Kernel Resolution Synthesis for Superresolution

Karl Ni, *Student Member, IEEE,* and Truong Nguyen *Fellow, IEEE*

***Abstract*— This work considers a combination classification-regression based framework with the proposal of using learned kernels in modified support vector regression to provide superresolution. The usage of both generative and discriminative learning techniques is examined first by assuming a distribution for image content for classification and then providing regression via semi-definite programming (SDP) and quadratically constrained quadratic programming (QCQP) problems. The advantage of the proposed method over other learning-based superresolution algorithms include reduced problem complexity, specificity with regard to image content, added degrees of freedom from the nonlinear approach, and excellent generalization that a combined methodology has over its individual counterparts.**

***Index Terms*— superresolution, support vector regression, kernel matrix, kernel learning, interpolation, resolution, scaling**

I. Introduction

Single image superresolution is the ill-posed problem of determining high-resolution image content given low-resolution data. In determining the high-resolution image content, additional or new information must be introduced aside from the given low-resolution data. This information can come in many forms, including but not limited to a set of shifted low-resolution images of a single scene [1], assumed relationships between existing pixel values and edges [2], a training set [3], or any other data that would aid in enhancing visual acuity.

This work is concerned with maximizing the use of information inherent in a training set. The first step in utilizing the training set would be to describe the domain. This can be done through classification, in which content-based treatment offers good generalization. There are currently several classification based algorithms for superresolution, including [4], which due to its structural soundness, provides the classification framework for our classification-regression solution. [4] offers a stochastically modeled MMSE filtering technique by using localized choices for filters separated by Expectation Maximization (EM) under an assumption that a Gaussian mixture accurately describes localized image distributions.

While an excellent approximation that is robust to errors, linear interpolation in general has a tendency to average or smooth out image content, often requiring a presharpening step. In addition, with MMSE linear filtering, coefficients are chosen to represent an entire class. Should unsupervised clustering be insufficient, there are no accommodations for un-split classes and the errors carry over to observed results. Therefore, as a regression step, the proposed method substitutes a modified support vector regression (SVR) for its counterpart as a solution to these issues, thereby providing better function estimation and a good nonlinear approach. As SVR relies heavily on its kernel, its choice significantly affects the range and values of the regression and is optimized as well.

The remainder of this paper explores these issues. Support vector regression is reviewed and the kernel problem is formulated as a convex optimization problem in SDP and QCQP forms in Sec. II as review from previous works, [5] and [6]. With the derived learning technique, Sec. III describes the algorithm in its entirety including a brief overview of the original resolution synthesis framework.

The authors are with the Video Processing Laboratory at the University of California, San Diego *http://videoprocessing.ucsd.edu*. This work is supported by a grant from Qualcomm Inc. with matching funds from the U.C. Discovery Program.

II. Kernel Learning for SVR

The support vector machine (SVM), originally proposed in [7], is a supervised learning technique that determines a high-dimensional functional from a training set Ω,

$$\Omega = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\} . \quad (1)$$

The goal of SVR is to use relationships learned through Ω, and be able to generalize these relationships to unseen test points. In (1), $\mathbf{x}_i \in \Re^n$ and $y_i \in \Re, \forall i \in [1, N]$, and SVR estimates the function $f : \mathbf{x} \rightarrow y$ with the following optimization.

$$\min_{\mathbf{w},b,\xi} \left(\frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \left(\xi_i^+ + \xi_i^-\right) \right)$$

subject to

$$(\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) - y_i \leq \varepsilon + \xi_i^+$$
$$y_i - (\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) \leq \varepsilon + \xi_i^-$$

and

$$\xi_i^-, \xi_i^+ \geq 0, \ \forall \ i \in [1, N] \quad (2)$$

The high-dimensional mapping $\phi : \mathcal{X} \mapsto \mathcal{F}$ in (2) often better suits a representation of complicated relationships which could otherwise not be linearly realized. Within $\mathcal{F}$, a kernel function $K(\mathbf{s}, \mathbf{t})$ written as a kernel matrix $K(\cdot, \cdot)$ is defined to be a collection of dot products for an arbitrary ϕ that may or may not be known. With this in mind, the dual to (2) can be found and is written in (3).

