Image denoising using least squares wavelet support vector machines

Guoping Zeng (曾国平) and Ruizhen Zhao (赵瑞珍)

Institute of Information Science, Beijing Jiaotong University, Beijing 100044

Received May 14, 2007

We propose a new method for image denoising combining wavelet transform and support vector machines (SVMs). A new image filter operator based on the least squares wavelet support vector machines (LS-WSVMs) is presented. Noisy image can be denoised through this filter operator and wavelet thresholding technique. Experimental results show that the proposed method is better than the existing SVM regression with the Gaussian radial basis function (RBF) and polynomial RBF. Meanwhile, it can achieve better performance than other traditional methods such as the average filter and median filter.

OCIS codes: 100.3020, 100.2000, 100.5010, 100.7410.

Image denoising is used to improve the quality of an image corrupted by a lot of noises due to the imperfection of image acquisition systems and transmission channels. Having the property of multi-resolution, wavelet transform is one of the most widely used tools in image denoising[1−4]. Support vector machine (SVM), which is based on statistic learning theory, was proposed by Vapnik[5]. Lately, least squares support vector machine (LS-SVM)[6] is gaining more and more attention, mostly because it has some very attractive properties, regarding the implementation and the computational issues of teaching. In this paper, we combine LS-SVM and wavelet technique, and apply them in image denoising.

Suppose we have a training set $\{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in R^d$, $y_i \in R$, R^d represents the input space, d is the dimension. The basic approximation problem could be solved by finding a set of weights $\omega = \{\omega_1, \omega_2, \cdots, \omega_N\}$ with the function

$$y_i = f(x_i, \omega) = \sum_{i=1}^{N} \omega^* \phi_i(x) + b, \tag{1}$$

where $\phi(x)$ is the mapping function and b is a scalar.

The quadratic loss function is selected in LS-SVM. The following optimization problem of LS-SVM is formulated as

$$\text{minimize } J = \frac{1}{2}\|\omega\|^2 + \frac{1}{2}\gamma \sum_{i=1}^{N} \xi_i^2, \tag{2}$$

$$\text{subject to } y_i = \omega\phi(x_i) + b + \xi_i,\ i = 1, 2, \cdots, N, \tag{3}$$

where $\xi \in R^{N\times 1}$ is the error vector, γ is the regularization parameter.

In LS-SVM, one solves the constrained optimization problem (2) and (3) by constructing the Lagrangian:

$$L(\omega, b, \xi, \alpha, \gamma) = \frac{1}{2}\omega^{\mathrm{T}} \cdot \omega + \frac{1}{2}\gamma \sum_{i=1}^{N} \xi_i^2 - \sum_{i=1}^{N} \alpha_i (\omega\phi(x_i) + b + \xi_i - y_i), \tag{4}$$

where the parameters α_i $(i = 1, 2, \cdots, N)$ are the Lagrange multipliers, in which $\alpha_k \neq 0$ are called support vectors (SVs).

Then we define

$$\begin{cases} y = [y_1, y_2, \cdots, y_N]^{\mathrm{T}} \\ H = [1, 1, \cdots, 1]^{\mathrm{T}} \\ \alpha = [\alpha_1, \alpha_2, \cdots, \alpha_N]^{\mathrm{T}} \\ Z_{kh} = \phi(x_k)^{\mathrm{T}} \cdot \phi(x_h) \end{cases}, \tag{5}$$

where $H \in R^{N\times 1}$. Based on Mercer's condition, we define the kernel

$$K(x, x') = \phi(x)^{\mathrm{T}} \cdot \phi(x'), \quad k, h = 1, 2, \cdots, N. \tag{6}$$

Then we can obtain

$$\begin{bmatrix} 0 & H^{\mathrm{T}} \\ H & Z + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{7}$$

and the resulting LS-SVM model for function estimation becomes

$$f(x) = \sum_{k=1}^{N} \alpha_k K(x, x') + b, \tag{8}$$

where α, b are solutions of Eq. (7).

