Efficient Structured Support Vector Regression

Ke Jia[1,2], Lei Wang[1], and Nianjun Liu[1,2]

[1] College of Engineering & Computer Science, The Australian National University
[2] National ICT Australia (NICTA), Canberra, Australia
{ke.jia,lei.wang,nianjunl}@cecs.anu.edu.au

Abstract. Support Vector Regression (SVR) has been a long standing problem in machine learning, and gains its popularity on various computer vision tasks. In this paper, we propose a structured support vector regression framework by extending the max-margin principle to incorporate spatial correlations among neighboring pixels. The objective function in our framework considers both label information and pairwise features, helping to achieve better cross-smoothing over neighboring nodes. With the bundle method, we effectively reduce the number of constraints and alleviate the adverse effect of outliers, leading to an efficient and robust learning algorithm. Moreover, we conduct a thorough analysis for the loss function used in structured regression, and provide a principled approach for defining proper loss functions and deriving the corresponding solvers to find the most violated constraint. We demonstrate that our method outperforms the state-of-the-art regression approaches on various testbeds of synthetic images and real-world scenes.

1 Introduction

Structured prediction has recently attracted much attention and many approaches have been developed. Structured learning studies the problems in which both inputs and outputs are structured and exhibit strong internal correlations. It is formulated as the learning of complex functional dependencies between multivariate input and output representations. Structured learning has significant impact in addressing important computer vision tasks including image denoising [1], stereo [2], segmentation [3, 4, 5], object localization [6, 7], human pose estimation [8, 9], to name a few. A popular approach is to generalize from the max-margin binary/multiclass classification problems to incorporate structured information [10, 11, 5]. On the other hand, there still lacks a fundamental way to deal with structured prediction from regression viewpoint, which we believe is potentially a more general approach. The reason mainly comes from the continuous value range of regression outputs, where infinite possible labels exist. It is therefore difficult to be parameterize-modeled, efficiently trained and predicted. In the structured regression field, Weston *et al.* proposes a linear map model in [12] to unify support vector classification and regression, and develops a methodology to solve high dimension problems using joint kernel maps. The joint kernel based structured prediction method is also utilized in [7], which performs well to localize objects on real images. An alternative structured regression

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 586–598, 2011.

approach other than joint kernel is proposed in [9], in which output correlations are modeled in and the outputs act as auxiliary features.

However, although [12] tries to generalize support vector classification and regression and succeeds in independent training scenario, it fails in structured regression. The prediction on data in the form of Wx, which is formulated in [12], does not correctly count in label dependencies. Because feature map x does not depend on the label outputs in regression, no label correlation is modeled in the prediction. In fact, the "structured" term in [12] for regression can be regarded as a simple combination of node and neighboring local features. In addition, a fixed label loss function is given in [12], which makes the penalty of label discrepancy unchangeable for its specializations, both classification and regression. The dependency among label outputs is properly formulated in the projection function of [9], but it only considers the internal correlations among outputs. In this case, the label smoothness is applied equally to all nodes belonging to the same output and those having significant outputs. Obviously, valuable information in context is not fully utilized here to assist the smoothing. The work in [9] uses the original SVR constraints in their structured regression framework. The structured learning usually deals with data in very large scale. In this case, the SVR settings in [9] will accumulate quite a number of constraints. Consequently, it significantly increases the computational cost.

In this paper, we propose to address the problem of structured prediction from regression viewpoint. In particular, we have following three contributions. First, we devise a projection function that takes the label output as an additional weight for the pairwise feature. By doing this, our projection contains a full set of dependencies including three kinds of correlation between output variables, input variables and input-output interrelated variables. Second, by utilizing bundle method learning process, our algorithm efficiently solves the complex objective function in a small number of iterations. By further adopting the 1-slack trick, the number of constraints increases by only 1 in each iteration. A smaller sized constraint set significantly speeds up the training process on large scale data. This setting of constraints also improves the robustness of our algorithm, because it alleviates the impact of outliers. Third, we design a principled approach to define proper loss functions for tasks with various regression targets, and also to derive corresponding solvers to find most violated constraint with respect to the defined loss function. This provides an effective way for practitioners to design suitable loss functions for a given task. Focusing on our study case, we attain an appropriate loss function and derive an efficient solver following our principled approach. We empirically evaluate our approach on both synthetic and real-world data sets, and it outperforms the state-of-the-art regression methods.

