Learning Equivariant Structured Output SVM Regressors

Andrea Vedaldi Matthew Blaschko Andrew Zisserman — Oxford University

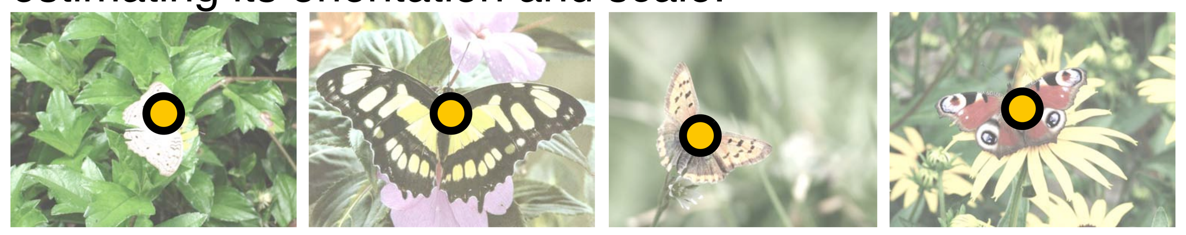
Overview

In many tasks the goal is to *learn a function that varies predictably with transformations of the input.* Examples:

- **Pose-invariant classification.** Recognize an object category *regardless* of the object translation, rotation, and scale.

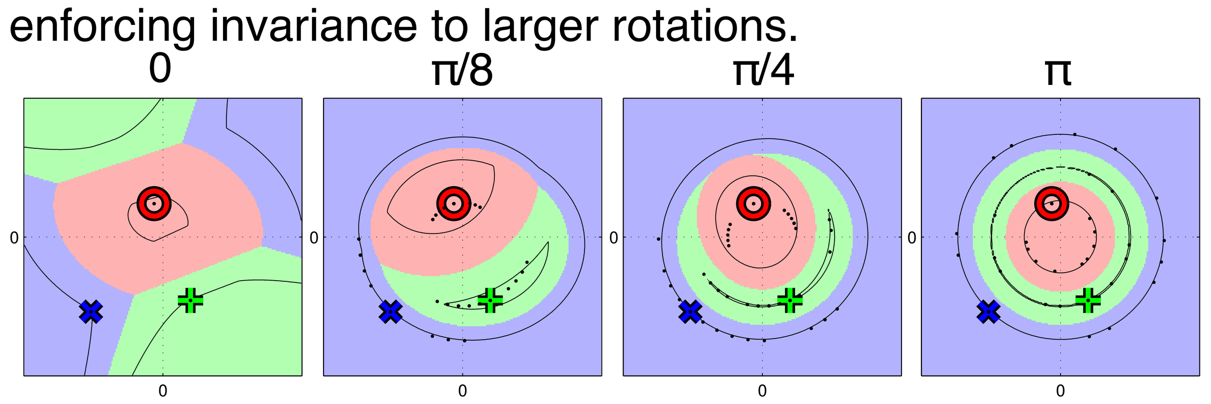
- **Pose regression.** Detect an object and estimate its translation, rotation, and scale.

- **Detection.** Find an object location (center), *without* estimating its orientation and scale.



Toy Example: Rotation-invariant classification

Learn three classes of 2D points (red **o**, green **+**, blue **x**). starting from just one example point per class and gradually enforcing invariance to larger rotations.



Learn a function parametrized by w

$$f(\cdot;w):\mathbb{R}^2 \to \{\circ,+,\times\}$$

trading-off its prediction accuracy and smoothness:

$$\frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}[y_i \neq f(x_i;w)]$$

If it is known that the class of a point is invariant to rotations of $[-\theta_0,\theta_0]$ radians, this can be enforced by taking the maximum classification error of the transformed data:

$$\frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\sup_{\theta\in[-\theta_0,+\theta_0]}[y_i \neq f(R(\theta)x_i;w)]$$

Contributions

1. A method to enforce **equivariance** of the function (and invariance as a special case).
2. Instantiation as a structured output SVM to efficiently handle arbitrary output spaces.
3. An application of constraint generation that can be interpreted as selecting useful virtual samples.