SVM uses SV kernel to map the data from the input space to a high-dimensional feature space in which we can process a problem in linear form. Zhang *et al.*[7] proposed a practical way to construct a wavelet kernel which combines wavelet technique with SVM using theoretic analysis. It showed the feasibility and validity of wavelet SVM in regression. The kernel, which satisfies the conditions of Mercer theorem, was defined as

$$K(x, x') = \prod_{i=1}^{N} h\left(\frac{x_i - x_i'}{a}\right) = \prod_{i=1}^{N} \left(\cos\left(1.75 \times \frac{(x_i - x_i')}{a}\right) \exp\left(-\frac{\|x_i - x_i'\|^2}{2a^2}\right)\right), \tag{9}$$

1671-7694/2007/110632-04

where a is a dilation coefficient, and $x, x' \in R^N$.

In this paper, we combine LS-SVM with the wavelet kernel described above, and we can get LS-WSVM. Then, the optimal decision function for regression can be obtained as

$$f(x) = \sum_{k=1}^{N} \alpha_k \prod_{i=1}^{N} h\left(\frac{x_i - x_i^k}{a_k}\right) + b, \tag{10}$$

where x_i^k is the ith component of the kth training sample, a_k $(k = 1, 2, \cdots, N)$ are the wavelet dilation coefficients.

Wavelet coefficients are correlated in a small neighborhood. A large wavelet coefficient probably has large coefficients at its neighbors. Cai[8] proposed an adaptive block thresholding algorithm in which the characteristics of the neighboring coefficients were considered to determine shrinkage thresholds. In this paper, we extend Cai's idea to the image situation, and use the new filter operator T which will be described below to modify the coefficient centered at the neighborhood window.

For each wavelet coefficient $d_{j,k}$ (j is the scale index, and k is the position index) of interest, we need to consider a neighborhood window around it. In previous experiment[9], the immediate neighborhood window sizes of 3×3 and 5×5 are good choices. Here we choose the diamond-like window illustrated in Fig. 1. On one hand, it has better effect than a 3×3 neighborhood window; on the other hand, it has higher calculation efficiency than a 5×5 neighborhood window. Besides, the size of diamond-like neighborhood can be changed at different levels. There are 13 coefficients in the diamond-like neighborhood window at the first level, and 5 coefficients at the next level.

A two-dimensional (2D) gray level image can be regarded as a continuous function $y = f(\overline{x}) : R^2 \to R^1$, where the input $\bar{x}$ is a 2D vector that equals the row and column indices of a pixel, and the output y is a scalar value indicating the gray level of that pixel. Based on the theory of LS-SVM for data regression, we regard the row and column indices of wavelet subband images as the input elements of LS-SVM, and the wavelet coefficients in the window as the output elements. As to the neighborhood of coordinates r, c, all the neighborhood elements are considered as the input vectors of LS-SVM, and then we can construct the nonlinear formula between input coordinate vectors and wavelet coefficients centered at the neighborhood window.

As illustrated by Liu *et al.*[10], with Eq. (7), we can get

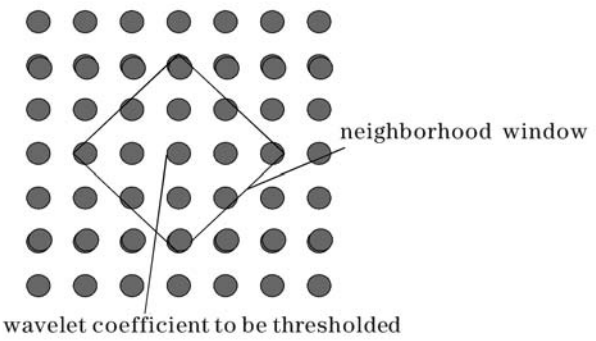

Fig. 1. Neighborhood window centered at the wavelet coefficient to be thresholded.

$$\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 & H^{\mathrm{T}} \\ H & Z + \gamma^{-1} I \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ y \end{bmatrix}. \tag{11}$$

We define $\Omega = Z + \gamma^{-1} I$, matrices A and B

$$\begin{cases} A = \Omega^{-1} \\ B = H^{\mathrm{T}} \Omega^{-1} / H^{\mathrm{T}} \Omega^{-1} H \end{cases}, \tag{12}$$

then Eq. (11) can be rewritten as

$$\begin{cases} b = By \\ \alpha = A(I - HB)\, y \end{cases}. \tag{13}$$

So, we can get the wavelet estimate function of the LS-SVM with Gaussian radial basis function (RBF) as

$$f(r, c) = \sum_{i=1}^{N} \alpha_i \exp\left\{-\frac{\left(|r - r_i|^2 + |c - c_i|^2\right)}{\sigma^2}\right\} + b, \tag{14}$$

where r_i, c_i are the image coordinates, and σ is the width of Gaussian kernel.

