Learning Equivariant Structured Output SVM Regressors

Andrea Vedaldi, Matthew B. Blaschko, Andrew Zisserman

Learning Equivariant Structured Output SVM Regressors

Andrea Vedaldi Matthew Blaschko Andrew Zisserman

Department of Engineering Science
University of Oxford
{vedaldi,blaschko,az}@robots.ox.ac.uk

Abstract

Equivariance and invariance are often desired properties of a computer vision system. However, currently available strategies generally rely on virtual sampling, leaving open the question of how many samples are necessary, on the use of invariant feature representations, which can mistakenly discard information relevant to the vision task, or on the use of latent variable models, which result in non-convex training and expensive inference at test time. We propose here a generalization of structured output SVM regressors that can incorporate equivariance and invariance into a convex training procedure, enabling the incorporation of large families of transformations, while maintaining optimality and tractability. Importantly, test time inference does not require the estimation of latent variables, resulting in highly efficient objective functions. This results in a natural formulation for treating equivariance and invariance that is easily implemented as an adaptation of off-the-shelf optimization software, obviating the need for ad hoc sampling strategies. Theoretical results relating to vicinal risk, and experiments on challenging aerial car and pedestrian detection tasks show the effectiveness of the proposed solution.

1. Introduction

In applications such as object detection and object classification, the output changes in a predictable way to certain transformations of the input images. For instance, if the image is rotated then the location output of an object detector should also move accordingly, i.e. the object location is *equivariant* with the image rotation. On the other hand, whether an image contains a certain object does not depend on the image rotation, i.e. the label output of an object classifier is rotation *invariant*. In both cases rotation is a *nuisance factor* that, by affecting the appearance of the object, complicates extracting the information of interest, the location or presence of the object.

A common way of handling nuisance transformations is to explicitly model and estimate them as *latent* factors [6, 32]. In our example, this amounts to estimating the rotation of each object both at training (design) and testing (application) time. Doing so has two significant drawbacks: (i) computations are wasted in estimating irrelevant information (the object rotation) and (ii) the learning problem becomes non-convex due to the latent factors [32].

In this paper we develop instead invariant/equivariant algorithms that (i) estimate only the information of interest (e.g. *not* rotation) and (ii) are learned by solving a convex optimization problem. By *avoiding the introduction of latent factors*, these methods can improve the efficiency of learning and, more importantly, testing.

In classification problems, the goal is usually to train invariant classifiers. There is a large body of literature dealing with this problem [14]. Invariance has been enforced or encouraged at the level of (a) the training data by generating virtual samples [21] (for example for pose invariant keypoint recognition [15] or for tolerance to lighting and small pose changes in the case of object detection [13]); (b) the data representation (tangent distance [26], jittered, tangent, and invariant kernels [25, 22, 30, 7]); and (c) the learning objective (vicinal risk minimization [3], invariance in learning distance functions [12]).

Problems such as object detection, image segmentation, and image parsing cannot be encoded naturally as classification problems. In these cases one may use structured output learning [27, 29], which allows the learning of functions with complex outputs, such as object poses, segmentations, and parse trees. In this paper we propose *a method to incorporate invariance or equivariance [24] into structured learning problems*. For training we utilize a cutting plane strategy, which efficiently and optimally generates samples that incorporate desired invariance and equivariance. Our method can be seen as an extension of [28] to the case of equivariant learning.

Our approach unites the approaches (a–c) above in a single formulation by generalizing structured output regression learning. For example, in the case (a) of generating virtual samples, it is not clear *a priori* how many samples need to be generated and how dense they should be. The structured

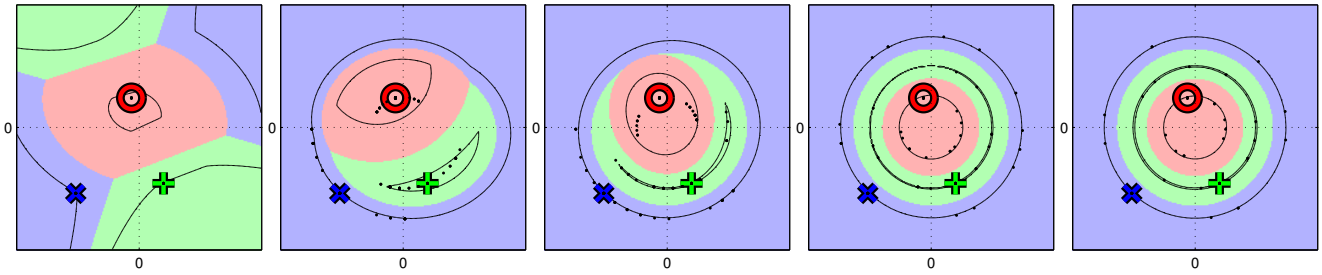

Figure 1: **Illustrative example.** We use the structured and invariant SVM to learn a polynomial SVM for the problem of separating three classes of 2-D dots (red, blue, and green) by sampling a single point (denoted with respectively a circle, an 'x' sign, and a cross) from each class (see the text for the formulation details). We impose invariance to rotations of up to respectively 0, $\pi/8$, $\pi/4$, $\pi/2$ and π radians. The estimated classes are indicated by the colored areas, the black lines represent the margin, and the black dots the set of transformed samples added by the cutting plane iterations. Note that the selection of such "virtual samples" is sparse.

learning formulation deals with this in a principled manner by only generating those samples necessary to optimize the objective function. Importantly, this benefit is achieved without necessitating or precluding the use of invariant feature representations, which may inadvertently discard information relevant to the task.

