Learning to Rank using High-Order Information: Tutorial

Puneet Kumar Dokania

(http://cvn.ecp.fr/personnel/puneet/)

Supervised by: M. Pawan Kumar

Ecole Centrale de Paris and INRIA Saclay

July 8, 2014

Presentation Outline

Action Recognition

Problem:

Action Recognition

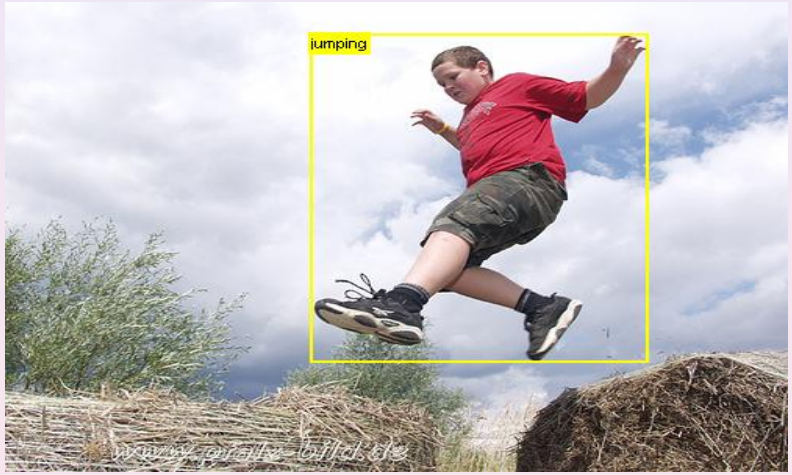
Problem:

- Given bounding box x
- predict the action

Action Recognition

Problem:

- Given bounding box x
- predict the action



Action Recognition

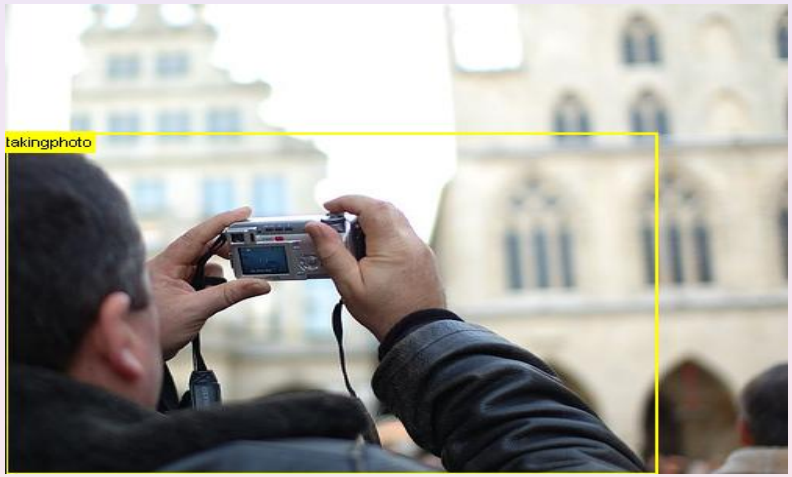
Problem:

- Given bounding box x
- predict the action

Action Recognition

Problem:

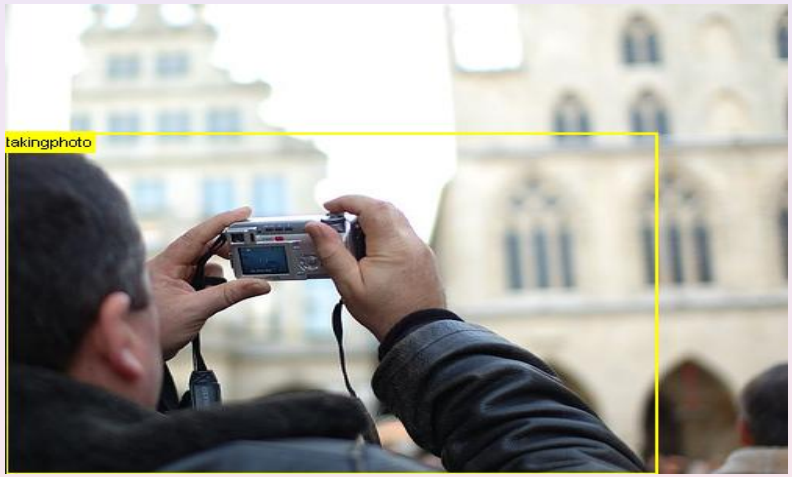
- Given bounding box x
- predict the action



Action Recognition

Problem:

- Given bounding box x
- predict the action

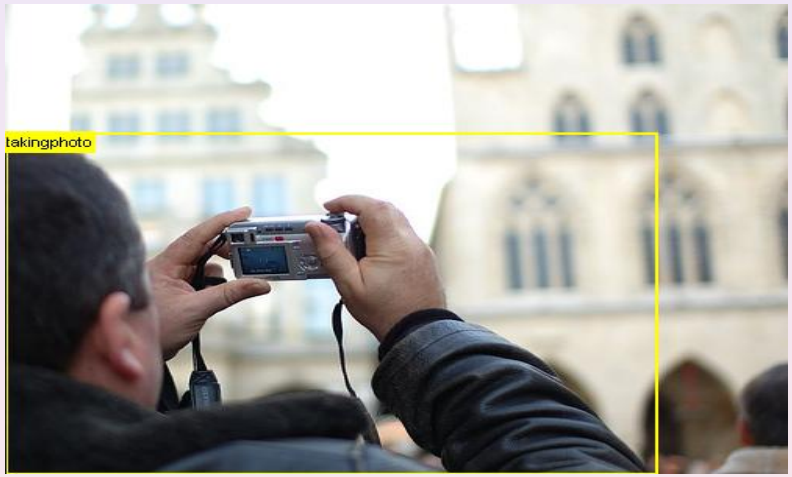


Solution (ML):

Action Recognition

Problem:

- Given bounding box x
- predict the action



Solution (ML):