The outline of the paper is as follows. The proposed approach is described in Section 2, in which the detail of discussion about loss function and most violated constraint is also included. Comparison of our approach to related methods including M^{3}Ns, Joint Kernel Maps and SOAR$_{svr}$ is also presented in Section 2. Section 3 reports the experimental results on synthetic and real data sets, together with the empirical analysis. A conclusion is drawn in Section 4.

2 Our Approach

2.1 Problem Description

Let us denote an image instance as $X \in \mathcal{X}$ and its observation as $Y \in \mathcal{Y}$. They are defined over a graph $G = (V, E)$ of size $|V| = d$, respectively. Y is a continuous space defined over its pixels with each pixel assigned a real value $y_i \in \mathcal{R}$, and $Y = (y_1, y_2, \cdots, y_d)^T$. More specifically, let $i \in V$ index a node i and $ij \in E$ index an edge between vertexes i and j of the graph G.

In the training phase we have access to a set of T ground-truth images as $\{(X_t, Y_t)_{t=1}^T\}$. Our aim is to learn a W-parameterized projection function $F : \{X, Y, W\} \to Y'$. Here we need the model to take the correlations between output variables into account, *i.e.,* in the projection function, observation Y plays a role of an auxiliary for visual features to generate the global label outputs. The model parameter W is learned over the training set of T images. We require the optimal output of F for X_t to be as close as possible to its ground-truth value Y_t. Meanwhile, we need the model to generalize well on unseen images. In our work, for each node i, we assume that the projection F can be locally modeled by a linear function

$$f_i : \{X, Y, W\} \to y_i' \equiv f_i(x_i, \mathbf{y}^{-i}, W) = \langle w_v, x_i \rangle + \frac{1}{N} \sum_{j \in \mathcal{N}_i} \langle w_e, x_{ij} \rangle \, y_j, \tag{1}$$

where $\mathbf{y}^{-i}$ is the $(d-1)$-dimensional output vector without the i-th entry, x_i and x_{ij} (or w_v and w_e) are local node and edge components of X (or W), and $\mathcal{N}_i$ denotes the set of the N neighboring nodes of the i-th node. Denoting $\phi(x_i, \mathbf{y}^{-i}) = (x_i, \frac{1}{N} \sum_{j \in \mathcal{N}_i} x_{ij} y_j)$, we have $f_i(x_i, \mathbf{y}^{-i}, W) = \phi(x_i, \mathbf{y}^{-i})W$, where $W = (w_v^T, w_e^T)^T$. At the image level, we write the features in a matrix form as $\Phi(X, Y) = \{\phi(x_i, \mathbf{y}^{-i})|_{i=1}^d\}$. Thus, the projection function F can be expressed as

$$F(X, Y, W) = \{f_i(x_i, \mathbf{y}^{-i}, W)|_{i=1}^d\} = \Phi(X, Y)W. \tag{2}$$

Now, given an unseen image X, this regression problem can be formally described as predicting the graph output Y^* by minimizing a loss function defined between an output Y and the projection upon it,

$$Y^* = \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \ \mathcal{L}_I(Y, F(X, Y, W)) = \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \sum_{i=1}^{d} L(y_i, f_i(x_i, \mathbf{y}^{-i}, W)).$$

The loss function $L(a, b)$ evaluates the difference between two labels a and b. It returns a positive value when $a \neq b$, and zero otherwise. Evidently, the minimum value of $\mathcal{L}_I$ should be zero, and this implies that the optimal output Y^* needs to satisfy $F(X, Y^*, W) = Y^*$. In terms of L_2-norm, it gives

$$\begin{aligned} Y^* &= \arg_{Y \in \mathcal{Y}} (F(X, Y, W) = Y) \\ &= \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \ \|F(X, Y, W) - Y\|^2. \end{aligned} \tag{3}$$

Notice that the minimum value of zero can always be achieved when $F(X, Y, W)$ is a linear function of Y. We will show this later.

2.2 Our Max-Margin Formulation

Given an image-label pair (X_t, Y_t), we would like the Euclidean distance between the predicted label, $F(X_t, Y_t, W)$, and the ground-truth Y_t to be minimum for any candidate output Y. This yields, for the set of T training images,

$$\|F(X_t, Y_t, W) - Y\|^2 - \|F(X_t, Y_t, W) - Y_t\|^2 \geq \Delta(Y_t, Y) \;\; \forall t, Y. \tag{4}$$

where $t = 1, 2 \cdots, T$. Here $\Delta(Y_t, Y)$ denotes another loss function, which plays the role of *margin* between the true output Y_t and any other candidate output Y. This loss function could in general be an arbitrary function defined over the graph that measures the discrepancy between two label assignments. In other words, it is non-negative, symmetric, and attains zero if $Y_t = Y$. For the structured support vector regression, we need to avoid using the loss function like $\Delta(Y_t, Y) = \|Y - Y_t\|^2$. A comprehensive discussion about the $\Delta(Y_t, Y)$ will be given in Section 2.3.