1.1. Regularized risk minimization

Let $\mathcal{X}$ denote the input space (e.g. natural images) and $\mathcal{Y}$ the output space (e.g. object locations). In standard structured output learning, given training data $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, N$ and a parametric form $f(x; w)$ of the function $\mathcal{X} \to \mathcal{Y}$ to be estimated (e.g. the object detector), one minimizes a regularized empirical risk estimate of the form

$$\min_w \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N} \Delta(y_i, f(x_i; w)). \qquad (1)$$

The loss $\Delta(y_i, \widehat{y})$ measures how well the prediction $\widehat{y} = f(x_i; w)$ approximates the desired output y_i (e.g. the overlap error between the estimated and predicted object bounding box [1]). The regularization term $\|w\|^2$ penalizes overly complex models and it is traded off with the empirical average loss (risk) by the parameter $C > 0$.

In this work, we modify the problem (1) in order to penalize deviations of $f(x; w)$ from specified invariance or equivariance requirements. The idea is illustrated in Fig. 1 for the problem of invariant multi-class classification with hinge loss. Here we learn to discriminate between three different classes $\mathcal{Y} = \{\text{red}, \text{green}, \text{blue}\}$, which are subsets of $\mathcal{X} = \mathbb{R}^2$. The classes have circular symmetry, but only *one training point* is given for each (the red circle, green cross, and blue x). In the first panel, no invariance is enforced, and the estimated classes are incorrect. In the second panel, invariance to rotations t in the range $\mathcal{T} = [-\pi/8, +\pi/8]$ is enforced. This is done by maximizing the loss in (1) with respect to the transformation by considering

$$\min_w \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N} \sup_{t \in \mathcal{T}} \Delta(y_i, f(tx_i; w)). \qquad (2)$$

The blue class in the second panel is now roughly circular. By further increasing the range of invariant rotations to $\pm\pi/4$, $\pm\pi/2$, and $\pm\pi$ radians, circular symmetry is enforced more and more accurately (and with a principled choice of the transformed samples).

The first technical contribution of the paper is to show that costs such as (2) can be formulated as convex optimization problems (Sect. 2.1). In particular, our method extends previous approaches such as [17, 28] to handle *equivariant transformations* affecting the input and the output simultaneously. This is a very flexible framework that can be used to solve a variety of different problems, ranging from optimal ranking to object detection (Sect. 3), by the use of established large-scale optimization techniques (Sect. 2.2).

Representing large transformations of the inputs often requires the use of non-linear kernels, which are known to be slow, especially in the context of structured output learning. Thus in Sect. 3.1 we adopt a class of non-linear joint-kernels similar to [33], dubbed *slot kernels*, that are non-linear and local similar to a Gaussian kernel but are much more computationally efficient. Finally, in App. A we show that, by an appropriate choice of the loss function, the proposed formulation has a probabilistic interpretation in term of vicinal risk minimization [3].

2. Equivariant structured learning

In the following Sect. 2.1 first introduces the equivariant structured learning problem, extending (2) to the case of equivariant and invariant structured outputs. Then, Sect. 2.2 derives a corresponding equivariant structured SVM formulation, yielding a *convex* optimization problem, and shows

how standard cutting-plane solvers [11] can be used to efficiently obtain a solution.

2.1. General formulation

Our goal is to learn an equivariant or invariant function $f(x; w)$. We start by giving a formal definiton of equivariance. Consider pairs of corresponding transformations $t = (t_{\mathcal{X}}, t_{\mathcal{Y}}) \in \mathcal{T}$ of the input $t_{\mathcal{X}} : \mathcal{X} \to \mathcal{X}$ (e.g. image rotations) and of the output $t_{\mathcal{Y}} : \mathcal{Y} \to \mathcal{Y}$ (e.g. bounding box rotations). The learned function $f(x; w)$ is *equivariant* with $\mathcal{T}$ if $f(t_{\mathcal{X}} x) = t_{\mathcal{Y}} f(x)$ for all $t \in \mathcal{T}$. For instance, the object location $f(t_{\mathcal{X}} x; w)$ predicted on the rotated image $t_{\mathcal{X}} x$ should be equal to the rotated location $t_{\mathcal{Y}} f(x; w)$ predicted from the original image x. As a special case, fixing $t_{\mathcal{Y}} = \mathbf{1}$ to be the identity transformation encodes invariance: $f(t_{\mathcal{X}} x; w) = f(x; w)$. In order to simplify notation the shorthand tx and ty will denote the action of the transformation $t \in \mathcal{T}$ on the input x and the output y respectively.

As in Sect. 1.1, a way of encouraging equivariance is to penalize the maximum loss $\Delta(ty_i, f(tx_i; w))$ with respect to all transformations $t \in \mathcal{T}$. To allow weighting the transformations, we generalize this and allow the loss to depend directly on t and consider instead

$$\sup_{t \in \mathcal{T}} \Delta(t, y_i, f(tx_i; w)) \tag{3}$$

where the dependency on ty_i is implicit. In App. A this flexibility is used to give a probabilistic interpretation of (3) in term of vicinal risk [3]. By substituting the loss (3) into (2) one obtains the equivariant structured learning problem

$$\min_w \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N} \sup_{t \in \mathcal{T}} \Delta(t, y_i, f(tx_i; w)). \tag{4}$$

The loss can be any function such that $\Delta \geq 0$ and $\Delta(\mathbf{1}, y, y) = 0$ [11]. Hence (4) is typically non-convex and very hard to solve. Sect. 2.2 gives its relaxation to a convex formulation.