- Extract features $\phi(x)$, such as POSELET, HOG, GIST, etc.
- Use Machine Learning in the feature space

Empirical Risk Minimization

Input

- $\mathcal{D} = \{(x_i, y_i), \cdots, (x_n, y_n)\}$ a training set in $(\mathcal{X} \times \mathcal{Y})^n$
- $\mathcal{X}$ the space of patterns or data (typically, $\mathcal{X} \in \mathcal{R}^p$)
- $\mathcal{Y}$ the space of labels: $\mathcal{Y} \in \{-1, 1\}$

Empirical Risk Minimization

Input

- $\mathcal{D} = \{(x_i, y_i), \cdots, (x_n, y_n)\}$ a training set in $(\mathcal{X} \times \mathcal{Y})^n$
- $\mathcal{X}$ the space of patterns or data (typically, $\mathcal{X} \in \mathcal{R}^p$)
- $\mathcal{Y}$ the space of labels: $\mathcal{Y} \in \{-1, 1\}$

Output

- A function $f : \mathcal{X} \rightarrow \mathcal{Y}$, predict output given input $x \in \mathcal{X}$

Empirical Risk Minimization ...

Empirical Risk Minimization ...

- Define a loss function $l(t, y)$, the loss between the prediction t and the true response y

Empirical Risk Minimization ...

- Define a loss function $l(t, y)$, the loss between the prediction t and the true response y
- The empirical risk for a candidate f is defined as

$$R(f) = \frac{1}{n} \sum_i l(f(x_i), y_i) \tag{1}$$

Empirical Risk Minimization ...

- Define a loss function $l(t, y)$, the loss between the prediction t and the true response y
- The empirical risk for a candidate f is defined as

$$R(f) = \frac{1}{n} \sum_i l(f(x_i), y_i) \tag{1}$$

- Estimate $\hat{f}$

$$\hat{f} = \underset{f}{\operatorname{argmin}}\, R(f) + \Omega(f) \tag{2}$$

Empirical Risk Minimization ...

- Define a loss function $l(t, y)$, the loss between the prediction t and the true response y
- The empirical risk for a candidate f is defined as

$$R(f) = \frac{1}{n} \sum_i l(f(x_i), y_i) \tag{1}$$

- Estimate $\hat{f}$

$$\hat{f} = \underset{f}{\operatorname{argmin}}\, R(f) + \Omega(f) \tag{2}$$

- Since we optimize $l(f(x), y)$, we must use the same loss function to evaluate the performance of the classifier

Presentation Outline

SSVM

- Given Dataset $\mathcal{D} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$

SSVM

- Given Dataset $\mathcal{D} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$
- Define Joint Feature Map (encodes the structure) $\Psi(x_i, y_i)$

SSVM

- Given Dataset $\mathcal{D} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$
- Define Joint Feature Map (encodes the structure) $\Psi(x_i, y_i)$
- Scoring Function $S(x, y; \mathbf{w}) = \mathbf{w}^\top \Psi(x, y)$

SSVM

- Given Dataset $\mathcal{D} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$
- Define Joint Feature Map (encodes the structure) $\Psi(x_i, y_i)$
- Scoring Function $S(x, y; \mathbf{w}) = \mathbf{w}^\top \Psi(x, y)$
- Prediction: $y = argmax_{\bar{y}} \mathbf{w}^\top \Psi(x, y)$

Optimization

Optimization

- Loss function $\Delta(y_i, y_i^*)$ for i^{th} sample

Optimization

- Loss function $\Delta(y_i, y_i^*)$ for i^{th} sample
- Optimization (ERM)

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \tag{3}$$

Optimization

- Loss function $\Delta(y_i, y_i^*)$ for i^{th} sample
- Optimization (ERM)

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \tag{3}$$

$$\text{s.t.} \quad \mathbf{w}^\top \Psi(x_i, y_i^*) - \mathbf{w}^\top \Psi(x_i, y_i) \geq \Delta(y_i, y_i^*) - \xi_i, \forall y_i, \forall i$$

Optimization

- Loss function $\Delta(y_i, y_i^*)$ for i^{th} sample
- Optimization (ERM)

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \tag{3}$$

$$\text{s.t.} \quad \mathbf{w}^\top \Psi(x_i, y_i^*) - \mathbf{w}^\top \Psi(x_i, y_i) \geq \Delta(y_i, y_i^*) - \xi_i, \forall y_i, \forall i$$

- Generally exponentially many constraints

Optimization

- Loss function $\Delta(y_i, y_i^*)$ for i^{th} sample
- Optimization (ERM)

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i \tag{3}$$

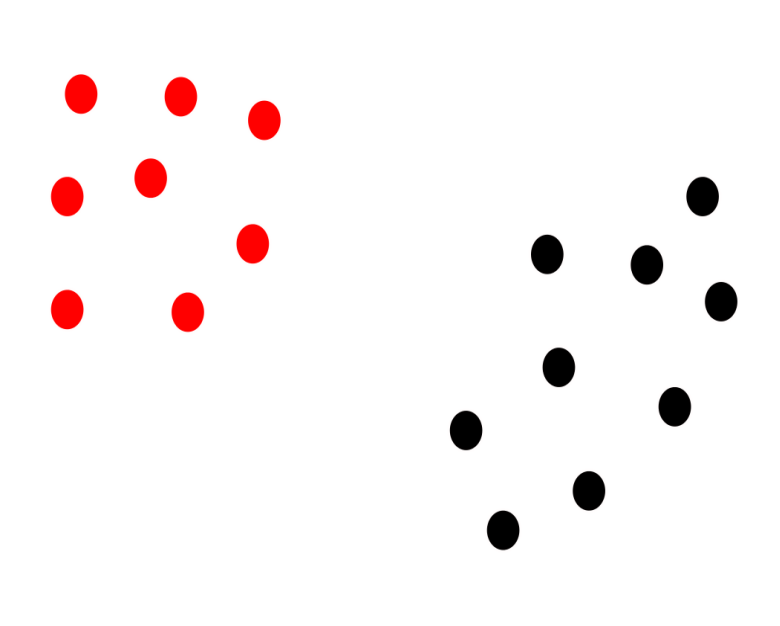
$$\text{s.t.} \quad \mathbf{w}^\top \Psi(x_i, y_i^*) - \mathbf{w}^\top \Psi(x_i, y_i) \geq \Delta(y_i, y_i^*) - \xi_i, \forall y_i, \forall i$$