By invoking (2), we expand left-hand-side (LHS) of (4) and after some algebra, it gives

$$2(Y_t^T - Y^T)\Phi(X_t, Y_t)W + Y^T Y - Y_t^T Y_t \geq \Delta(Y_t, Y) \;\; \forall t, Y. \tag{5}$$

Notice that the quadratic terms of W in the LHS of (4) have been canceled out, and this gives a linear function of W.

Now, we optimize (5) for all possible labels Y, and at the same time minimize the norm of W to avoid trivial solutions. Adding the slack variables $\xi_t \geq 0$ to account for violations, the optimization problem reads, for $\eta > 0$,

$$\begin{aligned} &\min_{W, \xi_t} \quad \frac{\|W\|^2}{2} + \frac{\eta}{T} \sum_{t=1}^{T} \xi_t \qquad\qquad (6) \\ &s.t. \;\; 2(Y^T - Y_t^T)\Phi(X_t, Y_t)W + Y_t^T Y_t - Y^T Y + \Delta(Y_t, Y) \leq \xi_t \;\; \forall t, Y. \end{aligned}$$

2.3 Parameter Learning and Inference Details

Bundle Method. For the optimization problem in (6), it has been shown in [13] that one may use a bundle method to find an approximate solution in polynomial time. For the convenience of analysis, we rewrite the constrains of (6) into the form of (4). The constraint related to the t-th image can thus be equivalently expressed as

$$\xi_t \geq \max_Y [\|F(X_t, Y_t, W) - Y_t\|^2 - \|F(X_t, Y_t, W) - Y\|^2 + \Delta(Y_t, Y)].$$

That is, the violations of constraint (RHS) is upper bounded by ξ_t (LHS). Given the current W, the bundle method can be used to optimize the objective function, which needs to identify the most violated constraint. Noticing that only the last two terms depend on Y, this leads to efficiently solve the following column generation problem

$$\begin{aligned} Y_m &= \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \, [\|F(X_t, Y_t, W) - Y\|^2 - \Delta(Y_t, Y)] \\ &= \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \, [\|Y_t' - Y\|^2 - \Delta(Y_t, Y)], \end{aligned} \tag{7}$$

where $Y_t' = F(X_t, Y_t, W)$ given current approximated W.

The bundle method used in our training procedure (Algorithm 1) is guaranteed to approach the optimal solution to arbitrary precision in less than $O(\frac{1}{\zeta})$ iterations where ζ is the small tolerance in stopping criterion. By further adopting the one-slack trick of [13], empirically it always converges in a small number of iterations, which promises a tightly-bounded size for the constraint set $\mathcal{S}$ in our algorithm.

Loss Function and Most Violated Constraint. Now, we solve the column generation problem in (7) for the most violated constraint as follows. This part has the following contributions. Firstly, we develop a general approach to define proper loss functions and derive the corresponding solver to find the most violated constraint. Secondly, focusing on the special case of ε-insensitive loss, we derive a serial of possible solvers to efficiently identify the most violated constraint, the formulation in [12] is shown to be a special example in the serial.

In the space of $\mathcal{R}^d$, there exists at least one 2D affine plane on which the labels Y_t', Y_t and an arbitrary label assignment Y forms a triangle as shown in Fig. 1. For ease of exploration, denote $\|Y_t' - Y_t\| = a$, $\|Y - Y_t\| = b$ and $\|Y - Y_t'\| = c$, the angle $\angle Y Y_t Y_t' = \theta$. Thus we have $c^2 = a^2 + b^2 - 2ab\cos\theta$.

