Learning Output-Kernel-Dependent Regression for Human Pose Estimation

Weiwei Guo weiwei.guo@elec.qmul.ac.uk Ioannis Patras i.patras@elec.qmul.ac.uk Multimedia and Vision Group Queen Mary, University of London London, UK

Abstract

This paper presents a novel method for continuous inter-dependent outputs prediction that predicts each of the multiple output variables based not only on the input but also on the rest of the outputs. We do so by using for each output kernel regression functions that are a convex combination of two kernels: the first kernel defined over the input and the second over the remaining outputs. We propose a scheme in which the relative importance of these two heterogeneous information sources is learned jointly with the parameters of the regression scheme. The inference for a new observation is obtained by fix-point iteration which is based on alternatively evaluating the learned outputdependent functions until convergence. We experimentally validate our proposed method on 3D human pose estimation problem using the HumanEva-I dataset.

1 Introduction

Discriminative approaches to human pose estimation involve learning a mapping from visual observations to articulated body configurations given a labelled training set comprising of an input image or its descriptor **x** and the corresponding 3D human pose **y**, i.e., $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$. Such discriminative learning-based methods have recently received considerable attentions due to their simplicity, computation efficiency and the fact that they can predict human pose from a single image without initialization.

In the context of human pose estimation, both the input (image descriptor) and the output (pose) are high-dimensional vectors but also strongly correlated. The dependencies between output variables arise from body kinematic structure and physical constraints that can not occur in arbitrary configurations. For instance, the positions of the elbow centres are dependent on the positions of the shoulders, the ankles dependent on the position of knees etc. Unfortunately, most of current discriminative methods learn separate scalar-output mapping functions for each output element that dependent only on the input. Thus they neglect the dependencies between the output elements a fact that may degrade the estimation accuracy.

In this paper, we introduce a novel regression model that exploits such inter-dependencies of outputs, named Learning Output-Kernel-Dependent Regression (LOKDR). The basic idea is that each of output variables is not only regressed with the input image descriptors but also dependent on the remaining outputs. The basic idea is similar to the one presented by Bo and Sminchisescu [1], but our model has two appealing differences. First, it has the ability of

learning the relative importance of these two heterogeneous contributions to the prediction of each of the outputs automatically as part of training process. By contrast, it is learned as a post-processing step by time-consuming cross validation in [2]. Second, the inference process is a simpler fix-point iteration procedure that evaluates the learned regression models iteratively until convergence replacing the BFGS quasi-Newton optimization algorithm used in [2].

The rest of the paper is organized as follows: we review the related work in Section 2 and describe in detail the proposed model in Section 3. In Section 4 we present the experimental results. Finally, Section 5 concludes this paper.

2 Related work

Mappings between observations and body poses have been learned by either Bayesian Mixture of Experts (BME) [[]], Gaussian Process (GP) regression [[]], or Relevance Vector Machine (RVM) []. However, these discriminative methods build individual scalar-output mappings for each of the output variables disregarding the fact that the multiple outputs are inter-dependent and structured. On the other hand, structured output prediction methods have recently inspired methods in natural language parsing (NLP), image segmentation and understanding and articulated object parsing [[], [], [], []].

Often explicit graphic models like Conditional Random Fields (CRF) or Markov Random Fields (MRF) [I] are used to model the inter-dependencies between outputs. Maximal marginal SVM-like algorithms for the discrete inter-dependent outputs have been proposed by Tsochantaridis *et al.* [II]. The continuous outputs version is extended for object detection and body pose estimation in [I]. Aside from explicitly modeling the relations between outputs, Weston *et al.* [II] presented Kernel Dependency Estimation (KDE) that implicitly models the correlations by similarity measure. KDE decouples output correlations by first applying Kernel PCA over the outputs and then learns the mappings from the input space to dimension-reduced space by ridge regression. Bo and Sminchisescu [I] proposed Twin Gaussian Process (TGP) for structured continuous output prediction that employed Kullback-Leibler divergence-based dependency measure.

