Latent and Structured SVMs

Laurens van der Maaten

Latent SVM

- How can we train an object detector with a pictorial structures model?

Latent SVM

- How can we train an object detector with a pictorial structures model?

- Let's first consider the standard linear SVM:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \max(0, 1 - y_n \mathbf{w}^\top \mathbf{x}_n)$$

Latent SVM

- How can we train an object detector with a pictorial structures model?

- Let's first consider the standard linear SVM:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \max(0, 1 - y_n \mathbf{w}^\top \mathbf{x}_n)$$

classifier score

Latent SVM

- How can we train an object detector with a pictorial structures model?

- Let's first consider the standard linear SVM:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\max(0, 1 - y_n\mathbf{w}^\top\mathbf{x}_n)$$

classifier score

hinge loss

Latent SVM

- How can we train an object detector with a pictorial structures model?

- Let's first consider the standard linear SVM:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\max(0, 1 - y_n\mathbf{w}^\top\mathbf{x}_n)$$

regularizer (max. margin)

classifier score

hinge loss

Latent SVM

- How can we train an object detector with a pictorial structures model?

- Let's first consider the standard linear SVM:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \max(0, 1 - y_n \mathbf{w}^\top \mathbf{x}_n)$$

regularized loss

regularizer (max. margin)

classifier score

hinge loss

Latent SVM

- How can we train an object detector with a pictorial structures model?

- Let's first consider the standard linear SVM:

$$\underbrace{L(\mathbf{w})}_{\text{regularized loss}} = \underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{\text{regularizer (max. margin)}} + C \sum_{n=1}^{N} \underbrace{\max(0, 1 - y_n \underbrace{\mathbf{w}^\top \mathbf{x}_n}_{\text{classifier score}})}_{\text{hinge loss}}$$

- Latent SVM introduces *latent variables* modeling part locations:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{n=1}^{N} \max\left(0, 1 - \max_{x_1, y_1, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}_n; x_1, y_1, \ldots, x_{|V|}, y_{|V|})\right)$$

* Felzenszwalb *et al.*, 2010

Latent SVM

- To compute the loss, we need to find the best part locations:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\max\left(0, 1 - y_n \max_{x_1,y_1,\ldots,x_{|V|},y_{|V|}} s(\mathbf{I}_n; x_1, y_1, \ldots, x_{|V|}, y_{|V|})\right)$$

* Felzenszwalb *et al.*, 2010

Latent SVM

- To compute the loss, we need to find the best part locations:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \max\left(0, 1 - y_n \max_{x_1, y_1, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}_n; x_1, y_1, \ldots, x_{|V|}, y_{|V|})\right)$$

score of a part configuration

* Felzenszwalb *et al.*, 2010

Latent SVM

- To compute the loss, we need to find the best part locations:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \max\left(0, 1 - y_n \max_{x_1,y_1,\ldots,x_{|V|},y_{|V|}} s(\mathbf{I}_n; x_1, y_1, \ldots, x_{|V|}, y_{|V|})\right)$$

score of optimal part configuration

score of a part configuration

* Felzenszwalb *et al.*, 2010

Latent SVM

- To compute the loss, we need to find the best part locations:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\max\left(0, 1 - y_n \max_{x_1,y_1,\ldots,x_{|V|},y_{|V|}} s(\mathbf{I}_n; x_1, y_1, \ldots, x_{|V|}, y_{|V|})\right)$$

parameters of global and part filters

score of optimal part configuration

score of a part configuration

* Felzenszwalb *et al.*, 2010

Latent SVM

- To compute the loss, we need to find the best part locations:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \max\left(0, 1 - y_n \max_{x_1, y_1, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}_n; x_1, y_1, \ldots, x_{|V|}, y_{|V|})\right)$$

parameters of global and part filters

score of optimal part configuration

score of a part configuration

- Recall that the score of a pictorial-structures model is given by:

$$s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|}) = \mathbf{w}_0^{\mathrm{T}}\phi(\mathbf{I}; x_0, y_0) + \sum_{i \in V} \mathbf{w}_i^{\mathrm{T}}\phi(\mathbf{I}; x_i, y_i) + \sum_{(i,j) \in E} d_{ij}\phi_d(x_i - x_j, y_i - y_j)$$