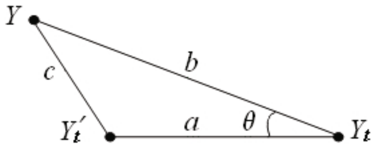

Fig. 1. The labels Y, Y_t' and Y_t in the triangular locations on a Euclidean plane

Assume that the label loss is a function of b, denoted as $\Delta(b)$. We can rewrite the most violated constraint (7) as:

$$\begin{aligned} Y_m &= \underset{b,\theta}{\operatorname{argmin}} \, [c^2 - \Delta(b)] \\ &= \underset{b,\theta}{\operatorname{argmin}} \, [a^2 + b^2 - 2ab\cos\theta - \Delta(b)]. \end{aligned}$$

Denoting $g(b, \theta) = a^2 + b^2 - 2ab\cos\theta - \Delta(b)$, we can get $\frac{\partial g}{\partial \cos\theta} = -2ab < 0$ because of $a > 0$ and $b > 0$, which means smaller g will always be obtained from larger $\cos\theta$. Thus, the minimum g should come from $\cos\theta = 1$, that is $\theta = 0$. Therefore $c^2 = a^2 + b^2 - 2ab = (b-a)^2$ and also $g(b) = (b-a)^2 - \Delta(b)$.

Let's consider the partial derivation of g with respect to b:

$$\frac{\partial g}{\partial b} = 2b - 2a - \frac{\partial \Delta(b)}{\partial b}. \tag{8}$$

If the loss function is defined as those like $\Delta(Y_t, Y) = \|Y - Y_t\|^2 = b^2$, it will lead to $\frac{\partial g}{\partial b} = -2a < 0$. This makes $g(b)$ monotonically decrease with the increasing value of b. As a result, the minimum of $g(b)$ cannot be effectively achieved and this prevents the most violated constraint Y_m from being identified. Such a case should be avoided.

We can also rewrite (4) into the triangular form as

$$\begin{aligned} c^2 - a^2 \geq \Delta(b) \quad &\Rightarrow \quad (b-a)^2 - a^2 \geq \Delta(b) \\ &\Rightarrow \quad a \leq \frac{b^2 - \Delta(b)}{2b}. \end{aligned} \tag{9}$$

In regression, a number of different loss functions could be utilized. Here we focus on the commonly used ε-insensitive squared loss, and the other loss functions can be analyzed in a similar way. By invoking the ε-insensitive loss defined as $a \leq \varepsilon$, we would like to find out b and $\Delta(b)$ satisfying

$$\frac{b^2 - \Delta(b)}{2b} = \varepsilon \quad \Rightarrow \quad \Delta(b) = b^2 - 2b\varepsilon. \tag{10}$$

By combining (10) and (8), we obtain the partial derivation $\frac{\partial g}{\partial b} = 2\varepsilon - 2a > 0$. Therefore, when $b \in [\lambda, \infty)$, where λ is a positive constant, the optimal solution of the most violated constraint would be obtained at the point $b = \lambda$.

Because $\Delta(b) \geq 0$, b should be bounded at least $b \geq 2\varepsilon$ according to (10). For simplification, we choose $b \in [3\varepsilon, \infty)$ and work out the smallest $g(b)$ at $b = 3\varepsilon$. Thus, we have $\Delta(Y_t, Y_m) = \Delta(b) = 3\varepsilon^2$. Recall that $\theta = 0$, the most violated constraint Y_m can be calculated given Y_t' and Y_t,

$$Y_m = Y_t + 3\varepsilon \frac{Y_t' - Y_t}{\|Y_t' - Y_t\|}. \tag{11}$$

If choosing $b = 4\varepsilon$, we will come to $\Delta(b) = 8\varepsilon^2$. This is just the loss function that is directly defined and used in [12]. As shown, it is a special case of our definition. More importantly, our analysis explains when and why such a loss function and the alike would work.

Finally, the primal quadratic program (6) yields,

$$\begin{aligned} &\min_{W,\xi} \quad \frac{\|W\|^2}{2} + \eta\xi \qquad (12) \\ &s.t. \quad \frac{1}{T}\sum_{t=1}^{T}[2(Y_m^T - Y_t^T)\Phi(X_t, Y_t)W + {Y_t}^T Y_t - Y_m^T Y_m + 3\varepsilon^2] \leq \xi, \\ &\qquad \xi \geq 0. \end{aligned}$$

Algorithm 1. Bundle Method Parameter Learning

```
Input: data $X_t$, labels $Y_t$, sample size $T$, insensitive radius $\varepsilon$, tolerance $\zeta > 0$
Initialize constraint set $\mathcal{S} \leftarrow \varnothing$, parameter $W \leftarrow \mathbf{0}$
repeat
   for $t = 1$ to $T$ do
      $Y'_t \leftarrow F(X_t, Y_t, W)$
      $Y_m \leftarrow Y_t + 3\varepsilon \frac{Y'_t - Y_t}{\|Y'_t - Y_t\|}$
   end for
   Increase constraint set $\mathcal{S} \leftarrow \mathcal{S} \cup \{Y_m\}$
   $(W, \xi) \leftarrow$ Optimize (12) using all existing $Y_m \in \mathcal{S}$
until $\frac{1}{T}\sum_{t=1}^{T}[2(Y_m^T - Y_t^T)\Phi(X_t, Y_t)W + {Y_t}^T Y_t - Y_m^T Y_m + 3\varepsilon^2] \leq \xi + \zeta$
```