3 Proposed model

Our LOKDR method models the output inter-dependency implicitly by associative kernels over outputs. More specifically, for predicting each of the outputs y^t , we define two kernels one over the original input **x** and one over the rest of the outputs y^{-t} . This allow both the input and the other outputs contribute to the prediction of y^t in a single discriminative kernel regression. In addition, it allows learning the relative importance of each contributions automatically.

As illustrated in Figure 1, the scalar output y^t is not only related to the input vector **x** but also related to the remaining outputs $\mathbf{y}^{-t} = [y^1, \dots, y^{t-1}, y^{t+1}, \dots, y^d]^T$. Following [**1**], each output variable y^t can be predicted from the augmented feature vector $\tilde{\mathbf{x}}^t = [\mathbf{x}, \mathbf{y}^{-t}]^T$, where the \mathbf{y}^{-t} contributes to the prediction of the output y^t as auxiliary features. For kernel regression, we define a joint kernel over the augmented features. The proposed kernel is a convex combination of two kernels, one defined over the input vector \mathbf{x} and the other over

the rest the outputs \mathbf{y}^{-t} . Formally,

$$k(\tilde{\mathbf{x}}_1^t, \tilde{\mathbf{x}}_2^t) = \beta_1^t k_1(\mathbf{x}_1, \mathbf{x}_2) + \beta_2^t k_2(\mathbf{y}_1^{-t}, \mathbf{y}_2^{-t})$$
(1)

with $\beta_1^t, \beta_2^t \ge 0, \beta_1^t + \beta_2^t = 1$. In the kernel regression framework [II], given a set of training pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ we learn the dependency mapping from $\tilde{\mathbf{x}}^t$ to y^t as

$$y^{t} = f^{t}(\tilde{\mathbf{x}}^{t}) = \sum_{i=1}^{N} \gamma_{i}^{t} k(\tilde{\mathbf{x}}^{t}, \tilde{\mathbf{x}}_{i}^{t}) + b^{t}, \qquad (2)$$

 $\forall t \in \{1, \dots, d\}$, where $\{\gamma_i^t\}_{i=1}^N, \{\beta_1^t, \beta_2^t\}$ are the unknowns. In what follows we explain how to find the unknowns simultaneously during training in a *multiple kernel leaning* framework **[B**, **[D**].

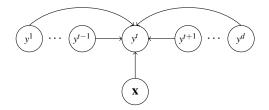


Figure 1: Graphical model for proposed regression method

3.1 Learning

For clarity of notation, and without loss of generality we drop the superscript *t*. Following the support vector machine (SVM) framework $[\square]$ we obtain the optimal values of the parameters $\{\alpha_i\}_{i=1}^N, \{\beta_1, \beta_2\}$ as the dual solutions of the following optimization problem:

$$\min \quad \frac{1}{2} \sum_{k=1}^{2} \frac{1}{\beta_{k}} \|\mathbf{w}_{k}\|^{2} + C \sum_{i=1}^{N} (\xi_{i} + \hat{\xi}_{i}) \\
\text{w.r.t.} \quad \{\mathbf{w}_{1}, \mathbf{w}_{2}\}, \{\beta_{1}, \beta_{2}\}, \{\xi_{i}, \hat{\xi}_{i} \ge 0\}|_{i=1}^{N}, b \\
\text{s.t.} \quad y_{i} - \sum_{k=1,2} f_{k}(\tilde{\mathbf{x}}_{i}) - b \le \varepsilon + \xi_{i} \\
y_{i} - \sum_{k=1,2} f_{k}(\tilde{\mathbf{x}}_{i}) - b \ge -\varepsilon - \hat{\xi}_{i} \\
f_{1}(\tilde{\mathbf{x}}) = \mathbf{w}_{1}^{T} \phi_{1}(\mathbf{x}), f_{2}(\tilde{\mathbf{x}}) = \mathbf{w}_{2}^{T} \phi_{2}(\mathbf{y}^{-t}) \\
\beta_{1} + \beta_{2} = 1, \beta_{1}, \beta_{2} \ge 0$$
(3)