With Eqs. (12) and (13), Eq. (14) can be rewritten as

$$f(r, c) = \{FA(I - HB) + B\}\, y = Ty, \tag{15}$$

where $F = [f_1, f_2, \cdots, f_N]$, $T = FA(I - HB) + B$, T is called filter operator. As to Gaussian RBF, $f_i = \exp\left\{-\frac{\left(|r - r_i|^2 + |c - c_i|^2\right)}{\sigma^2}\right\}$. As to polynomial kernel, $f_i = (r \cdot r_i + c \cdot c_i + 1)^d$, and d is the order of polynomials.

Given input vectors, kernel type, kernel parameters, and regularization parameter, we will get A and B from Eq. (12), then pre-calculate the filter operator T, and it is a constant matrix.

We use the filter operator T to each high-frequency subband, and define the window center (0,0) where $r_i = 0$ and $c_i = 0$. Then the regression estimated value of window center $\bar{f}(0, 0) = TY$. In our algorithm, the shrinkage strategy is to compare the estimated value $\bar{f}(0, 0)$ with the original coefficient value $f(0, 0)$ of that pixel. And we can get the modified wavelet coefficient f_{m} from the following rule:

$$f_{\mathrm{m}} = \begin{cases} \bar{f}(0,0) & \left|f(0,0) - \bar{f}(0,0)\right| < p \\ \alpha \times f(0,0) + (1-\alpha) \times \bar{f}(0,0) & \\ & \left|f(0,0) - \bar{f}(0,0)\right| \geq p \end{cases}, \tag{16}$$

where $\alpha \in [0, 1]$ is a parameter and $p = 0.3 \times \max(f)$ is a threshold, $\max(f)$ is the maximal value of the wavelet coefficients in that subband. This equation means that if the margin between estimated value and original value is limited in a fixed interval, we will consider it as signal, otherwise, we will consider it as noise and reconstruct it. In Eq. (16), the value of α is generally chosen between 0.5 and 1, and we choose $\alpha = 0.6$ in this paper.

Therefore, in the three detailed subbands of wavelet domain, our crucial proposed algorithm is as follows.

Step 1: Proceed with 2D orthogonal wavelet transform to the image corrupted by Gaussian noise and get the wavelet coefficients $d_{j,k}$.

Step 2: Calculate the filter operator T according to Eq. (15).

Step 3: Estimate the center coefficient value of the

window with the equation $\bar{f}(0,0) = TY$.

Step 4: Manipulate wavelet coefficients according to the rule of Eq. (16).

Step 5: Repeat Steps 3 and 4 to each high subband at each level.

Step 6: Compute inverse wavelet transform to the manipulated image coefficients and obtain the denoised image.

To verify the proposed algorithm, sym4 wavelet is used for the wavelet decomposition throughout this work. We performed our experiments on the well-known 256×256 Lena and Girl gray images, which had been corrupted by the addition of simulated spatially Gaussian noise. As to the kernel, we chose the Gaussian RBF with $\sigma^2 = 0.2$, the polynomial kernel with the order $d = 1$, and the wavelet kernel described in this paper, respectively. The wavelet decomposition level was limited to two. To quantitatively evaluate the method, the peak signal-to-noise ratio (PSNR) was calculated and the results were compared with those obtained by using some traditional method, such as LS-SVM with Gaussian RBF, LS-SVM with polynomial kernel, average filtering, and median filtering.