- Generally exponentially many constraints
- Cutting plane - Loss augmented inference (most violated constraint)

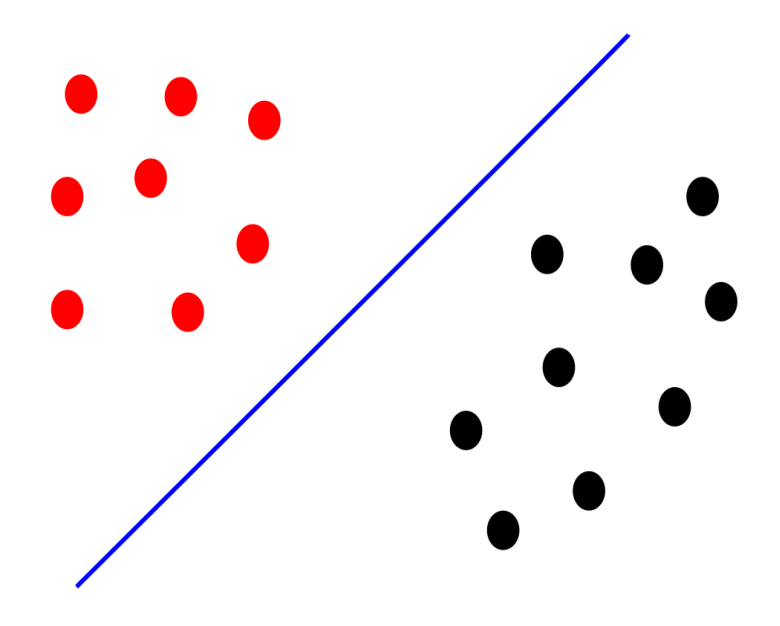
$$y = argmax_{\bar{y}}\{\mathbf{w}^\top \Psi(x_i, y_i) + \Delta(y_i, y_i^*)\}$$

SVM as a special case of SSVM

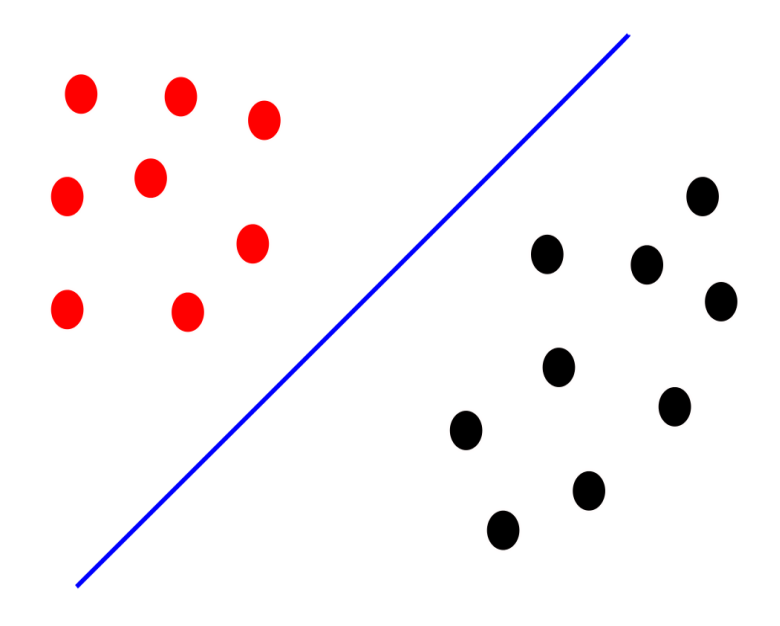
SVM as a special case of SSVM



SVM as a special case of SSVM



SVM as a special case of SSVM



- Given Dataset
  $\mathcal{D} = \{(x_i, y_i), \cdots, (x_n, y_n)\}$
- Feature mapping $\phi(x)$
- $\mathcal{L}_2$ regularized ERM

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 \quad + \quad C\sum_i \xi_i \tag{4}$$

$$\begin{aligned} \mathbf{w}^\top \phi(x_i) &\geq 1 - \xi_i, \forall y_i = +1 \\ \mathbf{w}^\top \phi(x_i) &\leq -1 + \xi_i, \forall y_i = -1 \\ \xi_i &\geq 0, \forall i \end{aligned}$$

- Prediction: Using sample scores
  $s(x; \mathbf{w}) = \mathbf{w}^\top \phi(x)$

Presentation Outline

Average Precision

Definition

Given true ranking r and predicted ranking $\hat{r}$

Average Precision

Definition

Given true ranking r and predicted ranking $\hat{r}$

$$AP(r, \hat{r}) = \frac{1}{rel} \sum_{j:r_j=1} Precision@j$$

where, $rel = |i : r_i = 1|$ is the number of relevant document. $Precision@j$ is the percentage of relevant documents.

Average Precision

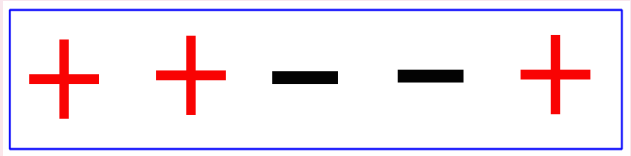
Definition

Given true ranking r and predicted ranking $\hat{r}$

$$AP(r, \hat{r}) = \frac{1}{rel} \sum_{j:r_j=1} Precision@j$$

where, $rel = |i : r_i = 1|$ is the number of relevant document. $Precision@j$ is the percentage of relevant documents.