2.2. Convex formulation

The first step in deriving a convex variant of (4) is to choose an appropriate parameterization for $f(x; w)$. As in a structured output SVM, we define $f(x; w)$ through a joint feature map $\Psi(x, y)$ to map the input-output pair (x, y) into a linear feature space (the feature map can be defined implicitly by a kernel function $K(x, y, x', y') = \langle \Psi(x, y), \Psi(x', y') \rangle$). Then $\widehat{y} = f(x; w)$ is defined as the output $\widehat{y}$ which maximizes the input-output compatibility score $\langle w, \Psi(x, y) \rangle$, linearly parameterized in the weight vector w, i.e.

$$f(x; w) = \arg\max_{\widehat{y} \in \mathcal{Y}} \langle w, \Psi(x_i, \widehat{y}) \rangle. \tag{5}$$

For this parameterization, it is easy to derive a convex upper bound to (3). For any fixed $t \in \mathcal{T}$ one has

$$\begin{aligned}\Delta(t, y_i, f(tx_i; w)) \leq \Delta(t, y_i, f(tx_i; w)) \times \\ [1 + \langle w, \Psi(tx_i, f(tx_i; w)) \rangle - \langle w, \Psi(tx_i, ty_i) \rangle]\end{aligned}$$

because, due to the maximization in (5), $\langle w, \Psi(tx_i, f(tx_i; w)) \rangle \geq \langle w, \Psi(tx_i, ty_i) \rangle$ and the quantity in brackets is greater than or equal to 1. By substituting $f(tx_i; w)$ with $\widehat{y}$ and by further maximizing the right hand side with respect to $\widehat{y}$ we obtain the desired upper bound

$$\begin{aligned}\Delta(t, y_i, f(tx_i; w)) \leq \sup_{\widehat{y} \in \mathcal{Y}} \Delta(t, y_i, \widehat{y}) \times \\ [1 + \langle w, \Psi(tx_i, \widehat{y}) \rangle - \langle w, \Psi(tx_i, ty_i) \rangle].\end{aligned}$$

Plugging this bound back into (4) yields

$$\begin{aligned}\min_w \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N} \sup_{(t, \widehat{y}) \in \mathcal{T} \times \mathcal{Y}} \Delta(t, y_i, \widehat{y}) \times \\ [1 + \langle w, \Psi(tx_i, \widehat{y}) - \Psi(tx_i, ty_i) \rangle].\end{aligned} \tag{6}$$

We call this the *equivariant structured SVM problem*. Notice that the standard structured SVM formulation [11] is recovered by setting $\mathcal{T} = \{1\}$, where 1 is the identity transformation. Note also that, in contrast to [6, 32], (6) *does not introduce latent factors and remains convex.*

Efficient optimization. The convex program (6) can be converted to the so called one-slack formulation [11], where the sup operator is translated into a large set (possibly infinite) of linear constraints:

$$\begin{aligned}&\min_{w, \xi} \frac{1}{2}\|w\|^2 + C\xi, \quad \text{s.t. } \forall\, t \in \mathcal{T}^N, \widehat{y} \in \mathcal{Y}^N \\ &\xi \geq \frac{1}{N}\sum_{i=1}^{N} \Delta(t_i, y_i, \widehat{y}_i)[1 + \langle w, \Psi(t_i x_i, \widehat{y}_i) - \Psi(t_i x_i, t_i y_i) \rangle].\end{aligned} \tag{7}$$

The problem (7) can be optimized using standard off-the-shelf solvers. These solvers handle the large number of constraints in (7) by exploiting the fact that, usually, only a fraction of them is needed to characterize the objective function around the optimum. For our model, finding these constraints amounts to *identifying a small set of useful virtual samples*. The algorithm can be summarized as follows:

1. Solve the problem (7) by considering only a subset S of constraints (initially this subset is empty). Obtain an estimate w of the model.

2. Given the current estimate w of the model, obtain the next constraint by solving for each $i = 1, \ldots, N$ the prob-

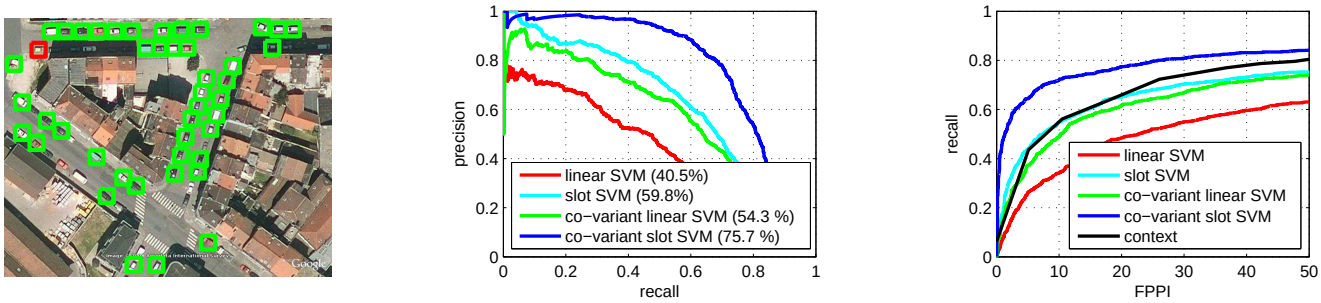

Figure 2: **Aerial car detector.** *Left:* example detections (correct in green, incorrect in red). *Middle:* precision-recall curves in the style of the PASCAL VOC challenge [5] for the linear and slot kernel structured SVM detectors and their equivariant versions. *Right:* FPPI vs recall curves for the different methods, including the method from [8] denoted as "context".