* Felzenszwalb *et al.*, 2010

Latent SVM

- To compute the loss, we need to find the best part locations:

$$L(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \max\left(0, 1 - y_n \max_{x_1,y_1,\ldots,x_{|V|},y_{|V|}} s(\mathbf{I}_n; x_1, y_1, \ldots, x_{|V|}, y_{|V|})\right)$$

parameters of global and part filters

score of optimal part configuration

score of a part configuration

- Recall that the score of a pictorial-structures model is given by:

$$s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|}) = \mathbf{w}_0^{\mathrm{T}}\phi(\mathbf{I}; x_0, y_0) + \sum_{i\in V}\mathbf{w}_i^{\mathrm{T}}\phi(\mathbf{I}; x_i, y_i) + \sum_{(i,j)\in E} d_{ij}\phi_d(x_i - x_j, y_i - y_j)$$

- We can now “simply” compute the gradient of the loss w.r.t. parameters

* Felzenszwalb *et al.*, 2010

Latent SVM

- The gradient of the latent SVM objective takes the form:

$$\frac{\partial L}{\partial \mathbf{w}_i} = \mathbf{w}_i + C \sum_{n=1}^{N} \max\left(0, -y_n \phi(\mathbf{I}_n; x_i^*, y_i^*)\right)$$

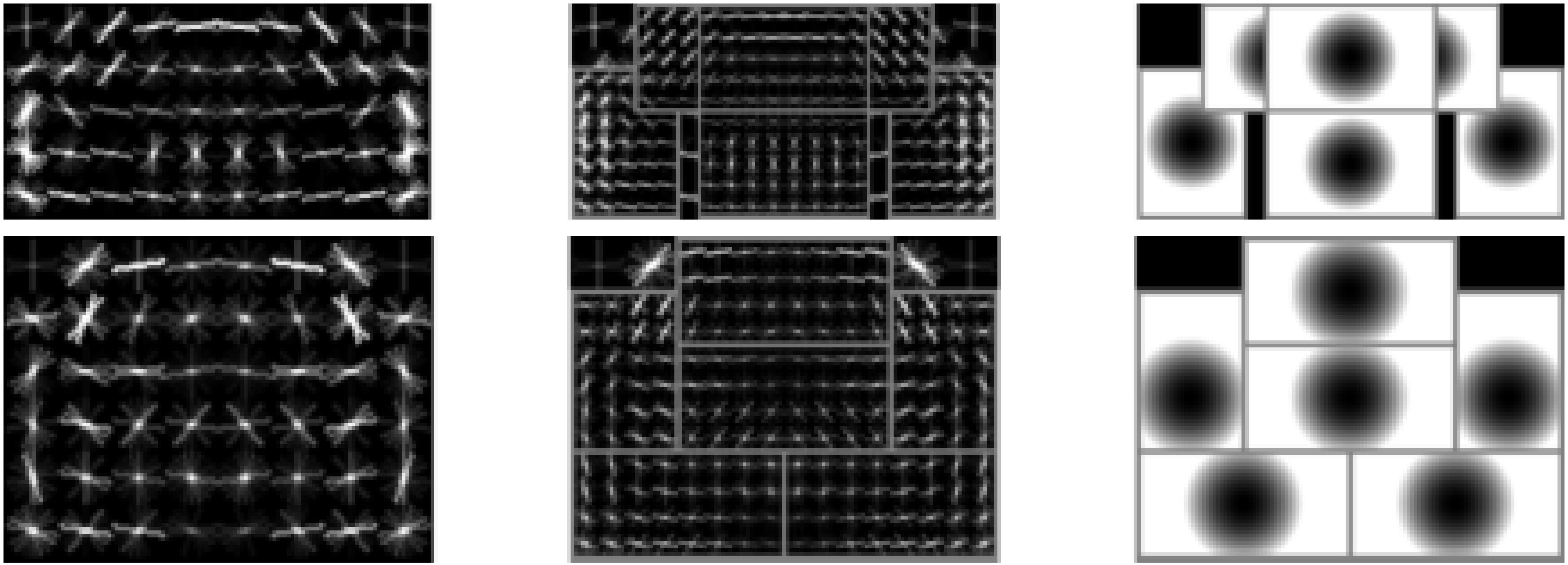
- Where we have defined the optimal part locations:

$$(x_1^*, y_1^*, \ldots, x_{|V|}^*, y_{|V|}^*) = \underset{x_1, y_1, \ldots, x_{|V|}, y_{|V|}}{\text{argmax}} \; s(\mathbf{I}_n; x_1, y_1, \ldots, x_{|V|}, y_{|V|})$$