Following [**D**], we eliminate one of $\{\beta_1, \beta_2\}$ and reformat this primal as a nested two-step optimization problem as follows:

$$\min_{\beta} \quad T(\beta) \quad \text{s.t} \quad 0 \le \beta \le 1^1 \tag{4}$$

where,

$$T(\beta) = \begin{cases} \min & \frac{1}{2} (\frac{1}{\beta} \| \mathbf{w}_{1} \|^{2} + \frac{1}{1-\beta} \| \mathbf{w}_{2} \|^{2}) \\ + C \sum_{i=1}^{N} (\xi_{i} + \hat{\xi}_{i}) \\ \text{w.r.t} & \mathbf{w}_{1}, \mathbf{w}_{2}, \{\xi_{i}, \hat{\xi}_{i} \ge 0\}|_{i=1}^{N}, b \\ \text{s.t.} & y_{i} - f_{1}(\tilde{\mathbf{x}}_{i}) - f_{2}(\tilde{\mathbf{x}}_{i}) - b \le \varepsilon + \xi_{i} \\ y_{i} - f_{1}(\tilde{\mathbf{x}}_{i}) - f_{2}(\tilde{\mathbf{x}}_{i}) - b \ge -\varepsilon - \hat{\xi}_{i} \\ f_{1}(\tilde{\mathbf{x}}) = \mathbf{w}_{1}^{T} \phi_{1}(\mathbf{x}), \\ f_{2}(\tilde{\mathbf{x}}) = \mathbf{w}_{2}^{T} \phi_{2}(\mathbf{y}^{-t}) \end{cases}$$
(5)

Setting to zero the derivatives of the Lagrangian of the problem (3) with respect to the primal variables and substituting KKT optimal conditions in the Lagrangian gives the following dual formation over the Lagrangian multipliers (i.e. the dual variables) α and $\hat{\alpha}$:²

$$T(\boldsymbol{\beta}) = \begin{cases} \max_{\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}} & -\frac{1}{2} \sum_{i,j} (\boldsymbol{\alpha}_i - \hat{\boldsymbol{\alpha}}_i) (\boldsymbol{\alpha}_j - \hat{\boldsymbol{\alpha}}_j) k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) \\ & -\varepsilon \sum_i (\boldsymbol{\alpha}_i + \hat{\boldsymbol{\alpha}}_i) + \sum_i (\boldsymbol{\alpha}_i - \hat{\boldsymbol{\alpha}}_i) y_i \\ \text{s.t.} & \sum_i (\boldsymbol{\alpha}_i - \hat{\boldsymbol{\alpha}}_i) = 0 \\ & 0 \le \boldsymbol{\alpha}_i, \hat{\boldsymbol{\alpha}}_i \le C \end{cases}$$
(6)

where $k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) = \beta k_1(\mathbf{x}_i, \mathbf{x}_j) + (1 - \beta)k_2(\mathbf{y}_i^-, \mathbf{y}_j^-)$. Once the the optimal values $\{\alpha_i^*, \hat{\alpha}_i^*\}|_{i=1}^N$ of the Lagrangian of dual Problem (6) are determined, the unknowns γ in Equation (2) can be obtained as $\gamma_i = (\alpha_i^* - \hat{\alpha}_i^*), \forall i$.

Thus, $T(\beta)$ of the inner problem (5) can be obtained by a standard SVM implementation such as LibSVM [**D**]. The outer problem (4) can be solved by projected gradient descent algorithm where the derivative of $T(\beta)$ with respect to β is computed as

$$d_{\beta} = \frac{dT(\beta)}{d\beta} = -\frac{1}{2} \sum_{i,j} (\alpha_i^* - \hat{\alpha}_i^*) (\alpha_j^* - \hat{\alpha}_j^*) \\ (k_1(\mathbf{x}_i, \mathbf{x}_j) - k_2(\mathbf{y}_i^-, \mathbf{y}_j^-)).$$

$$(7)$$

For the constraint $0 \le \beta \le 1$, the projected gradient descent is given by

$$\boldsymbol{\beta}^{(k+1)} \leftarrow \max\left(0, \min(1, \boldsymbol{\beta}^{(k)} + \boldsymbol{\eta}^{(k)} \boldsymbol{d}_{\boldsymbol{\beta}^{(k)}})\right)$$
(8)

where $\eta^{(k)}$ is the step size which is determined by a one-dimensional linear search. The algorithm stops at the optimum when

$$\left|\frac{dT(\beta)}{d\beta}\right| \le \varepsilon. \tag{9}$$

which approximates the first-order KKT optimality condition $[\square]$. The learning process is summarized in Algorithm 1.

