A Brief Overview of

Structured Support Vector Machines (S-SVM)

for Computer Vision

by Magnus Burenius at KTH

Disadvantages of standard binary SVM

- For object recognition it does not handle multiple classes naturally.
- It does not handle anything but zero-one loss.

Disadvantages of standard binary SVM

- For object detection the gathering of negative data is problematic.
- We cannot use an infinite set of negative data.
- What samples should be chosen?
- How do we define positive and negative for partially overlapping bounding boxes?
- Jittering of samples often needed to get good performance.
- These issues are handled by some ad-hoc heuristic.

Disadvantages of standard binary SVM

- For pose estimation it has the same disadvantages as for detection.
- Furthermore, if the object is decomposed into many parts there is a problem of aggregating the score from the parts.

Wikipedia on structured learning

Structured learning is the subfield of machine learning concerned with computer programs that learn to map inputs to arbitrarily complex outputs.

This stands in contrast to the simpler approaches of classification, where input data (instances) are mapped to "atomic" labels, i.e. symbols without any internal structure, and regression, where inputs are mapped to scalar numbers or vectors.

S-SVM Feature Function

- Let X and Y be the input and output and let Ω_X and Ω_Y be their sample spaces. These can be ANY spaces, not just integers or real vector spaces.
- A feature function Ψ is used to map a pair from these complicated spaces to something we can compute with:

$$\Psi : \Omega_X \times \Omega_Y \rightarrow R^d$$

S-SVM Prediction

- A classifier described by a vector ω predicts a class by solving

$$\underset{y}{argmax} \quad \omega \cdot \Psi(x, y)$$

- This imposes a restriction on Ψ

S-SVM Loss Function

During training the STRUCTURE of the output space is taking into account by defining a loss function

$$\Delta : \Omega_Y \times \Omega_Y \rightarrow R$$

which quantifies the loss of predicting y_p when the true output is y. It should fulfill

$$\Delta(y, y_p) \geqslant 0$$
$$\Delta(y, y_p) = 0 \quad iff \quad y = y_p$$

Δ should thus reflect the quantity which measures how well the classifier performs.

S-SVM Training

- Given a training set $(x_1,y_1)\ldots(x_N,y_N)$ of **only positives** and a regularization constant C a classifier ω is trained by solving the convex optimization problem

$$\min_{\omega} \;\; \|\omega\|^2 + C\sum_{n} \max_{y} \left(\Delta\left(y_n, y\right) + \omega\cdot\left(\Psi\left(x_n, y\right) - \Psi\left(x_n, y_n\right)\right)\right)$$

Software Library

- SVM^{struct} by Thorsten Joachims can be used to train a Structured SVM.

- Need to define the problem dependent:

  * Input and output spaces Ω_X and Ω_Y.