* Felzenszwalb *et al.*, 2010

Learned model

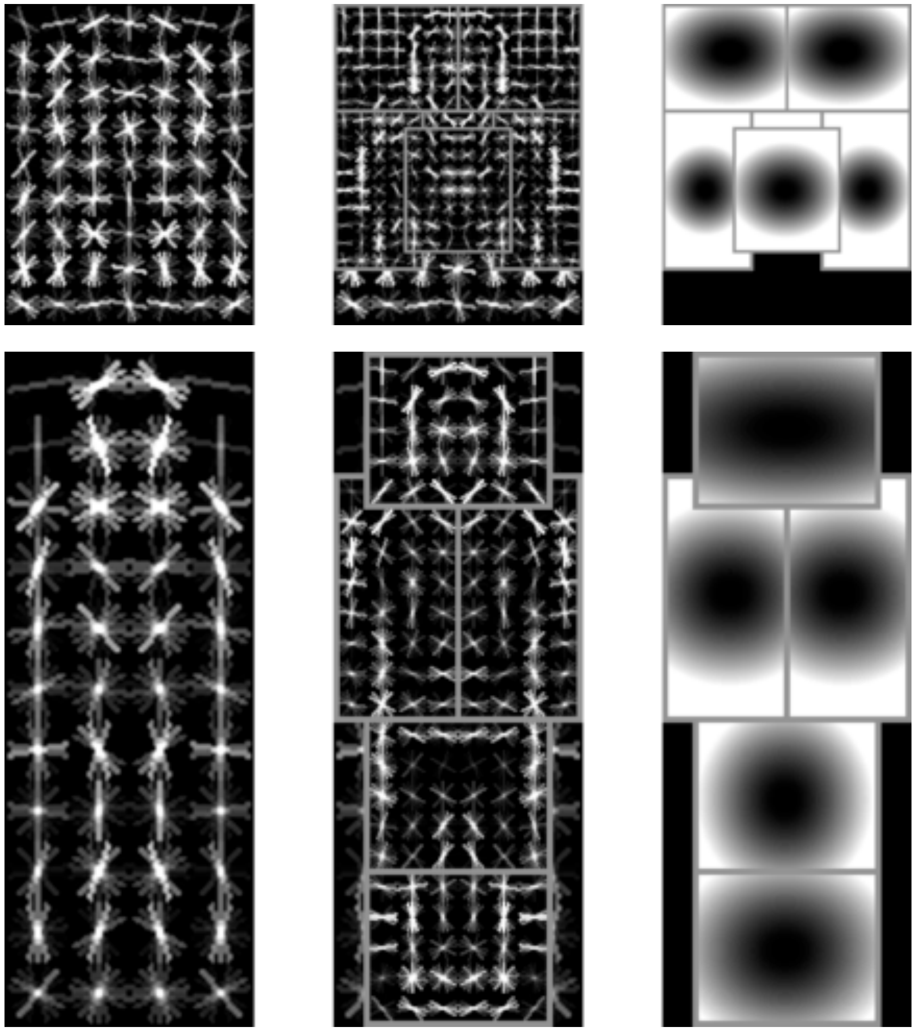
- Illustration of a learned car detector:



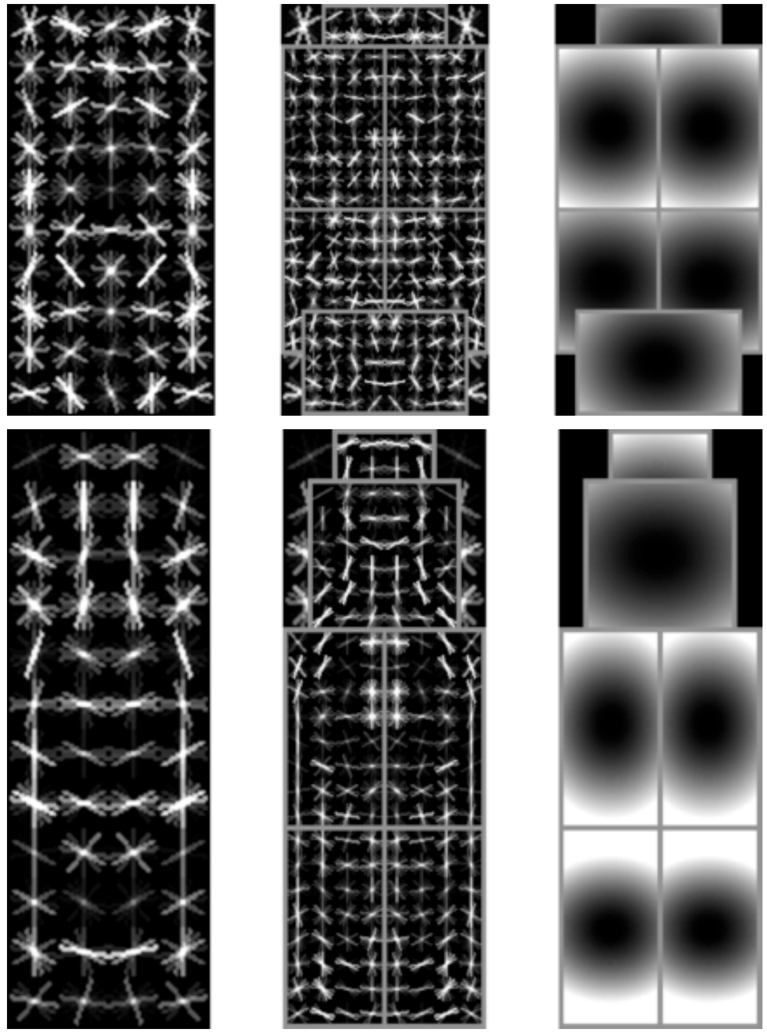
* Felzenszwalb *et al.*, 2010

Learned model

person



bottle



* Felzenszwalb *et al.*, 2010

Structured SVMs

Structured SVM

- A binary SVM makes a prediction by finding the highest-scoring label:

$$f(\mathbf{x}|\Theta) = \operatorname*{argmax}_{y \in \{-1,+1\}} s(y; \mathbf{x}, \Theta) = \operatorname*{argmax}_{y \in \{-1,+1\}} y\mathbf{\Theta}^\top \mathbf{x}$$

Structured SVM

- A binary SVM makes a prediction by finding the highest-scoring label:

$$f(\mathbf{x}|\Theta) = \underset{y\in\{-1,+1\}}{\operatorname{argmax}}\; s(y;\mathbf{x},\Theta) = \underset{y\in\{-1,+1\}}{\operatorname{argmax}}\; y\boldsymbol{\Theta}^\top\mathbf{x}$$

- *Structured SVMs* are generalization that searches for highest-scoring output:

$$f(\mathbf{x}|\boldsymbol{\Theta}) = \underset{y\in\mathcal{Y}}{\operatorname{argmax}}\, s(y;\mathbf{x},\boldsymbol{\Theta})$$

Structured SVM

- A binary SVM makes a prediction by finding the highest-scoring label:

$$f(\mathbf{x}|\Theta) = \operatorname*{argmax}_{y \in \{-1,+1\}} s(y; \mathbf{x}, \Theta) = \operatorname*{argmax}_{y \in \{-1,+1\}} y\Theta^\top \mathbf{x}$$

- *Structured SVMs* are generalization that searches for highest-scoring output:

$$f(\mathbf{x}|\Theta) = \operatorname*{argmax}_{y \in \mathcal{Y}} s(y; \mathbf{x}, \Theta)$$

set of all structures: label sequences, graphs, image segmentations, object locations, *etc.*

Structured SVM

- In detection, we aim to learn a function from image to bounding box + label

- Input $\boldsymbol{x}$ = *image*

- Output y = *(label, bounding box)*

Structured SVM

- In detection, we aim to learn a function from image to bounding box + label

- Input ***x*** = *image*

- Output *y* = *(label, bounding box)*

- Assume we have a score function for a structure y:

$$s(y; \mathbf{x}, \Theta)$$

  - For instance, the score a pictorial-structures model assigns to the bounding box