$$\max_{\alpha^+, \alpha^-} -\frac{1}{2} \sum_{i,j} \left\{ (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) K(\mathbf{x}_i, \mathbf{x}_j) \right\} - \epsilon \sum_i (\alpha_i^+ + \alpha_i^-) + \sum_i y_i (\alpha_i^+ - \alpha_i^-)$$

subject to

$$\sum_i (\alpha_i^+ - \alpha_i^-) = 0 \text{ and } 0 \leq \alpha_i^{+/-} \leq C$$

with the solution hyperplane as

$$g(\mathbf{x}) = \sum_i (\alpha_i^+ - \alpha_i^-) K(\mathbf{x}, \mathbf{x}_i) + b \quad (3)$$

where a dot product in $\mathcal{F}$ is defined by $K(\mathbf{s},\mathbf{t}) = \phi(\mathbf{s}) \cdot \phi(\mathbf{t})$, the kernel function.

Using the kernel matrix K, computational complexity is reduced because determining $d = \phi(\mathbf{s}) \cdot \phi(\mathbf{t})$, which is quite often intractable, is unnecessary when solving (3). This definition also allows ϕ to be unknown, in which case, K can be conceptually chosen to be a desired similarity metric depicting the "nearness" of two vectors. Thus, the selection of the kernel matrix K becomes important and should be sensitive to the training data.

Several works have explored the prospect of learning the kernel matrix [8], [9], [6], [10],[11]. Of particular interest is [8] in which a linear combination of smaller, known kernels is optimized to produce a large kernel with good feature representation for the classification problem. The analogous optimization for regression has been explored in [9], although errors lead the derivation to an incorrect outcome. These errors have been addressed in a previous work [6], and the following is the reformulation of a semi-definite programming (SDP) and quadratically constrained quadratic programming (QCQP) problem to learn the kernel for regression in much the same way that it has been derived for classification.

1) The SDP Problem: For simplification, let e be a vector of all ones, and

$$\begin{aligned} \alpha^+ + \alpha^- &= \beta^+ \\ \alpha^+ - \alpha^- &= \beta^- \end{aligned} \tag{4}$$

Then, after writing the Lagrangian and observing Slater's conditions, we take the dual of (3) and use the Schur complement lemma as done in [6], and the optimal kernel can be found with the following optimization problem.

$$\begin{aligned} \min_{K,t,\lambda,\nu_u^+,\nu_l^-,\nu_u^-} \quad & t \\ \text{s.t.} \quad & \begin{pmatrix} 2K & \gamma \\ \gamma^T & t - 2Ce^T(\nu_u^+ + \nu_u^-) \end{pmatrix} \succeq 0 \\ & \nu_u^+, \nu_u^-, \nu_l^- \succeq 0 \\ & \epsilon e + \nu_u^+ + \nu_u^- - \nu_l^- \succeq 0 \\ & K \succeq 0 \\ & trace(K) = c \end{aligned} \tag{5}$$

If K is linear combination of fixed kernels $\{k_i\}$, then (5) is an SDP, and we optimize with respect to the coefficients of the linear combination. That is, when minimizing with respect to

$$K = \sum_i \mu_i k_i(\cdot,\cdot) \tag{6}$$

we are optimizing over possible values μ_i. This result is general and can be applied to any problem.

2) The QCQP Problem: The QCQP arises from an added constraint, $\mu_i \geq 0$, which loses some generality, though it does ensure positive definiteness when inductively applying the learned kernel. On the other hand, the complexity of the kernel is never simplified because the positive eigenvalues of each $(\mu_i k_i)$ will never reduce kernel rank.