Example:



Average Precision

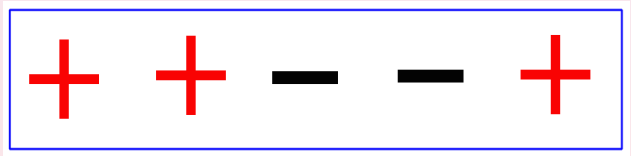
Definition

Given true ranking r and predicted ranking $\hat{r}$

$$AP(r, \hat{r}) = \frac{1}{rel} \sum_{j:r_j=1} Precision@j$$

where, $rel = |i : r_i = 1|$ is the number of relevant document. $Precision@j$ is the percentage of relevant documents.

Example:



$$\text{AP} = (\tfrac{1}{1} + \tfrac{2}{2} + \tfrac{3}{5})\tfrac{1}{3}$$

AP vs Accuracy

Relevant class: Jumping

AP vs Accuracy

Relevant class: Jumping

AP vs Accuracy

Relevant class: Jumping

$AP = (\frac{1}{1} + \frac{2}{2} + \frac{3}{3})\frac{1}{3} = 1.0$

$Accuracy = 6/6 = 1.0$

AP vs Accuracy

Relevant class: Jumping

$AP = (\frac{1}{1} + \frac{2}{2} + \frac{3}{3})\frac{1}{3} = 1.0$

$Accuracy = 6/6 = 1.0$

AP vs Accuracy

Relevant class: Jumping

$$AP = (\frac{1}{1} + \frac{2}{2} + \frac{3}{3})\frac{1}{3} = 1.0$$

$$Accuracy = 6/6 = 1.0$$

$$AP = (\frac{1}{2} + \frac{2}{3} + \frac{3}{6})\frac{1}{3} = 0.55$$

$$Accuracy = 1.0$$

Ranking or Classification?

Ranking or Classification?

- Most of the evaluations are based on AP - Ranking
- We have seen Ranking and Classification are not the same
- Optimizing Accuracy (0-1 loss) may lead to suboptimal AP (basic Machine Learning principle)

Ranking or Classification?

- Most of the evaluations are based on AP - Ranking
- We have seen Ranking and Classification are not the same
- Optimizing Accuracy (0-1 loss) may lead to suboptimal AP (basic Machine Learning principle)
- SVM optimizes 0-1 loss - Good for classfication
- Using AP based loss function directly into SSVM framework leads to intractable loss augmented inference

Presentation Outline

Problem Formulation

Problem Formulation

- Single input $\mathbf{x}$, Positive Set $\mathcal{P}$, Negative Set $\mathcal{N}$
- $\phi(x_i), \forall i \in \mathcal{P}$, $\phi(x_j), \forall j \in \mathcal{N}$

Problem Formulation

- Single input $\mathbf{x}$, Positive Set $\mathcal{P}$, Negative Set $\mathcal{N}$
- $\phi(x_i), \forall i \in \mathcal{P}$, $\phi(x_j), \forall j \in \mathcal{N}$
- Ranking

$$R_{ij} = \begin{cases} +1, & \text{if i is better ranked than j} \\ -1, & \text{if j is better ranked than i} \end{cases}$$

Problem Formulation ...

- Formulate the problem as a structure prediction problem

Problem Formulation ...

- Formulate the problem as a structure prediction problem
- Joint feature map $\psi(\mathbf{x}, R)$

$$\psi(\mathbf{x}, R) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} R_{ij}(\phi(x_i) - \phi(x_j))$$

- Loss function $\Delta(R, R^*) = 1 - AP(R, R^*)$

Problem Formulation ...

- Formulate the problem as a structure prediction problem
- Joint feature map $\psi(\mathbf{x}, R)$

$$\psi(\mathbf{x}, R) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} R_{ij}(\phi(x_i) - \phi(x_j))$$

- Loss function $\Delta(R, R^*) = 1 - AP(R, R^*)$
- Score $S(\mathbf{x}, R; \mathbf{w}) = \mathbf{w}^\top \psi(\mathbf{x}, R)$

Problem Formulation ...

- Formulate the problem as a structure prediction problem
- Joint feature map $\psi(\mathbf{x}, R)$

$$\psi(\mathbf{x}, R) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} R_{ij}(\phi(x_i) - \phi(x_j))$$

- Loss function $\Delta(R, R^*) = 1 - AP(R, R^*)$
- Score $S(\mathbf{x}, R; \mathbf{w}) = \mathbf{w}^\top \psi(\mathbf{x}, R)$
- Prediction (Ranking): $R = argmax_{\bar{R}} S(\mathbf{x}, \bar{R}; \mathbf{w})$,

Problem Formulation ...

- Formulate the problem as a structure prediction problem
- Joint feature map $\psi(\mathbf{x}, R)$

$$\psi(\mathbf{x}, R) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} R_{ij}(\phi(x_i) - \phi(x_j))$$

- Loss function $\Delta(R, R^*) = 1 - AP(R, R^*)$
- Score $S(\mathbf{x}, R; \mathbf{w}) = \mathbf{w}^\top \psi(\mathbf{x}, R)$
- Prediction (Ranking): $R = argmax_{\bar{R}} S(\mathbf{x}, \bar{R}; \mathbf{w})$, sort individual scores $s(x_i; w) = \mathbf{w}^\top \phi(\mathbf{x}_i)$, $\mathcal{O}(nlogn)$

Optimization

Optimization

- Objective Function

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{5}$$

Optimization

- Objective Function

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{5}$$
$$\text{s.t.} \quad S(\mathbf{x}, R^*; \mathbf{w}) - S(\mathbf{x}, R; \mathbf{w}) \geq \Delta(R, R^*) - \xi, \forall R$$

Optimization

- Objective Function

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{5}$$
$$\text{s.t.} \quad S(\mathbf{x}, R^*; \mathbf{w}) - S(\mathbf{x}, R; \mathbf{w}) \geq \Delta(R, R^*) - \xi, \forall R$$