lems

$$(t_i, \widehat{y}_i)^* = \operatorname*{argmax}_{t \in \mathcal{T}, \widehat{y} \in \mathcal{Y}} \Delta(t, y_i, \widehat{y}) \times [1 + \langle w, \Psi(tx_i, \widehat{y}) - \Psi(tx_i, ty_i) \rangle] \quad (8)$$

Generating this constraint can be interpreted as selecting the next N most difficult virtual samples.

3. Add the new one-slack constraint to S and repeat from 1 until convergence.

[11] gives strong guarantees on the efficiency (in terms of number of iterations) and accuracy of this method. As any structured output learning method, the overall efficiency depends on the cost of computing the maximum (8), which depends on the nature of the data and the structure of the transformations. Many methods exist for performing efficient inference for a number of special cases of structured prediction [1, 6, 8, 10, 27, 29]. The maximization in (8) is no more expensive than a structured SVM or latent SVM at training time, but the resulting test time inference is much more efficient as there is no inference of a latent factor. Furthermore, this method supports continuous classes of transformations, e.g. through gradient ascent methods, which can be handled only approximately by generating virtual samples.

3. Experiments

3.1. Rotation invariant object detection

We use the equivariant (rather than invariant) structured SVM formulation to incorporate invariance to arbitrary image rotations to an object detector in the style of [1]. The input to the structured SVM $\widehat{y} = f(x; w)$ is an image $x \in \mathcal{X}$ and the output $\widehat{y}$ is either the 2D location of the object of interest, or a flag indicating that the object is not contained in the image. Additionally, we require that the prediction $\widehat{y} = f(x; w)$ is consistent with arbitrary rotations of the image. This is encoded as a equivariance requirement: if the object is found at location y in image x, then the same object must be found at the rotated location ty in the rotated image tx.

Note that the rotation of the object is not of interest here, and in fact the detector will avoid estimating it; nevertheless, the location should be correctly determined regardless of the orientation of the object, which still affects its appearance.

Aerial car detection. As an example application, we consider the task of aerial car detection proposed in [8] (Fig. 2). The data consists of 30 aerial images with cars annotated (for a total of more than 1,000 cars with varying rotations). The performance of the detectors is measured according to the PASCAL criterion [5] (average precision-recall) and the criterion used in [8] (number of false positives per image) for a direct comparison. As low-level image features we use the HOG [4] implementation of [6] with cells of 5×5 pixels. A car is described by a block of 7×7 HOG cells.

We consider as transformations $t \in \mathcal{T}$ the set of rotations in the range $[0, 2\pi)$. t acts on the image x and on the object location y by rotating them by the same amount, so that they stay "aligned". The SVM kernel is the restriction kernel proposed by [1]. In term of joint feature maps, this kernel is given by a function $\phi(x, y)$ that returns the HOG descriptor of a block of 7×7 HOG cells extracted at location y. With this choice of the feature map, evaluating the structured SVM $f(x; w) = \operatorname{argmax}_y \langle w, \phi(x, y) \rangle$ is similar to running a sliding window detector based on a linear SVM and HOG features [4]. The loss function $\Delta(t, y_i, \widehat{y}) \in \{0, 1\}$ is also similar to [1] and is equal to zero if the predicted location $\widehat{y}$ is close enough to the transformed ground truth location ty_i, or if y_i and $\widehat{y}$ agree that no object is contained in the image.

Slot kernels. A linear HOG model is not sufficient to capture arbitrary object rotations. These could be handled by switching to a non-linear kernel such as a Gaussian, but non-linear kernels slow down structured SVMs significantly. Even approximated feature maps for the Gaussian kernel such [31, 16, 23] are slow for object detection as they require projecting each candidate image patch on a large set of basis of vectors.

This motivates us to utilize *slot kernels*, which are local

as the Gaussian kernels but are much more efficient. Let $q(\phi(x,y)) \in \{1,\dots,Q\}$ be a function that assigns one out of Q discrete labels to the local HOG descriptor $\phi(x,y)$. For efficiency we implement q with a KD-tree and we design the Q partitions using k-means. Then the slot kernel is given by the feature map $\Psi(x,y) = e_{q(\phi(x,y))} \otimes \phi(x,y)$, where e_q is the q-th element of the canonical basis of $\mathbb{R}^Q$. The feature map $\Psi(x,y)$ is a collection of Q linear models, only one of them being active at a time as indicated by the function q (this makes the kernel local). This idea is similar to a mixture of experts and, in the context of kernel learning, to [33, 20]. In our experiments we set $Q = 18$.