The formulation we obtained is derived in the same manner as [8], and is given in (7).

$$\begin{aligned} \max_{\beta^+,\beta^-,p} \quad & 2y^T\beta^- - 2\epsilon e^T\beta^+ - cp \\ \text{s.t.} \quad & p \geq \beta^- k_i \beta^- \\ & e^T\beta^- = 0 \\ & 0 \preceq \beta^+ + \beta^- \preceq 2C \\ & 0 \preceq \beta^+ - \beta^- \preceq 2C \end{aligned} \tag{7}$$

Again, k_i are the smaller positive semi-definite kernels in (6) for kernel construction. The μ_i values come out of the dual Lagrangian variables.

III. Kernel Resolution Synthesis

One solution for a localized approach to superresolution uses the learned kernel in Sec. II for SVR to predict high-resolution pixel values from low-resolution patches. That is, if I_{LR} and I_{HR} are the low and high-resolution image patches with sizes $D \times D$ and $U \times U$ respectively, to superresolve the center pixel of I_{LR} by a factor of U, we define vectors

$$\begin{aligned} \mathbf{x} &= \text{vectorize}(I_{LR}) - \text{center pixel}(I_{LR}) \in \Re^{D^2 \times 1} \\ \mathbf{y} &= \text{vectorize}(I_{HR}) - \text{center pixel}(I_{LR}) \in \Re^{U^2 \times 1} \end{aligned} \tag{8}$$

in a given training set Ω of $\mathbf{x}_i$ feature and $\mathbf{y}_i$ label pairs.

Although SVR has the capability to provide a general regression of all possible image content with fairly clear results due to its scalability properties, a single regressor for a large training set introduces substantial computational complexity as well as problem complexity. Depending on the data set, the problem quickly becomes intractable in (5) and (7), when the kernel constraint for each $k_i(\cdot,\cdot)$ scales according to N^2 where N is the number of training points. If $K(\cdot,\cdot)$ is a sum of M small kernels, the order will exceed $M \cdot N^2$ without even considering other inequality constraints in (7). Also, without further enhancements, the idea relies on the heavy machinery of SVR to recognize all types of image content, which increases the problem complexity due to the large variety of $\mathbf{x}$ in $\mathcal{X}$.

Our approach is to partition the problem into smaller problems according to data similarity and hence more easily approximated ones. The remainder of this section is an extension to this concept by using the resolution synthesis framework of [4].

In [4], superresolution is approached stochastically by determining the conditional expectation of high-resolution pixels given low-resolution patches and training data. This is expressed in (9).

$$g(\mathbf{x}) = E[\mathbf{y}|\mathbf{x},\Omega] \tag{9}$$

where $g(\mathbf{x})$ uses the training set Ω to estimate $f : \mathbf{x} \mapsto \mathbf{y}$.

As previously discussed, an all-encompassing function $g(\mathbf{x})$ is substituted by several smaller functions $g_j(\mathbf{x})$ according to image content. By dividing the domain $\mathcal{X}$ into several classes (done by assuming a Gaussian mixture as in [4]), we decide which g_j is useful for a given $\mathbf{x}$. Thus, (9) is rewritten in (10) as a weighted average of possible reconstructed values based on various image content.

Denoting the random variable J as the class number of input $\mathbf{x}$, we write:

$$E[\mathbf{y}|\mathbf{x}] = \sum_j E[\mathbf{y}|\mathbf{x}, J=j]P(J=j|\mathbf{x}) \tag{10}$$

where we can determine

$$P(J=j|\mathbf{x}) = \frac{P(\mathbf{x}|J=j)P(J=j)}{\sum_j P(\mathbf{x}|J=j)P(j=j)} \tag{11}$$

when the likelihoods $P(\mathbf{x}|J=j)$ are determined by EM.