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

* Tsochantaridis *et al.*, 2005

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

score of ground-truth

* Tsochantaridis *et al.*, 2005

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

score of alternative

score of ground-truth

* Tsochantaridis *et al.*, 2005

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

score of alternative

score of ground-truth

margin / task loss

* Tsochantaridis *et al.*, 2005

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

highest-scoring alternative

score of alternative

score of ground-truth

margin / task loss

* Tsochantaridis *et al.*, 2005

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

highest-scoring alternative | **score of alternative** | **score of ground-truth** | **margin / task loss**

- Learning amounts to minimizing the structured SVM loss w.r.t. parameters:

$$\Theta^* = \operatorname*{argmin}_{\Theta} \ell(\Theta; \mathbf{x}, y)$$

* Tsochantaridis *et al.*, 2005

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

highest-scoring alternative **score of alternative** **score of ground-truth** **margin / task loss**

- Learning amounts to minimizing the structured SVM loss w.r.t. parameters:

$$\Theta^* = \operatorname*{argmin}_{\Theta} \ell(\Theta; \mathbf{x}, y)$$

- Lower score of highest-scoring alternative relative to the ground-truth score

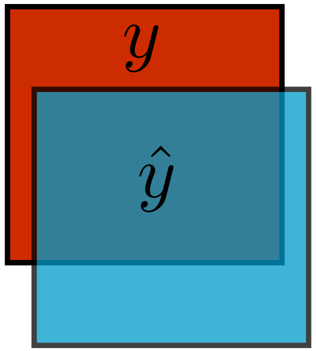
* Tsochantaridis *et al.*, 2005

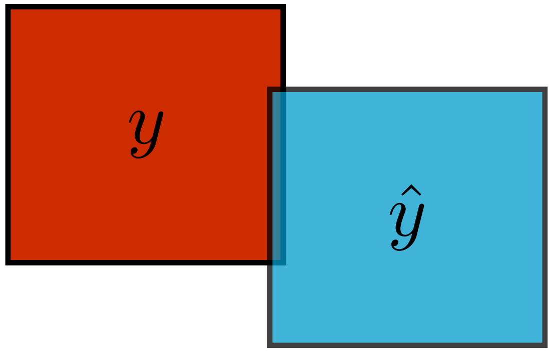
Structured SVM

- The task loss can take different forms depending on the application

Structured SVM

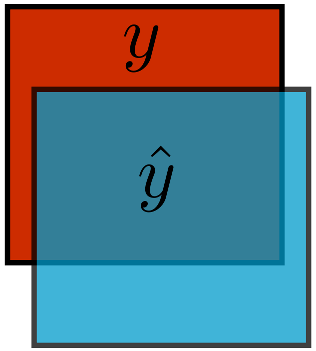
- The task loss can take different forms depending on the application
- For instance, when training an object detector: $\Delta(y, \hat{y}) = 1 - \frac{y \cap \hat{y}}{y \cup \hat{y}}$

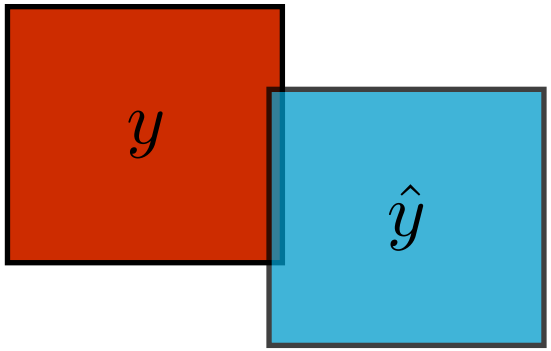
small margin: 

large margin: 

Structured SVM

- The task loss can take different forms depending on the application
- For instance, when training an object detector: $\Delta(y, \hat{y}) = 1 - \frac{y \cap \hat{y}}{y \cup \hat{y}}$

small margin: 

large margin: 

- Much overlap with target: slightly lower score than ground truth
- No overlap with target: much lower score than ground truth