Inference of Y^*. According to (3), the optimal prediction can be calculated by solving the following matrix algebra:

$$\begin{aligned} Y^* &= F(X, Y^*, W) = X_v w_v + (\frac{1}{N}\sum_j X_e w_e)Y^* \\ \Rightarrow \quad Y^* &= (I - \frac{1}{N}\sum_j X_e w_e)^{-1} X_v w_v, \end{aligned} \tag{13}$$

where X_v and X_e are node and edge parts of standard feature map X.

Since the dimension of Y is generally high, which is the number of pixels in image processing. As a result, the inverse operation in (13) cannot be efficiently solved for a large-sized image. In this case, gradient-based optimizers can be used to efficiently attain an approximate solution, and we recommend to use unary outputs $X_v w_v$ as initialization.

2.4 Relationship to Existing Work

M^{3}Ns. The Max-Margin Markov Networks (M^{3}Ns) [11] is a structured version for SVM classification. We will show below that M^{3}Ns can be viewed as a special case of our SSVR framework.

For a structured classification, Y is a binary ("0" or "1") vector with the dimension of q^d, where q is number of categories for every node. There is one and only one "1" in Y, indicating one of the q^d possible label assignments for an image. The space of this Y is a special case of the continuous space $\mathcal{R}^d$ used in our SSVR, because $\mathcal{R}^d$ can be regarded as the same configuration vector with ∞^d dimension. As $Y^T Y = 1$ in the classification case, the constraint (5) is therefore

$$Y_t^T \Phi(X_t, Y_t)W - Y^T \Phi(X_t, Y_t)W \geq \Delta(Y_t, Y) \ \ \forall t, Y,$$

which is the same as that in M^{3}Ns. Here Y_t^T and Y^T can be regarded as a tensor to switch on only one column of $\Phi(X_t, Y_T)$ at one time, due to that there is only one "1" in Y and the others are all "0". Thus, M^{3}Ns is a special case of SSVR under conditions of discrete label space.

Joint Kernel Maps. As we described in Section 1, the objective function in [12], which takes the form of Wx, is commonly used in unary regression and structured classification algorithms, but it will fail and lost its "structured" sense in structured regression because it is just a simple combination of local node and edge features in this case. Our objective function in (1) correctly integrated the structured information, where the neighboring outputs will affect the result of central point as a smooth term.

Due to the modification of the objective function, the inference algorithm has to be changed in our framework to generate a global optimization of output assignments, while that in [12] just needs some simple linear algebra to get pointwise result. We adopted a gradient-based algorithm as the inference engine to calculate the global smoothed output.

$\mathbf{SOAR}_{svr}$. Bo *et al.* proposed a closely related framework in [9], named SOAR_{svr}. SOAR is a rather general framework for structured regression. Our projection function in (1) can be regarded as a special case of that in SOAR. However, such specialization is necessary for a general framework like SOAR to be able to work for the image-based regression, for example, estimating the disparity value for each pixel in our work. The projection function in SOAR defines a parameterized weight for each component of a sample. When straightforwardly applied to a pixelwise image regression problem, SOAR will associate different weights to the locations of different pixels in an image. This will cause problems because the location of a given pixel changes with image translation, scaling, and rotation. To address this issue, the weights in our SSVR framework are designed to be independent of the pixel locations, which makes it able to handle the above image transformations and thus work for pixelwise image regression.

We also developed the projection function of SOAR into (1) by incorporating the pairwise features into the smooth term, as shown in the second term in (1). In contrast, only label outputs are taken into account in SOAR. This modification is important because the pairwise features, which measure the discrepancy between the features of neighboring pixels, will certainly provide more information about their similarity and thus help to achieve better cross-smoothing over pairwise pixels.