By approximating the class conditional expectations with a regression device, i.e. $E[\mathbf{y}|\mathbf{x}, J=j] = g_j(\mathbf{x})$, the solution becomes

$$E[\mathbf{y}|\mathbf{x}] = \sum_j g_j(\mathbf{x})P(J=j|\mathbf{x}), \tag{12}$$

and our approach is to apply SVR to estimate $g_j(\mathbf{x})$.

This is not a simple substitution in the implementation because it requires altering the optimization problem slightly. In [4], after forming the Gaussian PDF, each parameter for a given class is generated by contributions from each training point weighted by how far "in" the class the point is. For resolution synthesis, this is achieved by multiplying the posterior probability in an MMSE-type expression. For kernel resolution synthesis, this is not so easily done, as the parameters in kernel resolution synthesis arise from a nonlinear optimization.

From the dual problem in (3), the weighting of training points by their importance is analogous to the effect of C on the solution hyperplane. The C variable is actually a cost parameter whose value comes out of cross validation. In the dual problem, the larger the cost parameter, the more the $\alpha^{+/-}_{(i,j)}$ values can deviate for an exact regression, in effect granting freedom to closely fit the training data in exchange for flatness in the objective function. Therefore, for pair $(\mathbf{x}_i, y_i) \in \Omega$, limiting $\alpha^{+/-}_{(i,j)}$, also limits the effect of the i^{th} point on the solution hyperplane. In terms of the primal problem in (2), C scales the slack variables $\xi^{+}_{(i,j)}$ and $\xi^{-}_{(i,j)}$, restricting the quantity of points deviating from the solution hyperplane and by how much these points deviate.

Our answer is to scale each ξ_i for all points in Ω by how relevant the i^{th} point is to class j. This can be done with the product of all $\xi^{+/-}_{(i,j)}$ with their corresponding posterior probabilities P_{ij}. So, for the j^{th} regressor, the primal optimization problem is described by

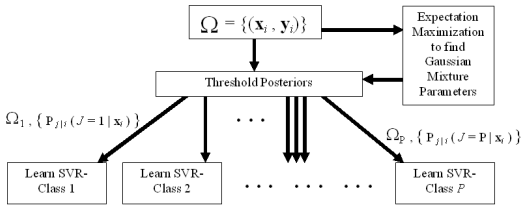
$$\min_{\mathbf{w}_j, b} \frac{1}{2}\|\mathbf{w}_j\|^2 + C \cdot \overrightarrow{P_{j|i}}(J=j|\mathbf{x}_i)^T (\overrightarrow{\xi^{+}_j} + \overrightarrow{\xi^{-}_j})$$

subject to

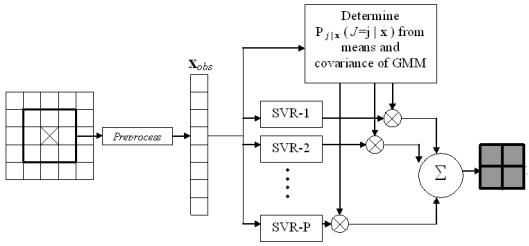
$$\begin{aligned} y_i - (\mathbf{w}_j \cdot \phi(\mathbf{x}_i) + b_j) - \epsilon &\leq \xi^{+}_{(i,j)} \\ (\mathbf{w}_j \cdot \phi(\mathbf{x}_i) + b_j) - y_i - \epsilon &\leq \xi^{-}_{(i,j)} \\ \xi^{-}_{(i,j)}, \xi^{+}_{(i,j)} &\geq 0 \end{aligned} \tag{13}$$

where $\overrightarrow{P_{j|\mathbf{x}_i}}(J=j|\mathbf{x}_i)$ denotes a vector of length N containing the posterior probabilities of classes. This way, we can allow more slack for the variables that are less important (i.e. have smaller posterior probabilities.) The probability vector in (13) can be equivalently placed throughout the rest of the derivations in Sec. III.