- Cutting Plane (loss augmented inference):
  $\bar{R} = argmax_R\{S(\mathbf{x}, \bar{R}; \mathbf{w}) + \Delta(R, R^*)\}$,

Optimization

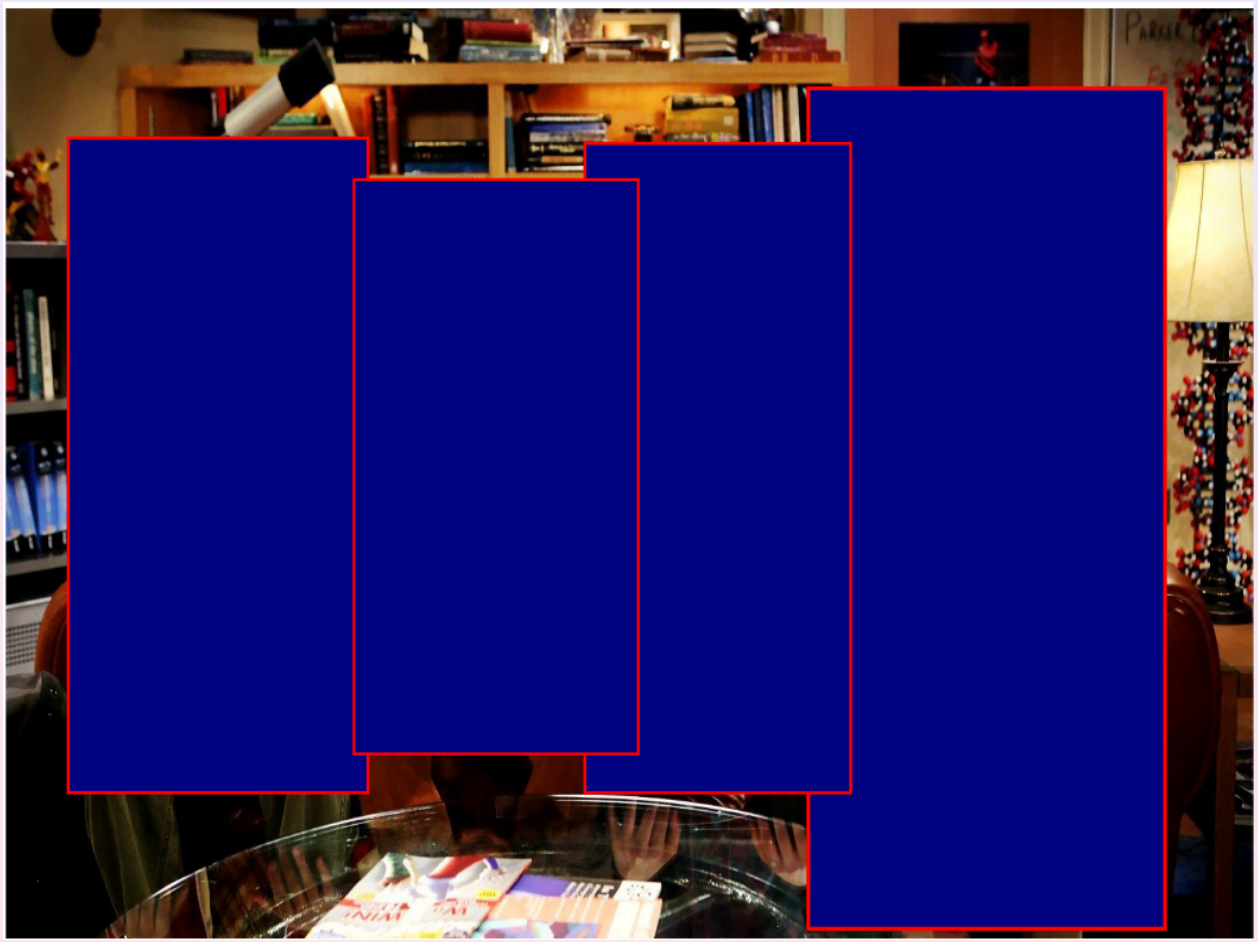
- Objective Function

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{5}$$
$$\text{s.t.} \quad S(\mathbf{x}, R^*; \mathbf{w}) - S(\mathbf{x}, R; \mathbf{w}) \geq \Delta(R, R^*) - \xi, \forall R$$