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

- The standard binary-classification SVM is a special case where:

$$y \in \{-1, +1\}$$
$$s(y; \mathbf{x}, \Theta) = y\Theta^\top \mathbf{x}$$

$$\Delta(y, \hat{y}) = \begin{cases} 0 \text{ iff } y = \hat{y} \\ 1 \text{ iff } y \neq \hat{y} \end{cases}$$

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

- The standard binary-classification SVM is a special case where:

$$y \in \{-1, +1\}$$
$$s(y; \mathbf{x}, \Theta) = y\Theta^\top \mathbf{x}$$

$$\Delta(y, \hat{y}) = \begin{cases} 0 \text{ iff } y = \hat{y} \\ 1 \text{ iff } y \neq \hat{y} \end{cases}$$

  - Working out the loss leads to:

$$\max \left[y\Theta^\top \mathbf{x} - y\Theta^\top \mathbf{x} + 0, (1 - y)\Theta^\top \mathbf{x} - y\Theta^\top \mathbf{x} + 1 \right] =$$
$$2 \max \left[0, 1 - 1y\Theta^\top \mathbf{x} \right]$$

Structured SVM

- Structured SVMs minimize the following loss function:

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

- The standard binary-classification SVM is a special case where:

$$y \in \{-1, +1\}$$
$$s(y; \mathbf{x}, \Theta) = y\Theta^\top \mathbf{x}$$

$$\Delta(y, \hat{y}) = \begin{cases} 0 \text{ iff } y = \hat{y} \\ 1 \text{ iff } y \neq \hat{y} \end{cases}$$

- Working out the loss leads to:

$$\max \left[y\Theta^\top \mathbf{x} - y\Theta^\top \mathbf{x} + 0, (1 - y)\Theta^\top \mathbf{x} - y\Theta^\top \mathbf{x} + 1 \right] =$$
$$2 \max \left[0, 1 - 1y\Theta^\top \mathbf{x} \right]$$

hinge loss for binary classification

Structured SVM

- The gradient of structured SVM loss w.r.t. the model parameters is given by:

$$\nabla_{\Theta}\ell(\Theta;\mathbf{x},y) = \nabla_{\Theta}s(y^*;\mathbf{x},\Theta) - \nabla_{\Theta}s(y;\mathbf{x},\Theta)$$

  - where: $y^* = \underset{\hat{y}}{\operatorname{argmax}}\left(s(\hat{y};\mathbf{x},\Theta) + \Delta(y,\hat{y})\right)$

Structured SVM

- The gradient of structured SVM loss w.r.t. the model parameters is given by:

$$\nabla_{\Theta}\ell(\Theta;\mathbf{x},y) = \nabla_{\Theta}s(y^*;\mathbf{x},\Theta) - \nabla_{\Theta}s(y;\mathbf{x},\Theta)$$

  - where: $y^* = \underset{\hat{y}}{\operatorname{argmax}}\left(s(\hat{y};\mathbf{x},\Theta) + \Delta(y,\hat{y})\right)$

- This is a very natural way of saying:
  - The positive example is the detection with the highest score
  - The negative example is the detection with the second-highest score

Structured SVM: Application

- In detection, we aim to learn a function from image to bounding box + label
- Input $\boldsymbol{x}$ = *image*; output y = *(label, bounding box)*

Structured SVM: Application

- Training data for this problem takes the same form:

Structured SVM: Application

- To train a structured SVM, we need to define the *task loss*:

$\Delta(\mathbf{y}, \widehat{\mathbf{y}}) = \text{overlap err.}$

$\Delta(\mathbf{y}, \widehat{\mathbf{y}}) = 1$

$$\Delta(\mathbf{y}, \widehat{\mathbf{y}})$$

$\Delta(\mathbf{y}, \widehat{\mathbf{y}}) = 1$

$\Delta(\mathbf{y}, \widehat{\mathbf{y}}) = 0$

Structured SVM: Application

- How do we compute the loss (and the loss gradient) in this application?

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

Structured SVM: Application

- How do we compute the loss (and the loss gradient) in this application?

$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

  - Perform sliding-window search with the current detector!

Structured SVM: Application

- How do we compute the loss (and the loss gradient) in this application?