The learning algorithm in our approach is also different from that in SOAR. An original ε-insensitive SVR learning procedure is adopted in SOAR, while we utilize the bundle method in our algorithm, motivated by three main advantages. First, our max-margin formulation bounds the prediction Y in a tighter insensitive zone compared to the SVR formulation used in both unary SVR and SOAR. Recall that the dimension of Y is d. The original SVR applies 1D regression over each dimension of Y, and its insensitive zone is a hyper-cube with edge length of ε. A sample may not be penalized even if its distance from the origin is as far as $\sqrt{d}\varepsilon$. Differently, we bound the insensitive zone with a hyper-sphere of radius ε, which is a tighter zone inscribing into the hyper-cube. A sample will surely be penalized once its distance from the origin is larger than ε. This insensitive zone can also be found in [14]. However, that work formulates the structured regression as a quadratic-constrained quadratic program, while our formulation solves the same problem in a quadratic program with linear constraints, which makes the learning process faster and lower the computational cost. Second, by utilizing the bundle method learning approach and adopting the one-slack trick, our algorithm finds the optimal solution in limited iterations with significantly less number of constraints than SOAR. In practice, our algorithm costs much less memory and is faster than SOAR for large-sized data sets. And the third, outliers can significantly affect the regression performance [15]. The existence of a small percentage of outliers is sometimes sufficient to make regression solvers produce very poor solutions. Real-world images have complex scene and always contain outliers that cannot be effectively explained by a regression model. Through identifying the most violated constraints, our approach uses a set of hyperplanes to approximate the objective function, forming a piecewise linear function. This avoids approximating the noise component in the objective function, and improves the robustness of the regression. This will be demonstrated by the experimental study.

To achieve a fair comparison between SOAR and our approach, we implement a SOAR-like algorithm (termed SSVR^{cube}), which utilizes the original SVR learning algorithm of SOAR, with the projection function modified by our formulation in (1) to deal with the pixelwise image regression tasks. We compare the results of both approaches in next section.

3 Experiments

Our approach has been evaluated on a variety of image testbeds. First we test on a synthetic binary images denoising data set, where the goal is to verify that the proposed approach indeed outperforms the state-of-the-art regression approaches. For data sets involving more complicated scenes, we show that our algorithm still outperforms them. During the experiments, we use the LibSVM package[1] for unary ε-SVR, and our approach is implemented in MATLAB 2006a.

3.1 Binary Synthetic Images

Our first goal in this section is to show the advantage of our approach over SVR by exploiting local pixel interactions, and also the improvement by utilizing bundle method learning procedure. We experiment with a binary denoising data set from [16]. The set contains 50 synthetic images corrupted with bimodel noises. Here all images are in the size of 64 × 64 pixels. 40 images are used in training, and they are divided into 4 equal groups (10 images in each group) to do the 4-folder tuning, where $\eta = 50$ is selected based on the tuning results. The rest 10 images are left aside for test.

[1] Available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm/

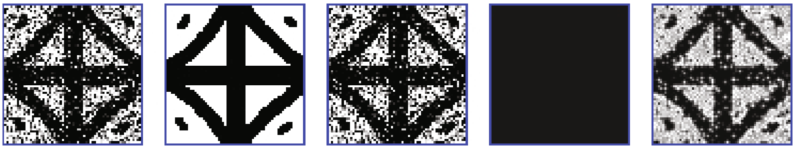

Fig. 2. Exemplar result of binary synthetic image denoising. From left to right: disturbed image, ground-truth and columns 3-5 are regression results of ε-SVR, SSVR^{cube} and our SSVR. Values lower than 0 or higher than 1 are adjusted to 0 or 1 respectively.

For a fair comparison, we use the same node features x_i for SVR, SSVR^{cube} and the proposed SSVR. The node features involving 2 dimensions which is the pixel grayscale value and 1-dim bias. We use the absolute difference of grayscale between the two neighboring pixels as the edge features x_{ij}, thus no additional information is provided to the structured learning approaches other than the unary one. The ground-truth label of pixels is either 0 or 1.

Table 1. Mean squared loss comparison of methods on the synthetic binary denoising dataset of [16]. The ε-SVR training time is roughly measured on LibSVM because the package does not record running time.

Methods	ε-SVR	SSVR^{cube}	SSVR
Training time (seconds)	7200	7.73	1.55
Mean squared loss	0.152	0.483	0.086

The experimental results in Fig. 2 and Table 1 validate that empirically our approach outperforms competition methods with a minimum mean squared loss of 0.086 and shortest training time. It is clear that our SSVR is more robust than other two algorithms. It can be observed that ε-SVR gives out a result almost the same as original disturbed image, and SSVR^{cube} failed completely on the task, due to the heavy outliers condition. Our approach successfully attains a proper model, which obviously adjustes the corrupted pixels by lifting noisy dark pixels (low values). Some bright pixels are inversely influenced for value reducing, because the learned model acts in a smoothness behavior. The training time of our algorithm is 1.55 seconds, and it labels an image in 0.54 seconds on average, measured by running on a desktop with 2.4GHz intel CPU and 2Gb memory.