This solution has the potential to consume extensive computation both in CPU cycles and memory, and so a simplification of the problem would be to consider per class those points which meet a certain criterion with respect to their respective posterior probabilities. This is implemented by partitioning Ω into $\{\Omega_j\}$ and considering the i^{th} point for class j only if its posterior probability exceeds a certain threshold. The final algorithm is depicted in Fig. 1.



(a) Training Algorithm



(b) Testing Algorithm

Fig. 1. Kernel Resolution Synthesis Algorithm

Granted, this simplification rejects considerable amounts of data per class. Nevertheless, if SVR can indeed predict the relationship between low and high-resolution, then the regression may be sufficient for less relevant points in a given class. Furthermore, through experimentation, it turns out that only a few classes at any given test point are chosen and used for reconstruction the majority of the time. The implication from this is that for the test point $\mathbf{x}_{test}$, by multiplying $P_{j|\mathbf{x}}(j|\mathbf{x}_{test})$ with $g_j(\mathbf{x}_{test})$ in (10), we would maintain good accuracy by zeroing out test data that is irrelevant for a particular class anyway, leaving reconstruction for those classes which can accurately do so.

IV. Results and Analysis

The SDP ($\mu_i \geq 0$) and QCQP problems in Sec. II have been verified in a previous work [6] using the cvx [12] Matlab toolbox result. Further experiments on images and video frames for purposes of superresolution were carried out using MOSEK [13], which computes the QCQP problem in Sec. II-.2, alleviating the problem of the high complexity inherent in the SDP problem.

The algorithm was set up with $D = 5$ and $U = 2$, meaning that I_{LR} was 5×5 and I_{HR} was 2×2, with clustering features of size 3×3 for 4 training images in the CalPhotos image database from [14]. For fair comparison, the same training set is used for any relevant learning algorithms involving a training set to which we compare kernel resolution synthesis.

Comparisons to new edge-directed interpolation (NEDI) [2], subpixel edge localization (SEL) [15], resolution synthesis [4], and bicubic interpolation are shown quantitatively for frames in the bus sequence in Fig. 2 and qualitatively in Fig. 3. Kernel resolution synthesis not only achieves more accuracy in PSNR than its linear counterpart in [4] and other referenced methods, visual comparisons offer better clarity in Fig. 3 as well.

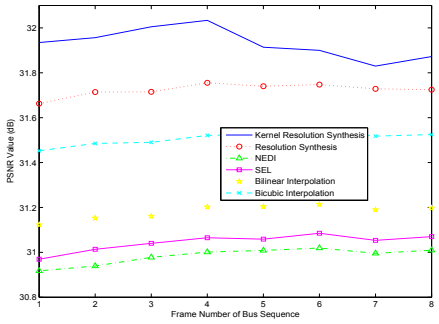
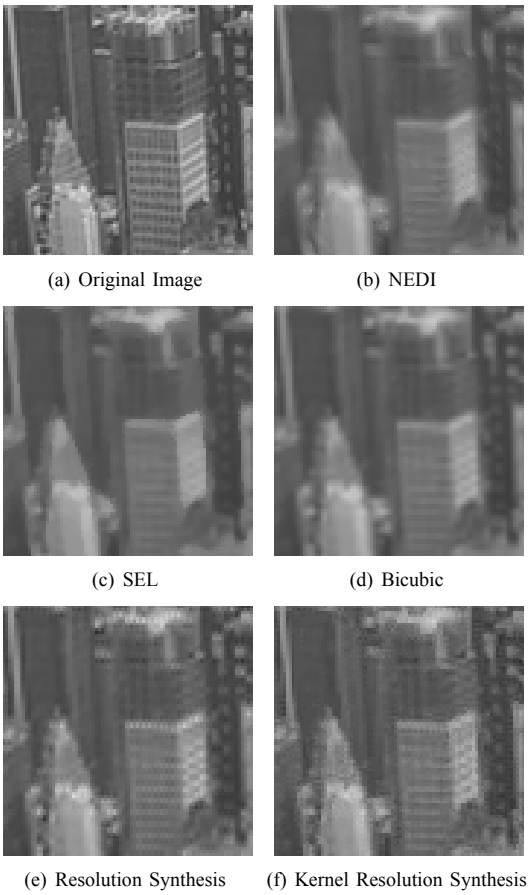


Fig. 2. PSNR Values in Video Frames: 8 Frames of the Bus Sequence.



