x-n, y+m), \ldots, g(x, y), \ldots, g(x+n, y-m), \ldots, g(x+n, y+m))$ (row by row). This vector becomes X for the support vector regression. y_i is the corresponding pixel of $f(x, y)$. By shifting this sampling window over all positions of the ensemble of the blurred images, we can obtain the training set for the support vector regression.

When features are in different numerical ranges, those in larger ranges may dominate the others. Thus, a proper scaling of the features before training can be very important.

3. EXPERIMENTS

3.1. SVR set up

As mentioned in the previous section, scaling is a very important preprocessing. In all the experiments, the images are scaled into range [0,1] before they are blurred. The original LENA image (512 by 512) is blurred with moving average filters with different support (3 by 3 and 4 by 4). Gaussian white noise is added to the two blurred images to obtain the noisy blurred images used for training of the SVR. The SNR of the noisy blurred images are 12.09 and 11.91 respectively. The LibSVM software package [9] is used with a radial basis function kernel to find $\phi(x)$.

In our experiment, $n = 3, m = 3$. Thus the size of the sampling window is 7 by 7. The length of the vector is 49. The 7 by 7 sampling window is shifted over the blurred images. The training sets are constructed from the pairs $(X; f(x, y))$. Since a low contrast patch has less of a contribution for the SVR to find the mapping between the blurred image and the true image, we only select patches whose contrast (variance) is above a certain threshold. To further speed up the training, the training set is randomly downsampled to 39 % of the original training set.

3.2. Robustness to untrained blurs

The rectangular averaging, Gaussian ($\sigma = 1$), motion blurs, circular averaging filters, and Gaussian noise are applied to the cameraman image (256 by 256) to make the test images. None of these filters is identical to the ones in the training set.

Table 1. ISNR (db) comparison of PCA and ML for different blurs and noise

Blurs	SVR	ML
Circular average ($\gamma = 3$,SNR=12.55)	2.34	-0.59
Motion ($L = 5, \Theta = 0$,SNR=11.88)	2.63	-0.54
Gaussian (5×5,SNR=13.58)	1.96	-1.64
Average (4×3,SNR=13.23)	2.29	-1.42

We use signal to noise ratio (**SNR**) to measure the degradation caused by blur and noise. SNR is defined as:

$$SNR = 10 \log_{10} \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (f(i,j))^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} (f(i,j) - g(i,j))^2} \quad (3)$$

The well established measurement of restoration quality is the Improvement in Signal to Noise Ratio of the image (**ISNR**). The formula for ISNR is then given as

$$ISNR = 10 \log_{10} \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (f(i,j) - g(i,j))^2}{\sum_{i=1}^{M} \sum_{j=1}^{N} (f(i,j) - \hat{f}(i,j))^2} \quad (4)$$

The trained $\phi(X)$ is used to perform the deconvolution of the the images. For comparison purpose, the images are also deblurred using the Lucy-Richardson maximum likelihood (ML) algorithm. The ML algorithm requires an initial guess of the blur PSF. When the PSF is specified, its size and the values it contains must be estimated. The size has been observed to be more important to the ultimate success of the restoration than the actual values in the PSF. Since 3 by 3 and 4 by 4 averaging filters are used to train SVR, the initial PSF are specified as a 3 by 3 averaging filter and a 4 by 4 averaging filter. The better results of the two are used in the comparison against SVR. The performance of the ML is very sensitive to the number of iterations. An improper number of iteration leads to over-convergence, and too many artifacts in the image. In all the experiments, when the default iteration number (10) gives a bad result, we manually choose a different iteration number for ML to produce a better result. The results are summarized in the table 1.

The radius γ of the circular averaging filter is 3. The motion filter corresponds to a camera horizontal motion of 5 pixels. For all the experiments, SVR always improves the image quality in terms of ISNR; while the ISNR for the ML are always negative. Observe that none of the blurs used in the testing were used in the training. Although the last line in the table is averaging filter, the PSF support is different. This experiment highlights an advantage of our proposed algorithm: SVR can perform blind deconvolution in the sense that the blurring filters do not appear in the training set and thus they do not need to be known. The knowledge about the PSF support of the blurring filter is needed to make ML work well. Also ML degrades in the presence of noise.

3.3. Real image example

Fig. 3 shows the comparative results on a real motion blurred image. The blur was caused by nonlinear motion of the object. Noise is obviously amplified in the ML deblurred image. SVR result is smoother than the ML result and it is sharper that the original image. Observe that the two results are quite different.