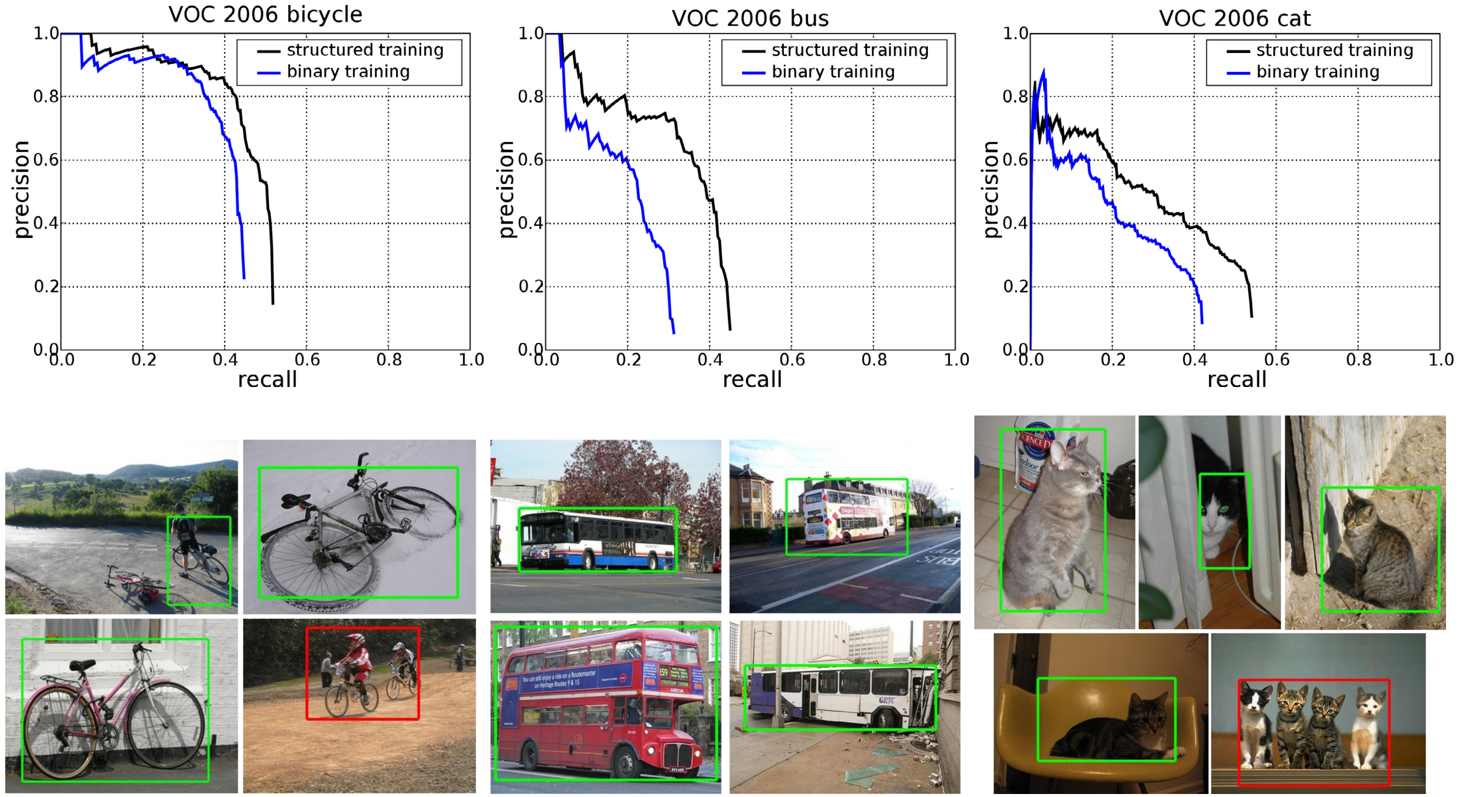
$$\ell(\Theta; \mathbf{x}, y) = \max_{\hat{y}} \left[s(\hat{y}; \mathbf{x}, \Theta) - s(y; \mathbf{x}, \Theta) + \Delta(y, \hat{y}) \right]$$

  - Perform sliding-window search with the current detector!