Table 1 lists the PSNR values for the Lena image at five noise levels. Figure 2 shows the comparison of the denoised Lena image corrupted with Gaussian noise, and Fig. 3 shows the comparison of PSNR between the original image and other three crucial methods. From the experimental results, we can get our conclusions as follows. 1) When the clean image is corrupted by Gaussian noise, it has better performance to choose wavelet kernel than Gaussian RBF kernel and polynomial kernel. 2) Wavelet kernel preserves most of the detail information than any other methods discussed in this paper. 3) At a high level of SNR, our proposed method has better efficiency than any of the other methods, such as average filter and median filter, but when the SNR is at a low level, median filter has better efficiency. Meanwhile, we should emphasize that all the methods we discussed in this paper will ruin the quality of the image if the SNR is too high, in this situation, we should ignore the corruption caused by the noise.

In this paper, with the regression theory of LS-SVM, we get a new filter operator in wavelet domain to modify the neighborhood wavelet coefficients. Wavelet transform and denoising theory are discussed firstly, and then wavelet support vector machine (WSVM) is constructed based on the wavelet kernel. To evaluate the method, we compared our proposed theory with other image denoising methods through processing the image with Gaussian noise. Experimental results show that our proposed algorithm is better than the conventional denoising scheme in terms of PSNR.

Fig. 2. Comparison of different denoising methods for Lena image with PSNR = 24.04. (a) Original Lena image; (b) noisy Lena image; and denoised images by using (c) the proposed method, (d) Gaussian RBF, (e) polynomial RBF, (f) median filtering.

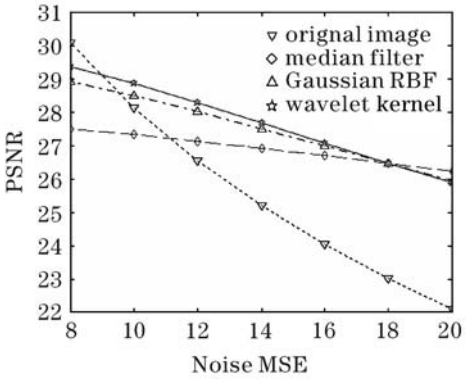

Fig. 3. Comparison of PSNR between the original image (Lena) and three methods. MSE: mean square error.

Table 1. PSNRs (dB) for Lena (mean = 99, std = 20.6458) and Girl (mean = 99.3750, std = 22.1549) Images, and the Denoised Images by Five Denoising Methods

Input	Lena 256×256				Girl 256×256			
	30.06	26.539	24.04	22.102	30.06	26.539	24.04	22.102
Gaussian RBF	28.921	28.005	26.974	25.93	28.664	27.562	26.495	25.21
Polynomial Kernel	28.514	27.608	26.526	25.429	28.195	27.226	26.147	24.962
Proposed	29.367	28.28	27.07	25.871	28.929	27.954	26.817	25.683
Average Filter	24.939	24.868	24.771	24.65	25.399	25.32	25.21	25.08
Median Filter	27.477	27.129	26.697	26.219	26.908	26.447	26.312	25.94

This work was supported by the Science Foundation of Beijing Jiaotong University under Grant No. 2005SM011. G. Zeng's e-mail address is cn_zeng222@126.com, and R. Zhao's e-mail address is rzhzhao@bjtu.edu.cn.

References

1. S. G. Mallat, IEEE Trans. Pattern Analysis and Machine Intelligence **11,** 674 (1989).
2. T. T.Cai and B. W. Silverman, Sankhya B **63,** 127 (2001).
3. Q. Gao, Z. Sun, Z. Cao, and P. Cheng, Chin. Opt. Lett. **2,** 113 (2004).
4. H. Fang and D. Huang, Chin. Opt. Lett. **2,** 1 (2004).
5. V. N. Vapnik, *The Nature of Statistical Learning Theory* (2nd edn.) (Springer-Verlag, New York, 2000) p.174.
6. J. A. K. Suykens and J. Vandewalle, Neural Processing Lett. **9,** 293 (1999).
7. L. Zhang, W. Zhou, and L. Jiao, IEEE Trans. System, Mar, Cybern. B **34,** 34 (2004).
8. T. T. Cai, Statist. Sin. **12,** 1241 (2002).
9. G. Y. Chen, T. D. Bui, and A. Krzyzak, Digital Object Identifier. **2,** 917 (2004).
10. H. Liu, Y. Guo, and G. Zheng, in *Proceedings of the* 6^{th} *World Congress on Intelligent Control and Automation* (in Chinese) 4180 (2006).