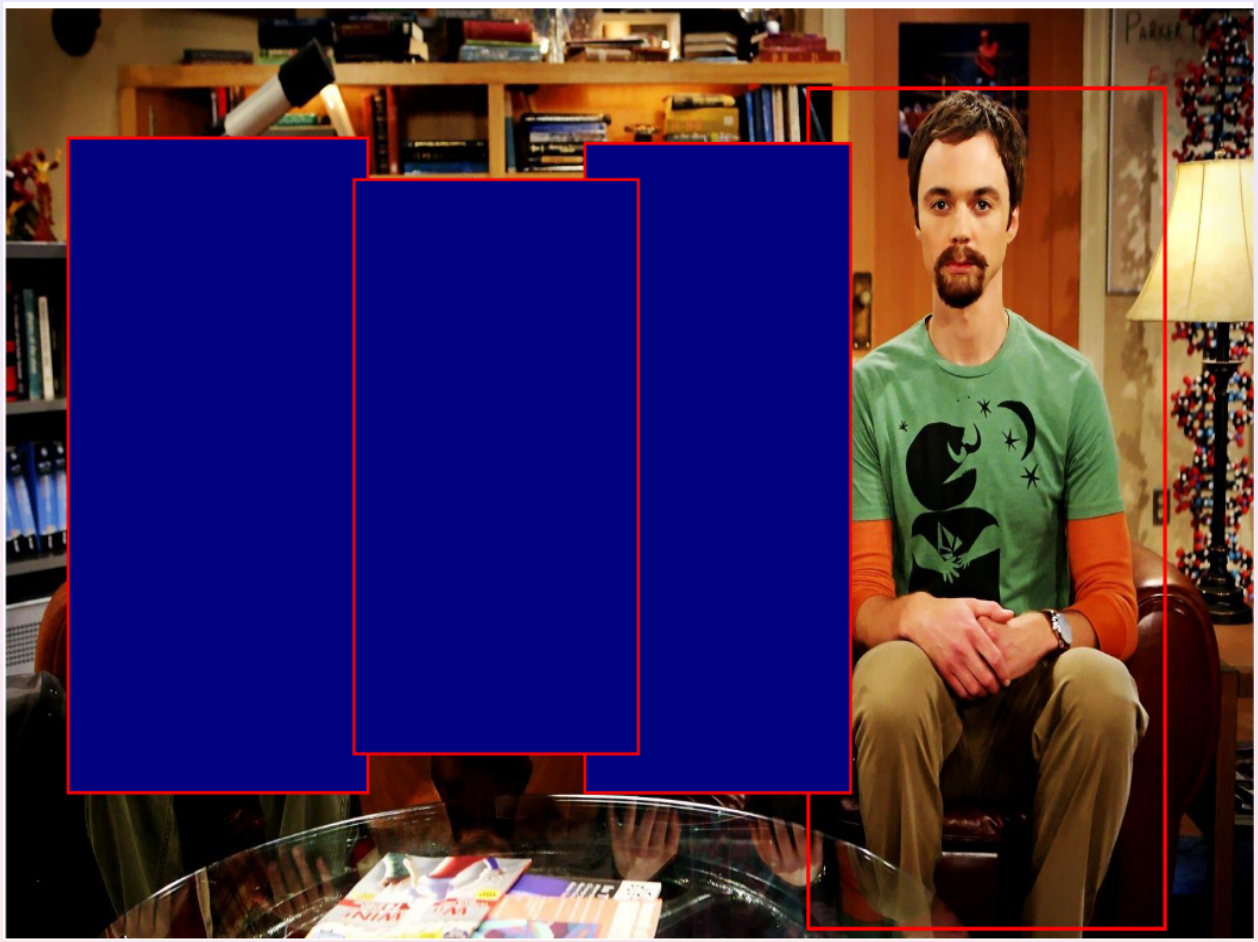
- Cutting Plane (loss augmented inference):
  $\bar{R} = argmax_R\{S(\mathbf{x}, \bar{R}; \mathbf{w}) + \Delta(R, R^*)\}$, greedy algorithm $\mathcal{O}(|\mathcal{P}||\mathcal{N}|)$ by Yue et.al.