(a) Original Image (b) NEDI

(c) SEL (d) Bicubic

(e) Resolution Synthesis (f) Kernel Resolution Synthesis

Fig. 3. Kernel Resolution Synthesis in comparison to various other techniques on a zoomed portion of the 6^{th} frame of the city video sequence.

Observing the visual results, resolution synthesis in Fig. 3(e) is closest in quality to the proposed method, but there are considerable errors near highly textured areas. Imperfections in Fig. 3(b) could be a by-product of a 2×2, two pass system in which [2] considers features independently. Experimental results assert that joint consideration is advantageous because optimization described by Sec. II results in a kernel with all 3×3 features for most classes. Comparisons to simpler interpolation techniques show that SEL enhances edges quite well, perhaps even better than NEDI, though the result looks slightly cartoonish. Additional images and comparisons can be found at research pages on UCSD's video processing website: `http://videoprocessing.ucsd.edu/~karl/krs_sr`

V. Conclusion

This work has proposed an approach to single image superresolution by offering a resolution synthesis framework for classification in conjunction with a nonlinear regression technique in the form of a modified SVR. Thus, both generative methods and discriminant learning methods are exploited to offer good numerical and visual results.

There are several issues open to future work. For example, one improvement becomes apparent when it is noted that for $U = 2$, current SVR predicts four outputs independently, when high-resolution inter-pixel relationships are highly correlated. Recent developments in the learning of vectored functions using operator-valued kernels [16] seem relevant and should greatly aid the solution.

References

[1] Patrick Vandewalle, Sabine Susstrunk, and Martin Vetter, "Superresolution images reconstructed from aliased images," in *Proceedings of Visual Communications and Image Processing Conference*, 2003, vol. 5150, pp. 1398–1405.

[2] X. Li and M. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, pp. 1521–1527, 2001.

[3] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based superresolution," *IEEE Computer Graphics Applications*, 2002.

[4] C. Brian Atkins and C. Bouman, *Classification based methods in optimal image interpolation*, Ph.D. thesis, Purdue University, 1998.

[5] Karl Ni, Sanjeev Kumar, T. Q. Nguyen, and N. Vasconcelos, "Single image superresolution based on support vector regression," *International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[6] Karl Ni, Sanjeev Kumar, and Truong Q. Nguyen, "Learning the kernel matrix for superresolution," *IEEE Conference on Multimedia Signal Processing*, to appear in October 2006.

[7] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[8] Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, 2004.

[9] S. Qiu and T. Lane, "Multiple kernel learning for support vector regression," Tech. Rep., University of New Mexico, 2005.

[10] Zhihua Zhang, Dit-Yan Yeung, and James T. Kwok, "Bayesian inference for transductive learning of kernel matrix using the tanner-wong data augmentation algorithm," *Proceedings of the* 21^{st} *International Conference on Machine Learning*, 2004.

[11] Brian Kullis, Matyas Sustik, and Inderjit Dhillon, "Learning low-rank kernel matrices," *International Conference on Machine Learning*, 2006.

[12] Michael Grant, Stephen Boyd, and Yinyu Ye, *CVX: Matlab Software for Disciplined Convex Programming*.

[13] *The MOSEK optimization toolbox for MATLAB manual. Version 4.0 (Revision 16)*, 2006.

[14] CalPhotos, *Cal Photos Image Collections*.

[15] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Transactions on Image Processing*, vol. 4, pp. 285–295, 1995.

[16] C. A. Micchelli and M. Pontil, "On learning vector-valued functions," *Neural Computation*, vol. 17, 2005.