²We omit the full derivations due to lack of space

3.2 Inference

Once the model is trained, the inference for a new observation becomes

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}} \sum_{t} (y^{t} - f^{t}(\tilde{\mathbf{x}}^{t}))^{2}.$$
(10)

In contrast to $[\square]$, we do not use gradient-descent algorithms to optimize **y**. Instead, we adopt a simpler fix-point iteration procedure that in our experiments converged a little faster to a similar optima. Let us define the vector-valued function $\mathbf{f} = [f^1, \dots, f^d]^T$. Then, the fix-point iteration is performed as $\mathbf{y}^{(n+1)} = \mathbf{f}(\mathbf{x}, \mathbf{y}^{(n)})$. We initialize $\mathbf{y}^{(0)}$ using learned output-independent (i.e. only dependent on the input **x**) prediction functions for each output dimension. The inference process is summarized in Algorithm 2.

Algorithm 1 Learning

- 1: Initialize β randomly or as 1/2
- 2: while stopping criteria Eq.(9) not met do
- 3: Compute the kernel matrix as $k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$
- 4: Use an SVM optimizer to obtain α , $T(\beta)$
- 5: Perform projected gradient descent for Problem (4)
- 6: end while

Algorithm 2 Inference

1: $n \leftarrow 0$ 2: Initialize $\mathbf{y}^{(0)}$ 3: repeat 4: for t = 1 to d do 5: $y^{t(n+1)} = f^t(\mathbf{x}, \mathbf{y}^{-t(n)})$ 6: end for 7: $n \leftarrow n+1$ 8: until $\|\mathbf{y}^{(n+1)} - \mathbf{y}^{(n)}\| < \varepsilon$

4 Experimental results

We validate our method on the HumanEva-I benchmark dataset [III] which comprises of 3 subjects performing multiple activities. We use the Walking, Jogging, Gestures and Box sequences observed by Camera C1 (8607 valid frames in total) where Table 1 lists the number of frames of each sequence. Each sequence is divided in half for training and testing, respectively. As in [I], [II], to speed up computations and to compare with other regression models, we perform local regression based on a reduced set provided by the K-nearest neighbors (K = 25). The training set consists of training sub-sequences of all motions and subjects and the test is performed on the testing subset.

Following [\blacksquare], we use an image descriptor vector **x** with 270 dimensions that is based on Histograms of Oriented Gradients (HoG) that are extracted from the silhouette resulted from background substraction. To obtain the HoGs, the silhouette bounding box is divided into a 6 × 5 grid and gradient orientations in each cell grid are quantized into 9 orientation

Action	S1	S2	S3	Total
Walking	1176	876	895	2947
Jog	439	795	831	2065
Gestures	801	681	214	1696
Box	502	464	933	1899
Total	2918	2816	2873	8607

Table 1: The number of frames of each sequence in the dataset

bins. The 57 dimensional pose vector \mathbf{y} encodes the relative 3D positions of 19 joint centers defined relative to "torsoDistal". The estimation error (in mm) is measured as the average Euclidean distance between the estimated joint position $\mathbf{y}_i^{(n)} \in \mathbb{R}^3$ and the true $\bar{\mathbf{y}}_i^{(n)} \in \mathbb{R}^3$ over all M joints and N frames [III]. That is, $Err = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{M} \sum_{i=1}^{M} ||\mathbf{y}_i^{(n)} - \bar{\mathbf{y}}_i^{(n)}||$. We use Gaussian kernels with kernel widths set to the median distance of nearest neighbours.

Table 2 shows the average estimation errors for all of the 3 subjects. For comparison, we present the results of WKNN (Weighted K-nearest neighbors), SVR (Support Vector Regression) and RVM (Relevance Vector Machine) for independent outputs as well as Bo's SOAR and our LOKDR models. It can be seen that our output-dependent trained LOKDR model performs better than output-independent trained models.