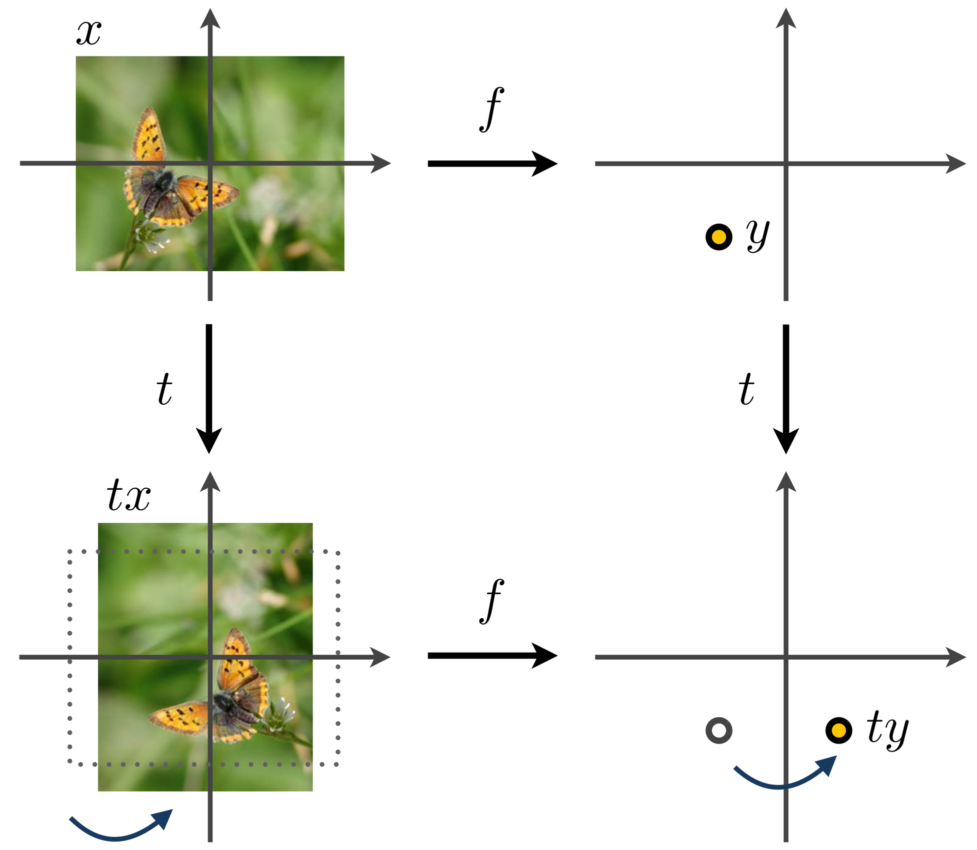
Formulation

A function $f:\mathcal{X}\to\mathcal{Y}$ is **equivariant** if its output varies with its input in a **predictable** way for given **transformations** $\mathcal{T}$:

$$\forall t \in \mathcal{T}: \quad f(tx;w) \approx tf(x;w)$$

The effect of t on the input and output spaces $\mathcal{X}$ and $\mathcal{Y}$ can be chosen arbitrarily. **Invariance** is obtained when t acts as the identity on the output ($ty = y$).

Example (co-variance). Let $y = f(x;w)$ be the location of a butterfly in an image x. If tx is the rotated image, then the butterfly location $ty = f(tx;w)$ should track the motion.



Encoding equivariance in the loss

For each example (x_i, y_i), maximize the loss with respect to all possible equivariant variations (tx_i, ty_i) of input and output:

$$\frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\sup_{t\in\mathcal{T}}\Delta(t,y_i,f(tx_i;w))$$

Note: the loss depends on the pair (t, y_i) rather than the application ty_i to enable weighting the transformations.