Results. Fig. 2 compares the various methods. The standard linear structured SVM detector performs relatively poorly. Adding equivariance to rotations improves performance significantly (+14% Average Precision), and using the non-linear slot kernel in place of the linear one is even better (+20% AP). However, the largest benefit by far is obtained by combining the non-linear kernel with rotation equivariance (+35% AP), illustrating the importance of being able to correctly handle image transformations and the need for using a non-linear representation to do so. We also note that this detector performs significantly better than the detector proposed originally by [8], despite the fact that their approach makes use of a sophisticated contextual model to aid discrimination.

3.2. Learning to rank with invariance

Learning to rank is a popular application of structured output SVMs [10, 2]. In these experiments we evaluate a structured SVM that simultaneously optimizes ranking and is invariant to a class of transformations of the input. We begin by first deriving an invariant binary SVM formulation that does not incorporate ranking.

Invariant binary SVM. The input to the binary SVM $\widehat{y} = f(x;w)$ is an object $x \in \mathcal{X}$ and the output $\widehat{y} \in \{-1,+1\}$ is its label. The quality of the prediction is measured in term of the standard 01-loss $\Delta(t, y_i, \widehat{y}) = (1 - y_i\widehat{y})/2$. The goal is to perform consistently well up to a class of transformations $\mathcal{T}$ of the input (the transformations do not affect the output here). Let $\phi(x)$ be a feature of the input x and define the joint feature map $\Psi(x,y) = y/2 \begin{bmatrix}\phi(x)^\top, & B\end{bmatrix}^\top$, where $B > 0$ is a constant used as bias [11]. From (5), evaluating this structured SVM is the same as evaluating a standard binary SVM: $f(x;w) = \operatorname{sign}\left(\langle w_{\mathcal{X}}, \phi(x)\rangle + w_{\text{bias}} B\right)$, where $w = \begin{bmatrix}w_{\mathcal{X}}^\top, & w_{\text{bias}}\end{bmatrix}^\top$. The learning problem (7) is thus specialized to

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\xi, \quad \text{s.t. } \forall : t \in \mathcal{T}^N, \widehat{y} \in \mathcal{Y}^N$$
$$\xi \geq \frac{1}{N}\sum_i \frac{1 - y_i\widehat{y}_i}{2}(1 - y_i(\langle w_{\mathcal{X}}, \phi(t_i x_i)\rangle + w_{\text{bias}} B)).$$

Invariant rank SVM. The input to the rank optimizing SVM is a *sequence* $(x_1,\dots,x_N) \in \mathcal{X}^N$ of N data points and the output is a permutation that ranks positive points first [9]. Permutations are encoded as binary matrices $\widehat{y} \in \mathcal{Y}_N \subset \{-1,+1\}^{N\times N}$ where $\widehat{y}_{ij} = +1$ means that x_i is ranked before x_j. The loss function $\Delta(t, y, \widehat{y})$ is defined to be one minus the area under the ROC curve. [10] shows that optimizing this loss is the same as minimizing the number of incorrectly swapped pairs in the ranking. By adding transformation invariance, one gets the problem

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\xi, \quad \text{s.t. } \forall t \in \mathcal{T}^N, \widehat{y} \in \mathcal{Y}_N$$
$$\xi \geq \frac{1}{N^2}\sum_{ij:y_{ij}>0} \frac{1 - \widehat{y}_{ij}}{2}(1 - \langle w, \phi(t_i x_i) - \phi(t_j x_j)\rangle).$$

Note that the bias term is not needed since it is irrelevant for ranking. [10] shows that computing a maximally violated constraint for this problem can be done efficiently by sorting the samples by the score $\langle w, \phi(tx)\rangle$.

Pedestrian detection. We evaluate the invariant binary and rank SVMs on the DaimlerChrysler pedestrian classification benchmark [19]. The data consists of three training subsets with 800 positive 18×36 images (pedestrian) and 5000 negative ones (clutter) each and two analogous subsets for testing. While the DaimlerChrysler data includes virtual samples as well, these are discarded. Performance is measured in term of equal error rates (see [19] for details on the evaluation protocol).

As feature $\phi(x)$ we use the HOG-like descriptor of [18] that is suitable for small images. As transformations $\mathcal{T}$ we consider horizontal flipping and translation by one pixel in the eight directions, for a total of 18 transformations.

Motion as natural transformations. The DaimlerChrysler pedestrian instances are obtained from video tracks. While this technique yields cheaply a large quantity of training instances, it also results in a number of highly correlated data clusters, one for each tracked object. This breaks the fundamental i.i.d. assumptions on which most machine learning techniques rely. A way to solve this problem is to regard such a cluster as a single data point, and interpret its many members as *natural transformations* of the same object. As pedestrian movement is cyclic, this has an interesting interpretation as a local estimate of the manifold structure of pedestrian appearance. Our equivariant learning framework can then be used to incorporate invariance to this manifold. This has two advantages: (i) it reestablishes statistical independence of the samples and (ii) it significantly reduces the size of the training data, leaving the cutting plane algorithm to select a small number of representative points.