Presentation Outline

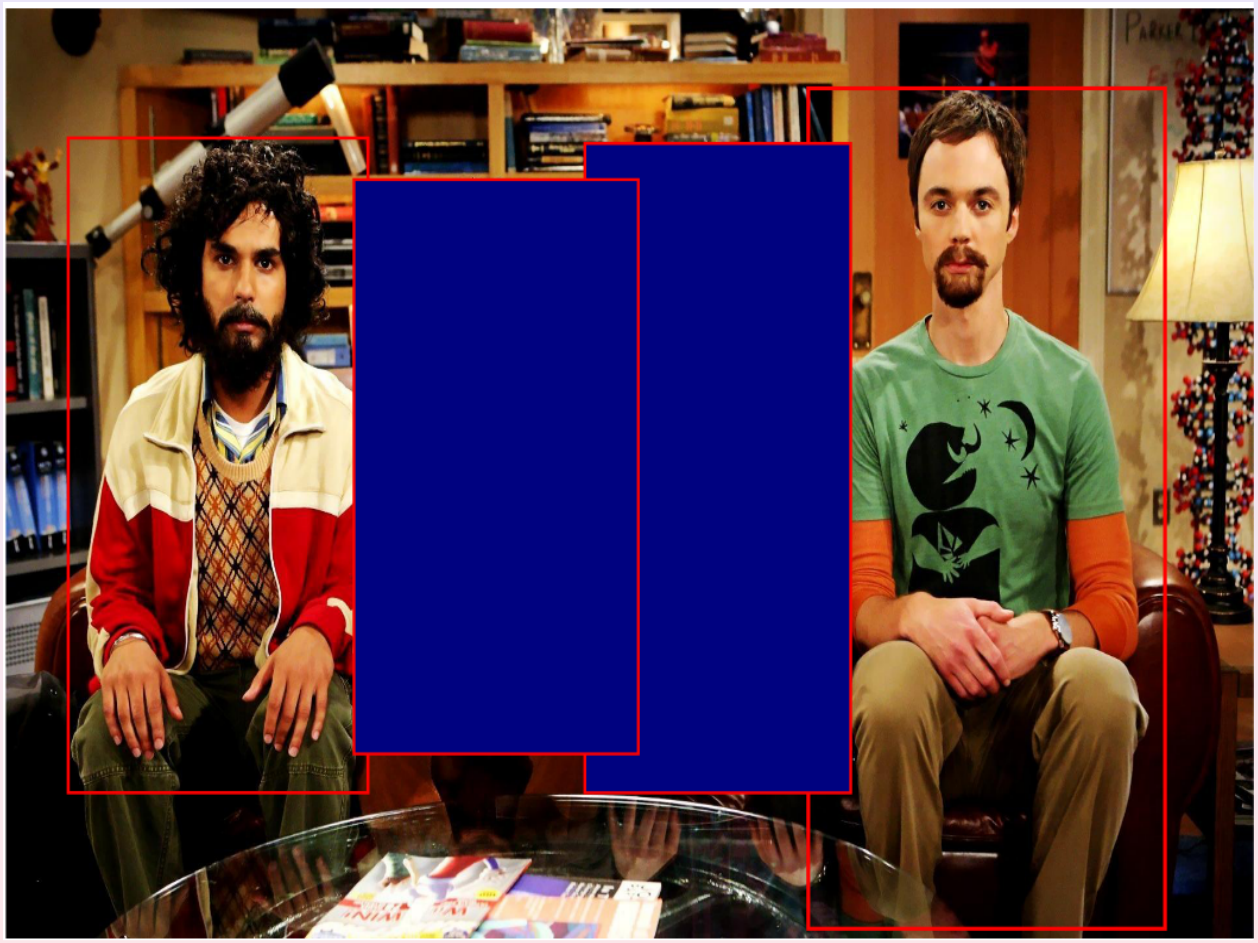
Example



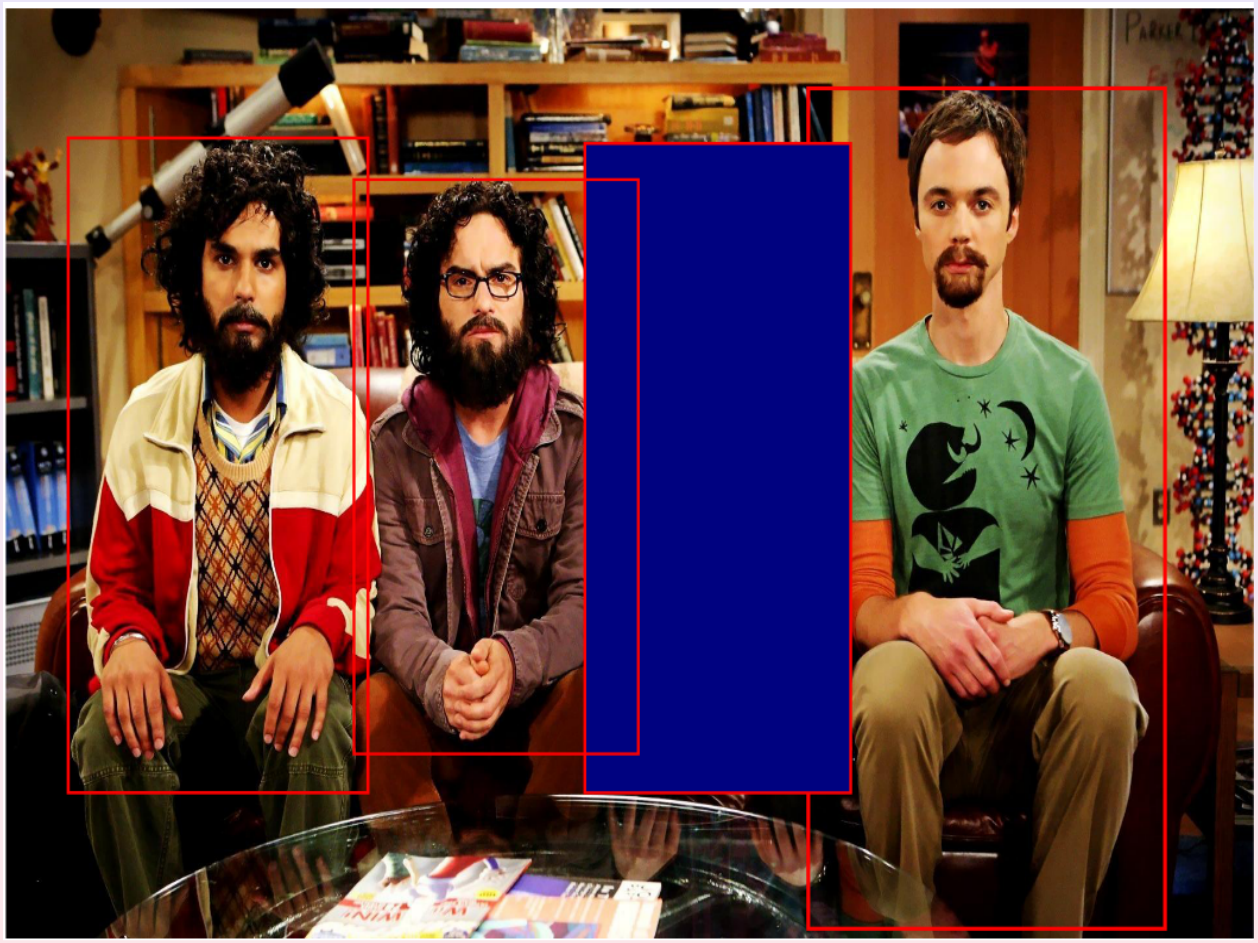
Example



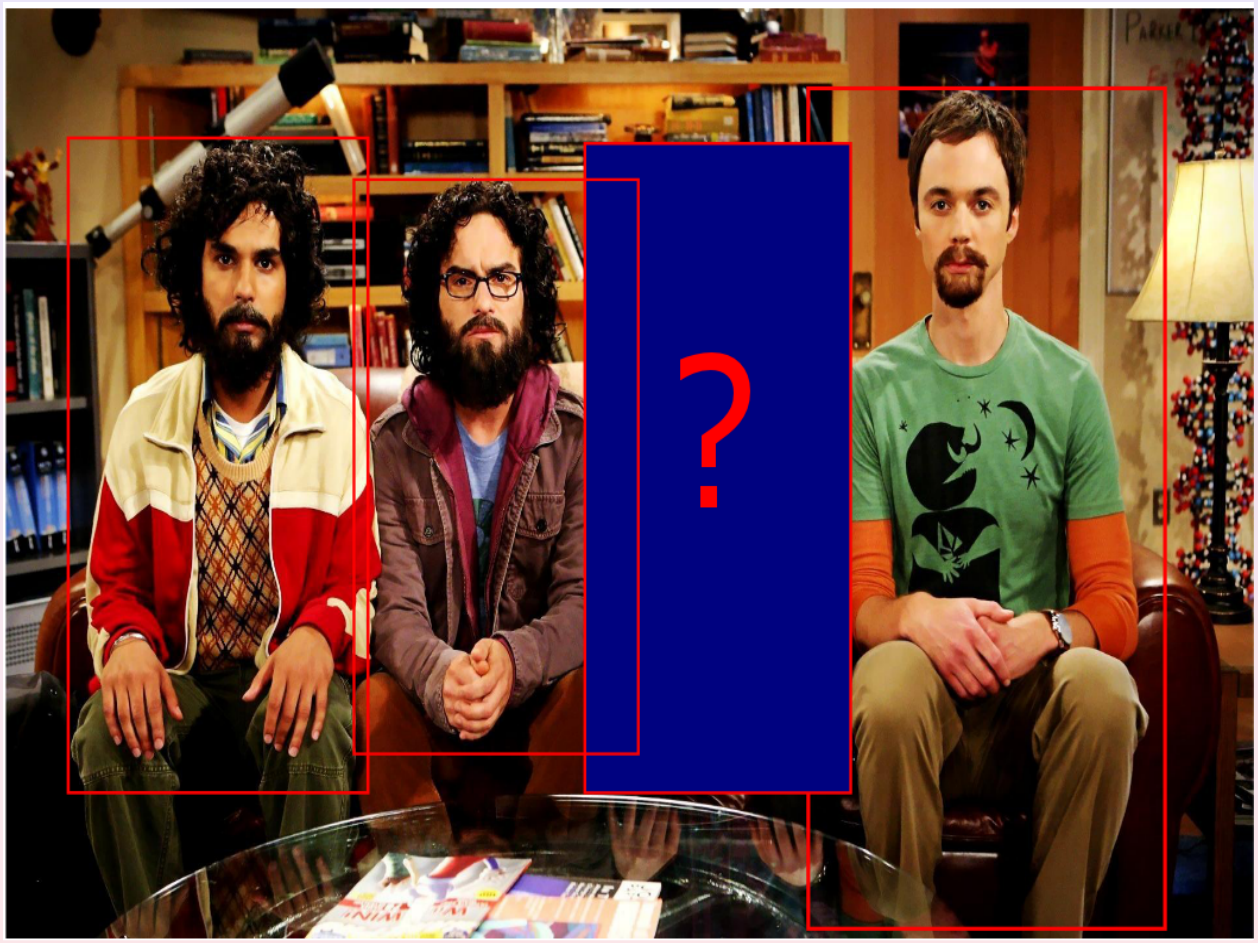
Example



Example



Example



Example

Why High-Order Information

Why High-Order Information

- Improves our prediction confidence

Why High-Order Information

- Improves our prediction confidence
- We must use it if available

Why High-Order Information

- Improves our prediction confidence
- We must use it if available
  - people strike similar poses
  - objects are of same/similar size
  - Friends have similar habit
  - Many more . . .

Why High-Order Information

- Improves our prediction confidence
- We must use it if available
  - people strike similar poses
  - objects are of same/similar size
  - Friends have similar habit
  - Many more . . .
- More Examples

Why High-Order Information

- Improves our prediction confidence