Example: weighted equivariant 01-loss

$$\Delta(t,y_i,f(tx_i;w)) = W(t)[ty_i \neq f(tx_i;w)].$$

Convex formulation: Equivariant structured SVM

Specialize the formulation as a structured output SVM to obtain a convex learning problem by:

1. Parametrizing $f(x;w)$ as
$$f(x;w) = \operatorname*{argmax}_{y\in\mathcal{Y}}\langle w,\Psi(x,y)\rangle$$
2. Making the loss convex by margin (or slack) rescaling:
$$\sup_{t\in\mathcal{T},y\in\mathcal{Y}}\Delta(t,y_i,\hat{y}) + \langle w,\Psi(x_i,\hat{y}) - \Psi(x_i,ty_i)\rangle$$

Algorithm

The learning problem is a quadratic program with a (potentially) ∞ number of constraints:

$$\min_{w}\frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i$$

$$\forall i,t,\hat{y}: \ \xi_i \geq \Delta(t,y_i,\hat{y}) + \langle w,\Psi(x_i,\hat{y}) - \Psi(x_i,ty_i)\rangle$$

Constraint generation and virtual sampling

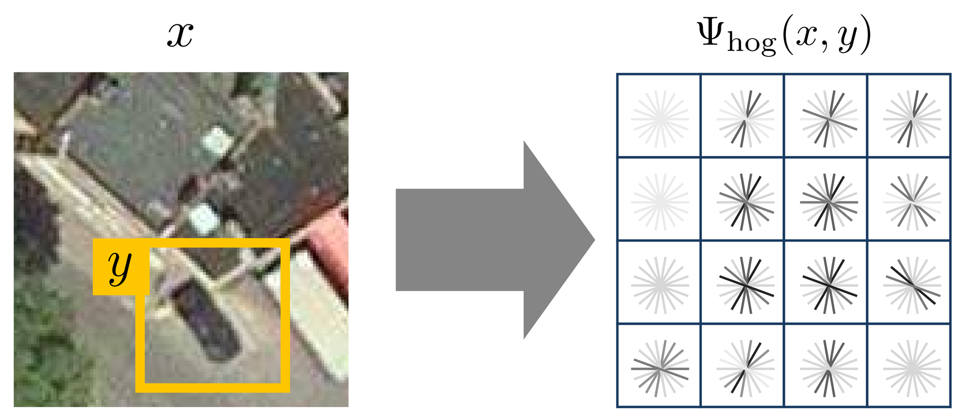
Constraint generation automatically and iteratively identifies a small subset of constraints active at the global optimum.

Since spanning transformations is the same as generating a large set of **virtual samples**, constraint generation selects relevant virtual samples.