In order to explore this idea for the DaimlerChrysler dataset, we use agglomerative clustering to recover the sequences of tracked pedestrians (normally this information

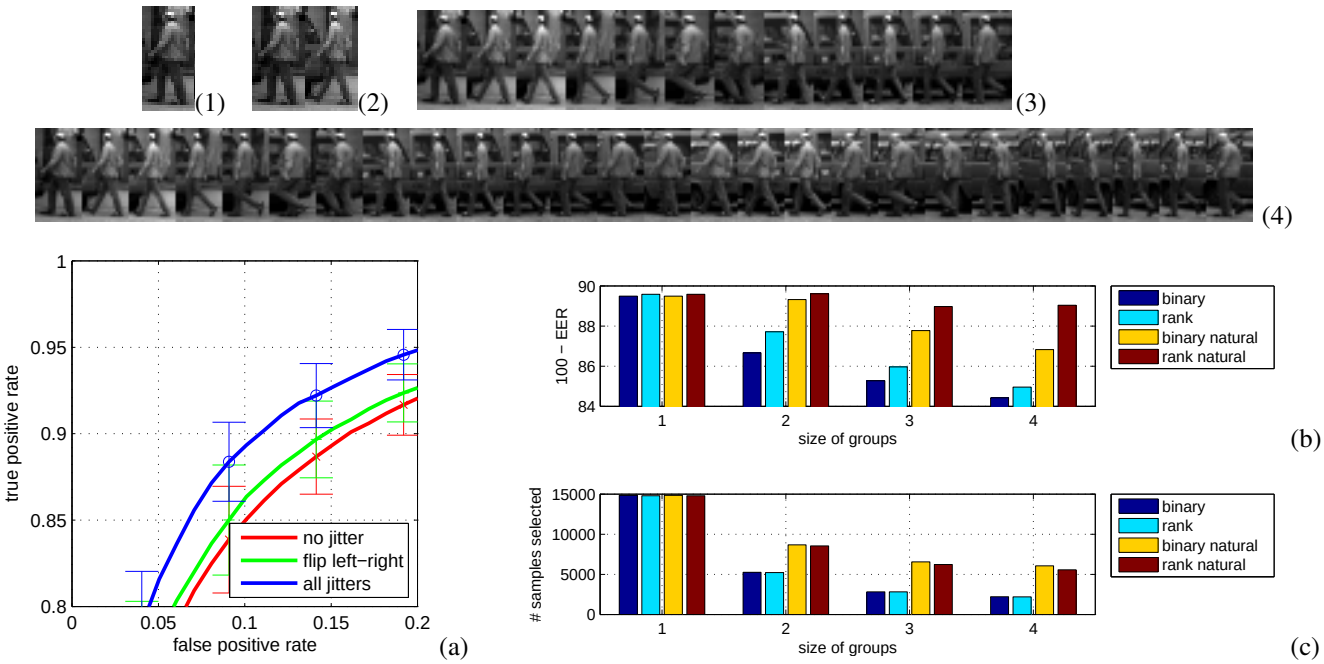

Figure 3: **Pedestrian dataset.** (1–4) examples of pedestrian groups (motion cycles) of increasing size. Groups define natural transformations. (a) ROC curves for the DaimlerChrysler benchmark data without transformation invariance, with invariance to flipping, and with invariance to flipping and translation (invariant binary SVM). (b) Classification accuracies versus number of data samples used by the cutting plane optimization for a standard SVM, the ROC-optimizing (rank) structured SVM, and the variants incorporating invariance. The horizontal axis is the degree of grouping of the training data (see Sect. 3.2). Larger groups cause only a fraction of the original data samples to be selected for training. However, only the structured rank SVM can deal properly with the resulting unbalanced training problem and maintain good performance.

would be provided as part of the dataset and this step would be unnecessary). We use four grouping thresholds, numbered from 1 to 4, resulting in larger and larger clusters.[1] Grouping 1 is the finest possible and corresponds to training with the unaltered data (Fig. 3.1–4). Each cluster is treated as a single data point modified by a set of transformations. Hence $t \in \mathcal{T}$ selects first an element in a group and then applies one of the 18 transformations as before.

Results. Fig. 3a-c shows the relative performance of the various SVMs on the DaimlerChrysler dataset. In Fig. 3a the equal error rate (EER) of the baseline binary SVM is 14% without any jitter, and improves to 13% when invariance to flipping is added, and to 11% when all eighteen transformations are considered (Fig. 3a). While here we are mostly interested in the relative improvement, we note that the performance reaches the state of the art (within statistical limits) for this setting [19].

Fig. 3b-c show the effect of natural transformations. The baseline invariant and the rank invariant SVMs use only one representative pedestrian per group. As the groupings get larger, the number of positive samples available for training decreases quickly by a factor of ten. The performance also drops (+6% EER), although not too dramatically, illustrating the redundancy of the instances within groups. Then, information is added back, this time in terms of invariance to natural transformations. This is shown in Figure 3 by the "binary natural" and "ranking natural" results. The number of samples used increases slightly, but is still far less than the overall number of samples, while performance increases significantly. In particular, the performance of the invariant rank SVM with natural transformations is virtually the same as using the entire dataset for training but only uses a fraction of the training samples. The advantage of the rank SVM is due in part to the fact that the data becomes more unbalanced with increasing group size (due to the reduced number of positive samples).

4. Conclusions

In this work we have introduced a novel formulation for the incorporation of equivariance and invariance in structured output SVM regressors. This is achieved by optimizing a convex upper bound to a regularized risk functional for structured outputs. The resulting optimization has the

[1] These clusters effectively recover sequential frames corresponding to individual pedestrians (Figure 3) and will be made available at the time of publication.