In order to compare our model that learns β with [\square] that keeps it fixed we present results on the walking sequences (sampled every 5 frames). In Figure 2 we present results by varying the β ($\beta = 1$ means that the model is trained output-independently). While [\square] set β using cross validation, our model learns it jointly with the other unknowns. This allows us to learn a different β for each local (i.e. defined over the k nearest neighbours) regressor and therefore achieve lower error that the one obtain for the optimal value of β for the dataset.

Motion	WKNN	SVR	RVM	SOAR [Our LOKDR
Walking	81.55	59.82	61.91	56.00	55.87
Jogging	82.55	57.37	58.11	54.18	54.28
Gestures	59.68	50.02	50.33	46.70	46.29
Box	74.90	66.17	66.46	64.54	64.52
Average	74.67	58.35	59.20	55.36	55.24

Table 2: Average estimation errors of WKNN, SVR, RVM, SOAR and our LOKDR models.

5 Conclusions

We have described a new model for output-dependent prediction called Learning Output-Kernel-Dependent Regression (LOKDR). In this model, the prediction of each output variable is based on two information sources: the original input and the remaining outputs. One appealing property of our model is that it can learn the relative importance of the two contributions automatically as part of learning process. Validation of our model is shown for human pose estimation task in Human-I dataset.

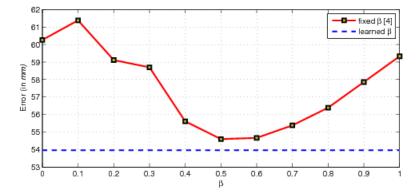


Figure 2: Average errors at different β , not being leaned compared with the learned β on Walking sequence performed by S1

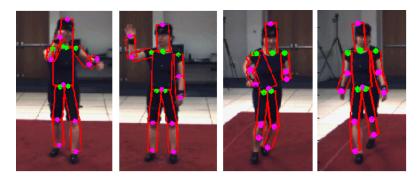


Figure 3: Samples of pose estimation

References

- [1] Learning with kernels: Support vector machines, regularization, optimization, and beyond. the MIT Press, 2002.
- [2] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *TPAMI*, 28(1):44–58, 2006.
- [3] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *IJCV*, 87:28–52, 2010.
- [4] Liefeng Bo and C. Sminchisescu. Structured output-associative regression. In *CVPR*, June 2009. Miami, USA.
- [5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. Technical report, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm.
- [6] C. Ionescu, L. Bo, and C. Sminchisescu. Structural svm for visual localization and continuous state estimation. In *ICCV*, September 2009. Kyoto, Japan.

8 WEIWEI GUO, IOANNIS PATRAS: LOKD REGRESSION FOR HUMAN POSE ESTIMATION

- [7] John D. Lafferty, Andrew Mccallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, June 2001. San Francisco, CA, USA.
- [8] R. Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. In *CVPR 2nd Workshop on EHuM*, June 2007. Minnesota, USA.
- [9] Alain Rakotomamonjy, Francis R.Bach, Stephane Canu, and Yves Grandvalet. Simplemkl. *JMLR*, 9:2491–2531, 2008.
- [10] Leonid Sigal, Alexandru Balan, and Michael Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, August 2009.
- [11] C. Sminchisescu, A. Kanaujia, and D. N. Metaxas. Bm3e : Discriminative density propagation for visual tracking. *TPAMI*, 29(11):2030–2044, 2007.
- [12] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6: 1453 1484, 2005.
- [13] R. Urtasun and T. Darrell. Sparse probabilistic regression for activity-independent human pose inference. In CVPR, June 2008. Alaska, USA.
- [14] Manik Varma and Bodla R. Babu. More generality in efficient multiple kernel learning. In *ICML*, Dec. 2009. Montreal, Quebec, Canada.
- [15] J. Weston, O. Chapelle, A. Elisseeff, B. Schölkopf, and V. Vapnik. Kernel dependency estimation. In *NIPS*, Dec. 2002. Canada.