Example: Co-variant object detection

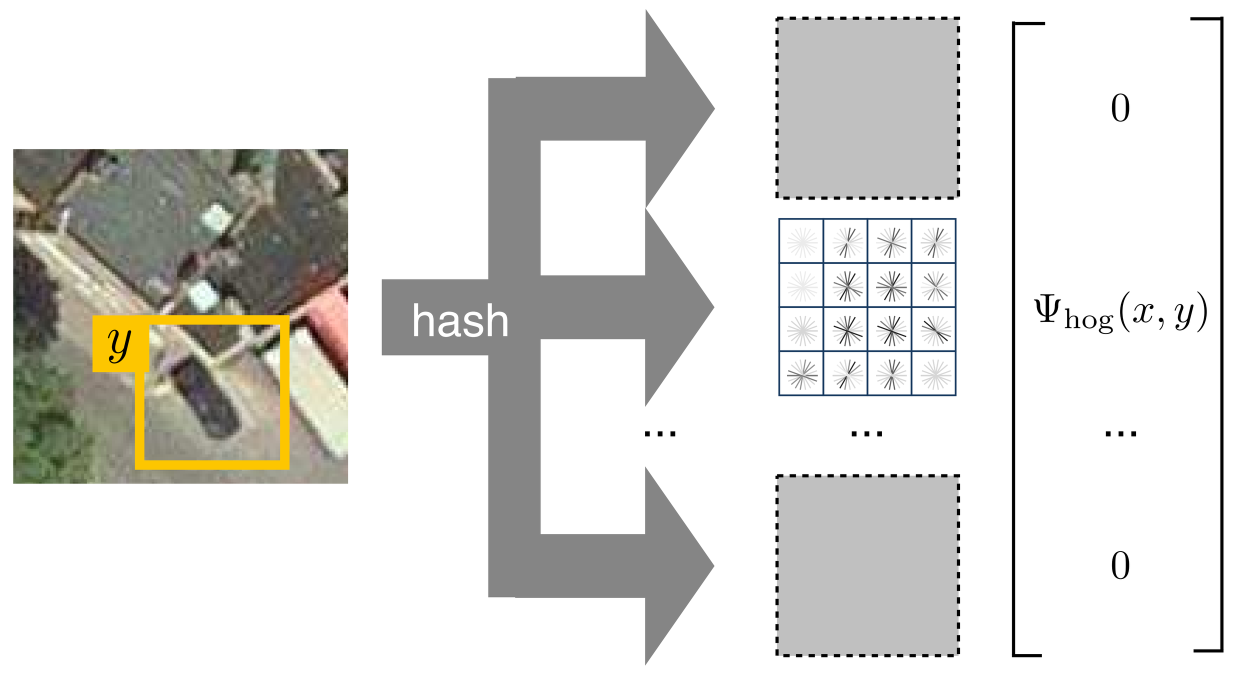
- **Input:** aerial image.
- **Output:** location of a car.
- **Loss:** 01-loss at 50% PASCAL VOC overlap.
- **Equivariance:** the detector must output the roto-translation of the image plane, but it **does not need to estimate the car rotation** (faster inference).

Linear kernel on HOG features



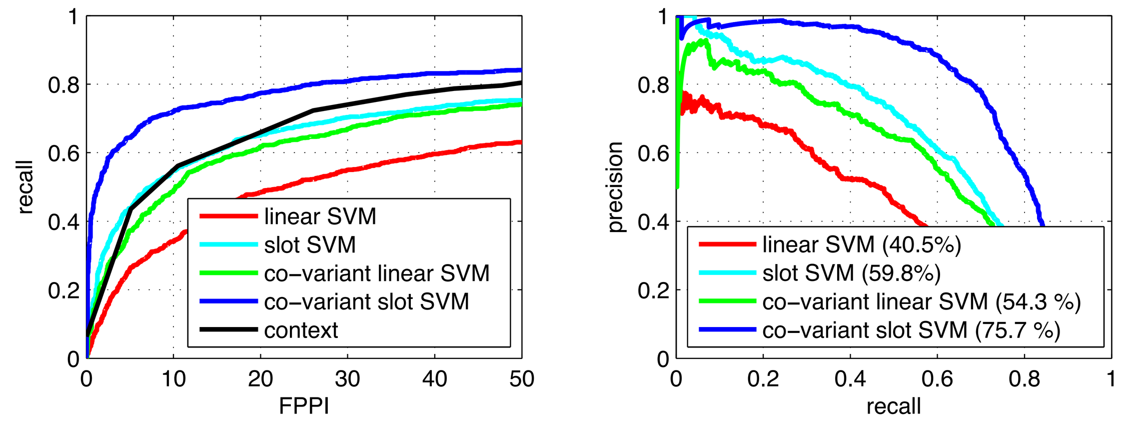
Fast non-linear kernel: slots

Linear HOG kernels blur rotating cars. We use instead **slot kernels,** a mixture of linear kernels indexed by a fast hashing function of the patch itself.



This is the same as a hashed mixture of linear SVMs.