3.2 Middlebury Stereo Datasets

To evaluate our approach on more complicated data sets, we test it on Middlebury Stereo Datasets from [17]. This data set consists of 6 scenes including *Art*, *Books*, *Dolls*, *Laundry*, *Moebius* and *Reindeer*. For each scene, 7 images are captured from different views (0 to 6), and 2 disparity maps related to view 1 and 5 are given as ground-truth. Images of scene *Laundry* and *Reindeer* are sized

Table 2. Comparison of methods on the disparity estimation. Fig. 3 displays some representative results on this dataset.

Methods	ε-SVR	SSVR^{cube}	SSVR
Training time (hours)	>200	1.65	1.29
Mean squared loss	n/a	5802.9	478.7

447 × 370 pixels, while others sized 463 × 370. We used the 12 images with ground-truth information available as our testbed, and two neighboring viewed images of each example are utilized for feature extraction. We left 3 images aside for test, and tuned the parameters using 3-folder cross-validation on 9 training images, the results suggest the parameter $\eta = 450$.

The ground-truth disparities are used as pixelwise labels, which value between 0 and 255. Two groups of features are adopted in our experiment, plus a 1-dim bias. First group is local visual features representing colors and textures, including 3-dim RGB color channels, 3-dim YCbCr color channels, and 11-dim texture features. For YCbCr color space, channel Y is image intensity, Cb and Cr are two color channels. Texture information is mostly contained in image intensity, so we applied the 9 Laws' masks [18] scaled 3 × 3 to channel Y. And the first Laws' mask is also applied to both color channels to extract haze. The local features include 17 dimensions in all. And we use 5-dim rough disparity estimations as features in the second group. For 5 different kinds of features (RGB color, YCbCr color, 4 Prewitt edge detectors oriented at 45° intervals filtered outputs, 3 × 3 Laws' mask response, and 5 × 5 Laws' mask response), we obtained the roughly approximated disparity maps respectively, using feature similarity. The disparity estimation of each pixel is attained by finding the most similar points along the same horizontal line in two neighboring viewed images. The top 2 similar points are chosen and average disparity of them is used as the estimation for current pixel. The 23-dimension node features plus 22-dimension edge features, which are absolute differences between neighbors, are input into different algorithms to regress the final disparities.

We compare our approach with other two methods, and results are shown in Fig. 3 and Table 2. The SSVR^{cube} result is scaled for better observation. LibSVM needs too much time to finish running, therefore its results is not available. All running times are measures on the same server with 2.8GHz CPU and 4Gb memory.

Our approach consistently obtains the lowest mean squared loss. The SSVR^{cube} algorithm, which uses the original SVR constraint set, is significantly influenced by outliers. The linear model it learned cannot properly represent the data distribution and highlight the outstanding features. Therefore, it over balances all labels to almost a constant, and the result keeps many intensity details, which plays an important role in features. Meanwhile, our approach demonstrates its robustness, and produces good result. On the large scale data set, the original SVR constraint set is extremely large, and thus it exponentially increases the complexity of quadratic program, running time for both ε-SVR and SSVR^{cube}

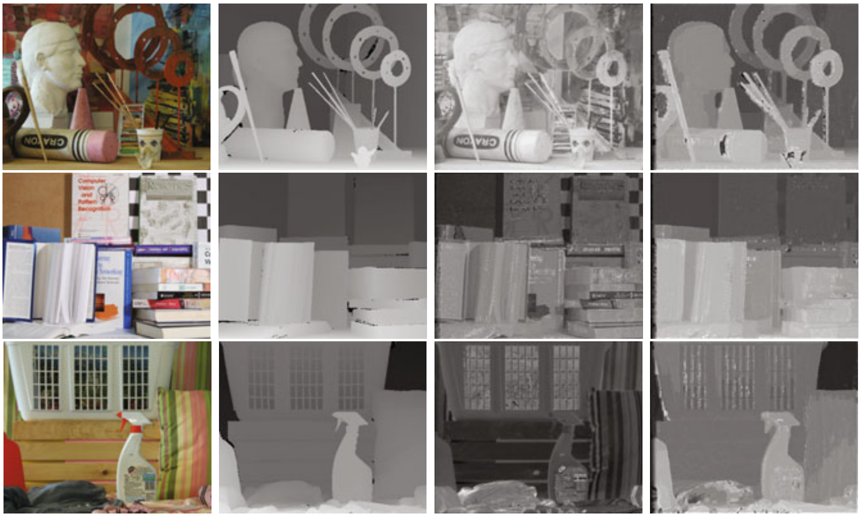

Fig. 3. Examples of disparity estimation on Middlebury stereo data sets. Columns from left to right: image, ground-truth, SSVR^{cube} result scaled for ease of view (original result is pretty dark), and prediction of our approach.