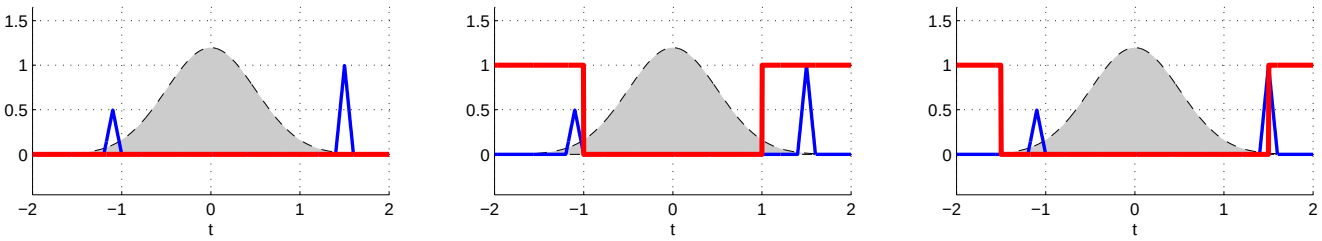

Figure 4: **Construction (11) of $\Delta(t)$.** In this example, the vicinal risk loss $\Delta_{\text{vr}}(t) \in [0, 1]$ has two triangular peaks at -1 and 3/2 (blue curve). The distribution $dP(t)$ is a Gaussian of variance $1/2$ (shaded area). The monotonic family U of bounding functions consists of the step functions of the type $\mu_{q,\delta}(t) = \chi_{\mathbb{R}-(-q,q)}(t)$ for all $q \geq 0$ and arbitrary $0 \leq \delta \leq 1$ (red curves). From left to right, the functions $\mu_{0,0}$, $\mu_{-1,1/2}$, and $\mu_{3/2,1}$ upper bound $\Delta_{\text{vr}}(0) = 0$, $\Delta_{\text{vr}}(-1) = 1/2$ and $\Delta_{\text{vr}}(3/2) = 1$ at $t = 0, -1, 2/3$ respectively. Among such curves, the one with largest expected value (11) is $\mu_{-1,1/2}$ (middle one), and this expected value upper bounds the expected value of $\Delta_{\text{vr}}(t)$.

structured output SVM as a special case by setting the set of equivariant transformations to the identity. By appropriately incorporating invariance into compatible definitions of joint feature functions and loss functions, we are able to efficiently optimize equivariant functions with available optimization software.

We have shown significant improvement over the baseline method on a challenging pedestrian dataset, and our proposed method is statistically tied with the state-of-the-art, which uses a different image representation [19]. Additionally, we have improved upon the state of the art for aerial car detection by incorporating rotation invariance into a discriminatively trained structured output detector. This indicates the framework's flexibility in learning invariance even with non-invariant kernels. Interestingly, use of the natural transformation manifold enabled a large reduction in the number of samples used without a corresponding decrease in performance.

In general, we propose that our algorithm be used in place of ad hoc sampling strategies or latent variable models to incorporate invariance and equivariance in computer vision. This has both practical and theoretical implications. From a computational perspective, the application of a cutting plane optimization strategy results in a large reduction in the number of generated samples to achieve the same performance. From the perspective of regularized risk, we have shown that the algorithm optimizes a convex upper bound of a natural extension of the regularized risk functional employed in classic structured output regression. This gives a principled foundation and an expectation of good generalization performance, as we have observed in our experiments.

Acknowledgements. This research is supported by the ERC grant VisRec no. 228180 and ONR MURI N00014-07-1-0182; M. Blaschko is funded by a Newton International Fellowship.

A. Probabilistic interpretation as vicinal risk

Vicinal risk minimization of Chapelle et al. [3] endows the transformation space $\mathcal{T}$ with a probability measure $dP(t)$ and minimizes the regularized *vicinal risk*:

$$\min_w \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N} \int \Delta_{\text{vr}}(ty_i, f(tx_i; w))\, dP(t). \quad (9)$$

Compared to our equivariant learning formulation (4), vicinal risk (9) considers the average rather than maximum error w.r.t transformations. We show next that it is possible to choose Δ in (4) so that the equivariant learning objective is an upper bound to (9).

For brevity of notation, define

$$\Delta_{\text{vr}}(t) = \Delta_{\text{vr}}(ty_i, f(tx_i; w)), \quad \Delta(t) = \Delta(t, y_i, f(tx_i; w)).$$

A sufficient condition for the equivariant risk (4) to upper bound the vicinal risk (9) is that

$$\int \Delta_{\text{vr}}(t)\, dP(t) \leq \sup_t \Delta(t). \quad (10)$$

An obvious choice that satisfies (10) is $\Delta(t) = \Delta_{\text{vr}}(t)$, as the average of a function cannot be larger than its supremum. However, such a bound is typically quite loose, as it does not account for any locality imposed by $dP(t)$. Much tighter bounds, particularly useful when $dP(t)$ is concentrated, can be obtained by considering an auxiliary family of bounding functions:

Lemma 1. *Let U be a family of functions $\mu : \mathcal{T} \to \mathbb{R}$ that (i) is monotonic (i.e. for any $\mu, \mu' \in U$ it is either $\mu \leq \mu'$ or $\mu' \leq \mu$), and (ii) for any transformation $t \in \mathcal{T}$ and scalar $\rho \in \mathbb{R}$ it has a member $\mu_{t,\rho}$ such that $\rho \leq \mu_{t,\rho}(t)$. Then the loss*

$$\Delta(t) = \int \mu_{t,\Delta_{vr}(t)}(q)\, dP(q) \quad (11)$$

satisfies the relation (10).