Results

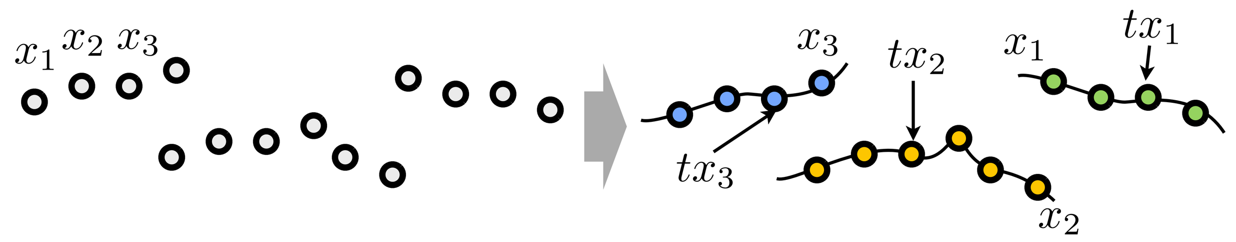


Example: Invariant learning to rank

- **Input:** images of pedestrians and clutter.
- **Output:** ranking of images, pedestrians first.
- **Loss:** 1 - ROC area.
- **Invariance:** ranking is invariant to small jitters and/or articulation of the walking pedestrian.

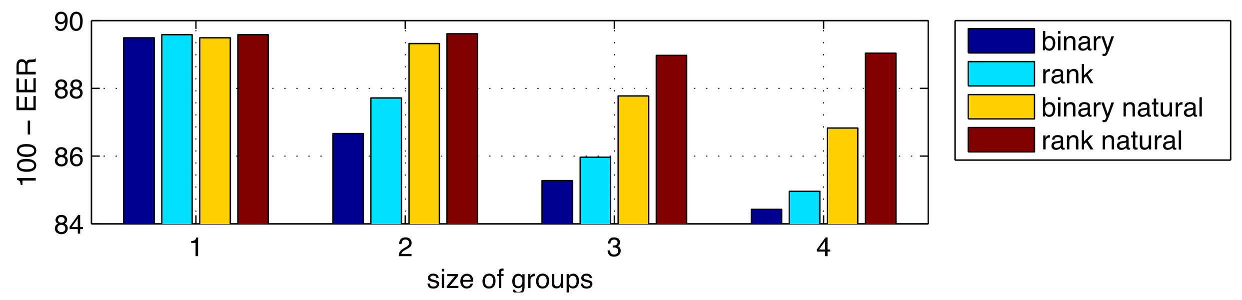
Natural transformations

Samples are not i.i.d. if pedestrians are tracked through walking cycles. **Idea:** treat each cycle as a single example, from which the transformation t selects a frame.

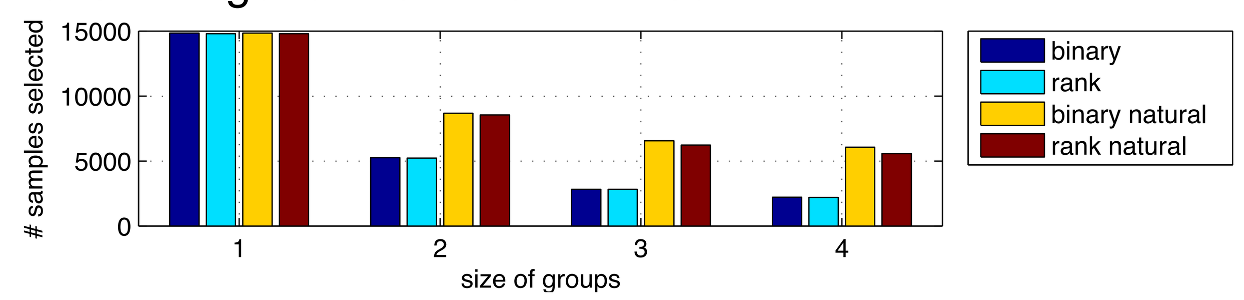


Results

Equal-error rate (EER) as grouping becomes more aggressive for standard and ranking SVMs and the invariant versions.



Number of samples, including virtual samples, selected by constraint generation in each case.



Conclusion: invariant learning to rank achieves the same performance by activating only a fraction of virtual samples.

References

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

O. Chapelle, J. Weston, L. Bottou, and V. Vapnik. Vicinal