- For other structures, we may have efficient ways to do maximization, too:

  - For instance, the *Viterbi algorithm* searches over all label sequences

Structured SVM: Applications

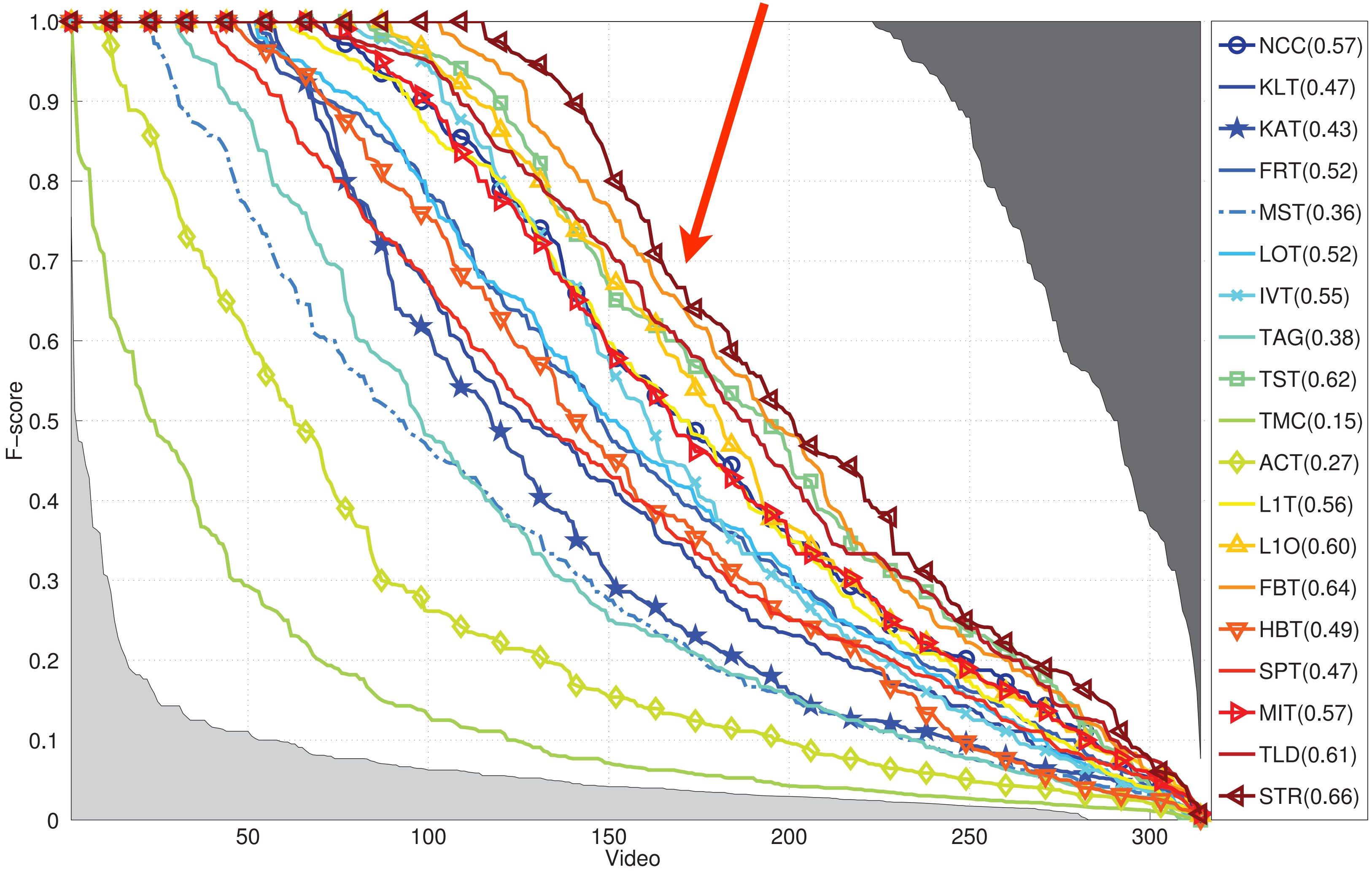


- Advantage of *structured training*: strongly overlapping bounding boxes may have almost the same score as the ground-truth

* Blaschko & Lampert, 2008

Structured SVM: Applications

- Structured output tracker (Struck; Hare *et al.*, 2010) currently state-of-the-art:



* Smeulders *et al.*, 2014

Questions?