Proof. For simplicity, we give the proof in the case in which $\mathcal{T}$ is a finite set of transformations, so that $dP(t)$ is a discrete measure (see also Fig. 4). Let $P(t)$ denote the probability of t. By contradiction, suppose that (10) is false, i.e. that there exists a t^*

$$\sum_q \Delta_{\text{vr}}(q)P(q) > \Delta(t^*) = \sum_q \mu_{t^*,\Delta_{\text{vr}}(t^*)}(q)P(q).$$

Since the expected value of $\Delta'_i(t)$ is larger than the one of $\mu_{t^*,\Delta'_i(t^*)}$, there must be a point t_0 such that $P(t_0) > 0$ and $\Delta'_i(t_0) > \mu_{t^*,\Delta'_i(t^*)}(t_0)$. But then for property (ii) of U there is a function $\mu_{t_0,\Delta'_i(t_0)}(t_0) \geq \Delta'_i(t_0) > \mu_{t^*,\Delta'_i(t^*)}(t_0)$. Since $\mu_{t_0,\Delta'_i(t_0)}$ is strictly larger than $\mu_{t^*,\Delta'_i(t^*)}$ at t_0, for the monotonicity property (i) it must be $\mu_{t_0,\Delta'_i(t_0)} \geq \mu_{t^*,\Delta'_i(t^*)}$ everywhere. But then the expected value of $\mu_{t_0,\Delta'_i(t_0)}$ is at least as large as the one of $\mu_{t^*,\Delta'_i(t^*)}$, and in fact strictly larger since $\mu_{t_0,\Delta'_i(t_0)}(t_0) > \mu_{t^*,\Delta'_i(t^*)}(t_0)$ and $P(t_0) > 0$. But this contradicts the fact that the expected value of $\mu_{t^*,\Delta'_i(t^*)}$ is the largest for all t. □

References

[1] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *Proc. ECCV*, 2008. 2, 4

[2] M. B. Blaschko, A. Vedaldi, and A. Zisserman. Simultaneous object detection and ranking with weak supervision. In *NIPS*, 2010. 5

[3] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal risk minimization. In *Proc. NIPS*, 2001. 1, 2, 3, 7

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 4

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc20010/workshop/index.html, 2010. 4

[6] P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2009. 1, 3, 4

[7] B. Haasdonk and H. Burkhardt. Invariant kernel functions for pattern analysis and machine learning. *Mach. Learn.*, 68(1):35–61, 2007. 1

[8] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *Proc. ECCV*, 2008. 4, 5

[9] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 115–132, Cambridge, MA, 2000. MIT Press. 5

[10] T. Joachims. A support vector method for multivariate peformance measures. In *Proc. ICML*, 2005. 4, 5

[11] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1), 2009. 3, 4, 5

[12] M. P. Kumar, P. H. S. Torr, and A. Zisserman. An invariant large margin nearest neighbour classifier. In *Proc. ICCV*, 2007. 1

[13] I. Laptev. Improvements of object detection using boosted histograms. In *Proc. BMVC*, 2006. 1

[14] F. Lauer and G. Bloch. Incorporating prior knowledge in support vector machine classification: A review. *Neurocomputing*, 71, 2008. 1

[15] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *PAMI*, 2006. 1

[16] F. Li, C. Ionescu, and C. Sminchisescu. Random fourier approximations for skewed multiplicative histogram kernels. In *Proc. DAGM*, 2010. 4

[17] G. Loosli, S. Canu, S. V. N. Vishwanathan, and A. J. Smola. Invariance in calssification: an efficient SVM implementation. In *Proc. ASMDA*, 2005. 2

[18] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Proc. CVPR*, 2008. 5

[19] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *PAMI*, 28(11), 2006. 5, 6, 7

[20] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. ECCV*, 2010. 5

[21] T. Poggio and T. Vetter. Recognition and structure from one 2d model view: observations on prototypes, object classes and symmetries. Technical Report AIM-1347, MIT, 1992. 1

[22] A. Pozdnoukhov and S. Bengio. Invariances in kernel methods: From samples to objects. *Pattern Recognition Letters*, 27(10), 2006. 1

[23] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proc. NIPS*, 2007. 4

[24] M. Reisert and H. Burkhardt. Learning equivariant functions with matrix valued kernels. *Journal of Machine Learning Research*, 8, 2007. 1

[25] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002. 1

[26] P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition - tangent distance and tangent propagation. *International Journal of Imaging Systems and Technology*, 2001. 1

[27] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Proc. NIPS*, 2003. 1, 4

[28] C.-H. Teo, A. Globerson, S. Roweis, and A. J. Smola. Convex learning with invariances. In *Proc. NIPS*, 2007. 1, 2

[29] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proc. ICML*, 2004. 1, 4

[30] C. Walder and O. Chapelle. Learning with transformation invariant kernels. In *NIPS*. 2008. 1

[31] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Proc. NIPS*, 2001. 4

[32] C.-N. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In *Proc. ICML*, 2009. 1, 3

[33] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV*, pages 141–154, 2010. 2, 5