  * Feature function $\Psi(x,y)$

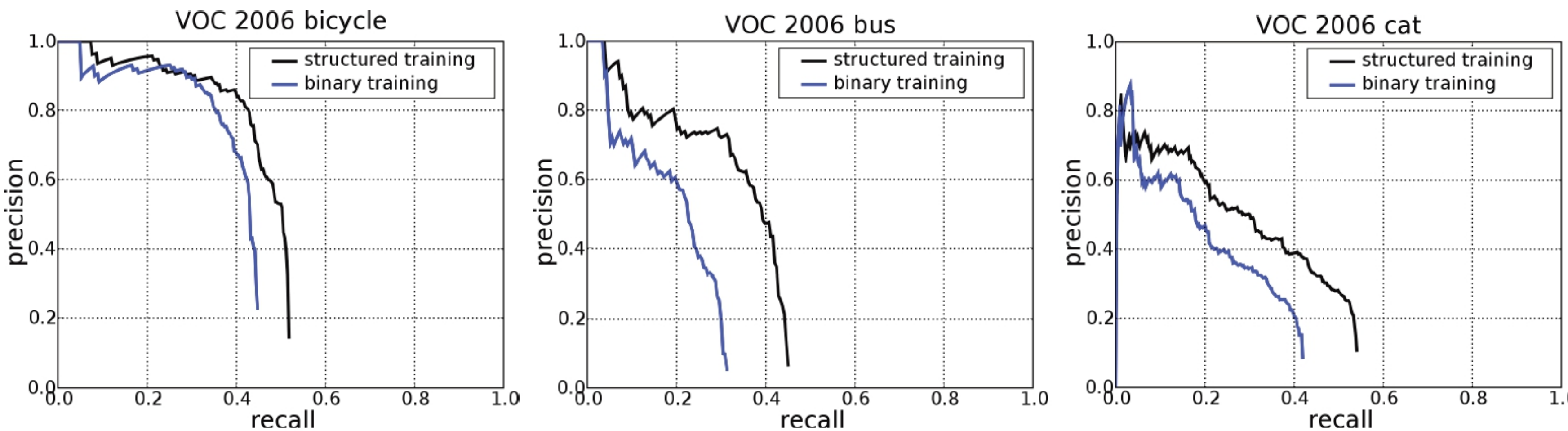
  * Loss function $\Delta(y_1,y_2)$

  * Function to perform maximization $\max_y \; \omega \cdot \Psi(x, y)$

- The library handles everything else, including the dynamic constraint generation, i.e. mining for relevant negatives.

Learning to Localize Objects with Structured Output Regression

Blaschko and Lampert ECCV 2008 (best student paper)



Precision-recall curves and example detections for the PASCAL VOC bicycle, bus and cat category (from left to right). Structured training improves both, precision and recall.

Learning to Localize Objects with Structured Output Regression

Blaschko and Lampert ECCV 2008 (best student paper)

	bike	bus	car	cat	cow	dog	horse	m.bike	person	sheep
structured training	.472	**.342**	.336	**.300**	**.275**	.150	**.211**	**.397**	.107	.204
binary training	.403	.224	.256	.228	.114	**.173**	.137	.308	.104	.099
best in competition	.440	.169	.444	.160	.252	.118	.140	.390	.164	**.251**
post competition	**.498**†	.249‡	**.458**†	.223*	—	.148*	—	—	**.340**$^{+}$	—

Table 2. Average Precision (AP) scores on the 10 categories of PASCAL VOC 2006. Structured training consistently improves over binary training, achieving 5 new best scores. In one category binary training achieves better results than structured training, but both methods improve the state-of-the-art. Results best in competition were reported in [17]. Results post competition were published after the official competition: †[25], ‡[2], *[7], $^{+}$[10].

Their Conclusion

- Structured learning can use more training data since it has access to **all** possible bounding boxes of the training images instead of a sparse sample.
- The search for relevant bounding boxes are integrated with the training.
- When used in a sliding window approach it handles partial detections better than a binary SVM since it trains on such data as well.

Ranking SVM

Special case of structured learning problem. Similar to S-SVM we have arbitrary input Ω_X and output Ω_X spaces and a feature function:

$$\Psi : \Omega_X \times \Omega_Y \rightarrow R^d$$

However, we do not want to predict a single output y* given the input x. Instead we want to rank a discrete set of given outputs

$$\{y_1, \ldots, y_N\} \subseteq \Omega_Y$$

based on the classifier response

$$y_i \geqslant y_j \quad iff \quad \omega \cdot \Psi(x, y_i) \geqslant \omega \cdot \Psi(x, y_j)$$

Ranking SVM

- Unlike S-SVM we do **not** know how to quantify the **structure** of the output space with a loss function Δ.

Ranking SVM

- Instead we assume the **structure** is described by training examples of input and ordered outputs

Ranking SVM

- A weak ordering over the output space Ω_Y can be defined by a discrete set

$D \subseteq \Omega_Y \times \Omega_Y$ such that $y_i \geqslant y_j \;\; if \;\; (y_i, y_j) \in D$

- Ranking SVM learns a classifier ω which tries to respect the ordering of the training data $(x_1,D_1),\ldots,(x_N,D_N)$

$$\min_{\omega} \;\; \|\omega\|^2 + C \sum_{i,j,n} \xi_{i,j,n}$$

$$s.t. \quad \xi_{i,j,n} \geqslant 0$$

$$\omega \cdot (\Psi(x_n, y_i) - \Psi(x_n, y_j)) \geqslant 1 - \xi_{i,j,n} \quad \forall k \, \forall (y_i, y_j) \in D_n$$

Applications

When does it make sense to use Ranking SVM instead of Structured SVM?

Next step

Learning Equivariant Structured Output SVM Regressors

by Vedaldi, Blaschko, Zisserman from ICCV 2011

They discuss how to introduce latent structures while still having convex training, unlike Latent SVM