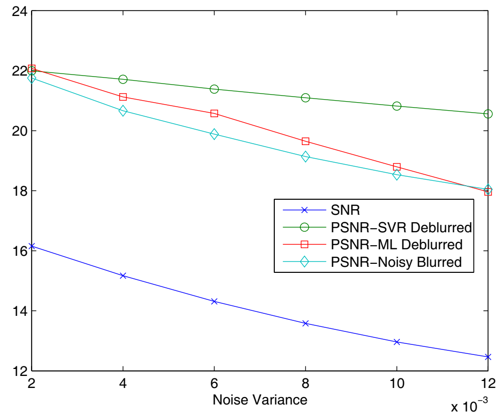
3.4. Robustness to noise

To test the robustness to noise, we conducted the following experiments: First, a blurred image is generated by convolving the cameraman image with the 5×5 Gaussian filter ($\sigma = 1$); then different levels of zero-mean Gaussian random noise are added to the blurred image. The noise variance changes from 0.004 to 0.014 in steps of 0.002. SNR is computed at each step to reflect the degradation. The training sets remains exactly the same as in the previous experiment. Figs. 2 (a,b) show the comparative results. Fig. 2a shows that as the variance of the noise increases, the PSNR of the SVR restored image drops slightly while that of the ML restored image drops dramatically. PSNR of the noisy blurred image also decreases as the noise variance increases. Fig. 2b shows the ISNR results. As the noise decreases, the performance of ML improves. However, SVR is robust to the change of the noise energy, which is important since the knowledge about noise may not be available. For the SVR restoration, the level of the noise does not affect the performance of SVR much.

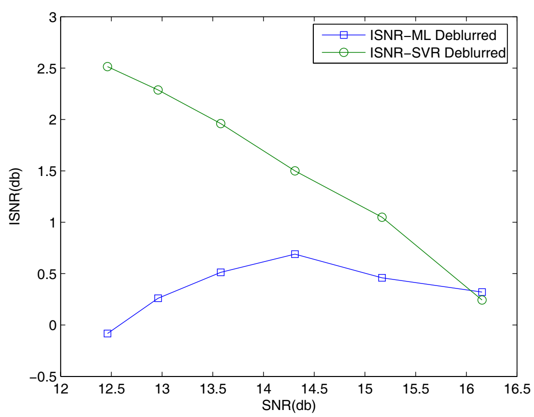
4. CONCLUSION

We have proposed a SVR based algorithm for the restoration of a noisy blurred image. SVR finds the optimal $\phi(X)$ from the training set, which is then used to perform the deconvolution. Some of the advantages of the proposed algorithm include the following:

- The algorithm can generalize to different types of blurs with different PSF support as well as varying noise.
- The algorithm is very robust to parameter selection. The only parameter is the size of the sampling window when the training set is made. It does not need to match the true blur PSF support. In our case, we use 7 by 7 for the sampling window, but the true PSF in the test images includes 4 by 3, 5 by 5 etc.
- Upgrading is easily done by simply including new examples.



(a) PSNR of ML and SVR for different noise level.



(b) ISNR comparison of ML and SVR for different SNRs.

Fig. 2. Noise robustness test

Fig. 3. Example of a real noisy blurred image. top: noisy blurred image; bottom left: SVR restored; bottom right: ML restored (Initial PSF Size 4×4, Iteration Number = 10).

5. REFERENCES

[1] D. Kundur and D. Hatzinakos, "Blind Image Deconvolution Revisited," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 61–63, Nov 1996.

[2] G. R. Ayers and J. C. Dainty, "Iterative Blind Deconvolution Method and Its Applications," *Optics Letters*, vol. 13, no. 7, pp. 547–549, July 1988.

[3] T. J. Holmes, "Blind Deconvolution Quantum-Limited Incoherent Imagery: Maximum-Likelihood Approach," *J. Opt. Soc. Am.*, vol. 9, pp. 1052 – 1061, 1992.

[4] W. H. Richardson, "Bayesian-based Iterative Method of Image Restoration," *J. Opt. Soc. Am*, vol. 62, pp. 55–59, 1972.

[5] A. K. Katsaggelos and K. T. Lay, "Maximum Likelihood Blur Identification and Image Restoration Using the EM Algorithm," *IEEE Transactions on Signal Processing*, vol. 39, pp. 729–733, March 1991.

[6] Ling Guan Kim-Hui Yap and Wanquan Liu, "A Recursive Soft-decision Approach to Blind Image Deconvolution," *IEEE Transactions on Signal Processing*, vol. 51, pp. 515–526, Feb. 2003.

[7] A.C. Likas and N.P. Galatsanos, "A Variational Approach for Bayesian Blind Image Deconvolution," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2222–2233, Aug. 2004.

[8] C. Cortes and V. Vapnik, "Support-vector Network," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[9] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a Library for Support Vector Machines*, 2001, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.