grow rapidly. Since our approach solves the problem iteratively with small constraint set, it takes the shortest time to learn the model. The average test time of one image is 13.94 seconds using a CSD gradient optimizer.

A more complicated feature space including 54 node features and 53 edge features was also experimented, the result of SSVR is similar to above one. But quadratic program in SSVR^{cube} cannot be solved because its redundant constraint set used up the memory, and LibSVM takes too long on calculation.

4 Conclusion

An efficient and robust structured support vector regression framework is proposed, and its performance is demonstrated through the problems of image denoising and disparity estimation in this paper. By incorporating the pairwise features into the projection function, our approach adaptively adjusts the impact of the neighboring pixels to the label of a given pixel according to their visual similarity. This advantage has been well demonstrated by the experiment on the binary synthetic data set. With the bundle method, our approach has significantly reduced the number of constraints by several orders since only the most violated ones are identified and used. As demonstrated by the experiment on the Middlebury stereo data set, our approach is superior to the existing methods on large-sized data sets in terms of both memory usage and running speed. Our analysis on the label loss function provides a principled way for practitioners to design suitable loss functions for a given task, which ensures the proper convergency of the bundle method learning process.

References

1. McAuley, J.J., Caetano, T.S., Smola, A.J., Franz, M.O.: Learning high-order mrf priors of color images. In: International Conference on Machine Learning (2006)
2. Carr, P., Hartley, R.: Minimizing energy functions on 4-connected lattices using elimination. In: International Conference on Computer Vision (2009)
3. Szummer, M., Kohli, P., Hoiem, D.: Learning cRFs using graph cuts. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 582–595. Springer, Heidelberg (2008)
4. Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A.: Discriminative learning of markov random fields for segmentation of 3d scan data. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2005)
5. Taskar, B., Chatalbashev, V., Koller, D.: Learning associative markov networks. In: International Conference on Machine Learning (2004)
6. Ionescu, C., Bo, L., Sminchisescu, C.: Structural svm for visual localization and continuous state estimation. In: International Conference on Computer Vision (2009)
7. Blaschko, M., Lampert, C.: Learning to localize objects with structured output regression. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 2–15. Springer, Heidelberg (2008)
8. Kim, M., Pavlovic, V.: Dimensionality reduction using covariance operator inverse regression. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)
9. Bo, L., Sminchisescu, C.: Structured output-associative regression. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009)
10. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research 6, 1453–1484 (2005)
11. Taskar, B., Guestrin, C., Koller, D.: Max-margin markov networks. In: Advances in Neural Information Processing Systems, vol. 16. MIT Press, Cambridge (2004)
12. Weston, J., Schölkopf, B., Bousquet, O.: Joint kernel maps. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 176–191. Springer, Heidelberg (2005)
13. Teo, C., Smola, A., Vishwanathan, S., Le, Q.: A scalable modular convex solver for regularized risk minimization. In: International Conference on Knowledge Discovery and Data Mining (2007)
14. Pérez-Cruz, F., Camps-Valls, G., Soria-Olivas, E., Pérez-Ruixo, J.J., Figueiras-Vidal, A.R., Artés-Rodríguez, A.: Multi-dimensional function approximation and regression estimation. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, p. 757. Springer, Heidelberg (2002)
15. Colliez, J., Dufrenois, F., Hamad, D.: Robust regression and outlier detection with svr: Application to optic flow estimation. In: British Machine Vision Conference (2006)
16. Vishwanathan, S.V.N., Schraudolph, N.N., Schmidt, M.W., Murphy, K.P.: Accelerated training of conditional random fields with stochastic gradient methods. In: International Conference on Machine Learning (2006)
17. Hirschmller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007)
18. Davies, E.: Laws' texture energy in texture. In: Machine Vision: Theory, Algorithms, Practicalities, 2nd edn. Academic Press, San Diego (1997)