- We must use it if available
  - people strike similar poses
  - objects are of same/similar size
  - Friends have similar habit
  - Many more . . .
- More Examples

Why High-Order Information

- Improves our prediction confidence
- We must use it if available
  - people strike similar poses
  - objects are of same/similar size
  - Friends have similar habit
  - Many more . . .
- More Examples

Why High-Order Information

- Improves our prediction confidence
- We must use it if available
  - people strike similar poses
  - objects are of same/similar size
  - Friends have similar habit
  - Many more . . .
- More Examples

Why High-Order Information

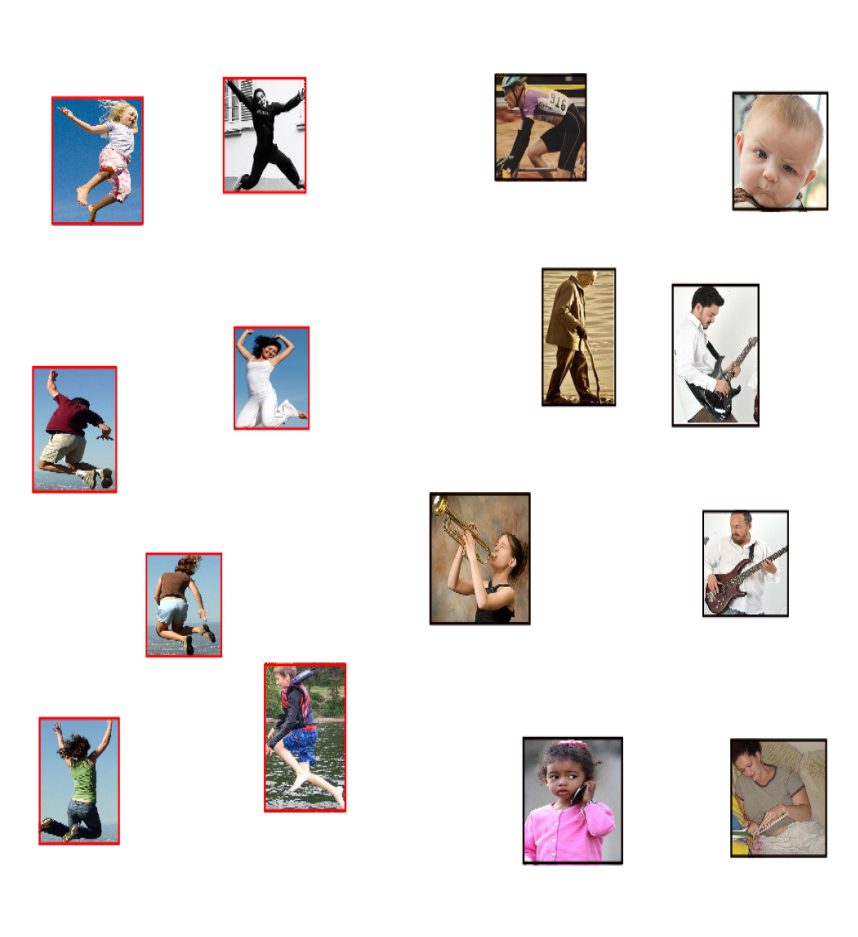
- Improves our prediction confidence
- We must use it if available
  - people strike similar poses
  - objects are of same/similar size
  - Friends have similar habit
  - Many more . . .
- More Examples

Why High-Order Information

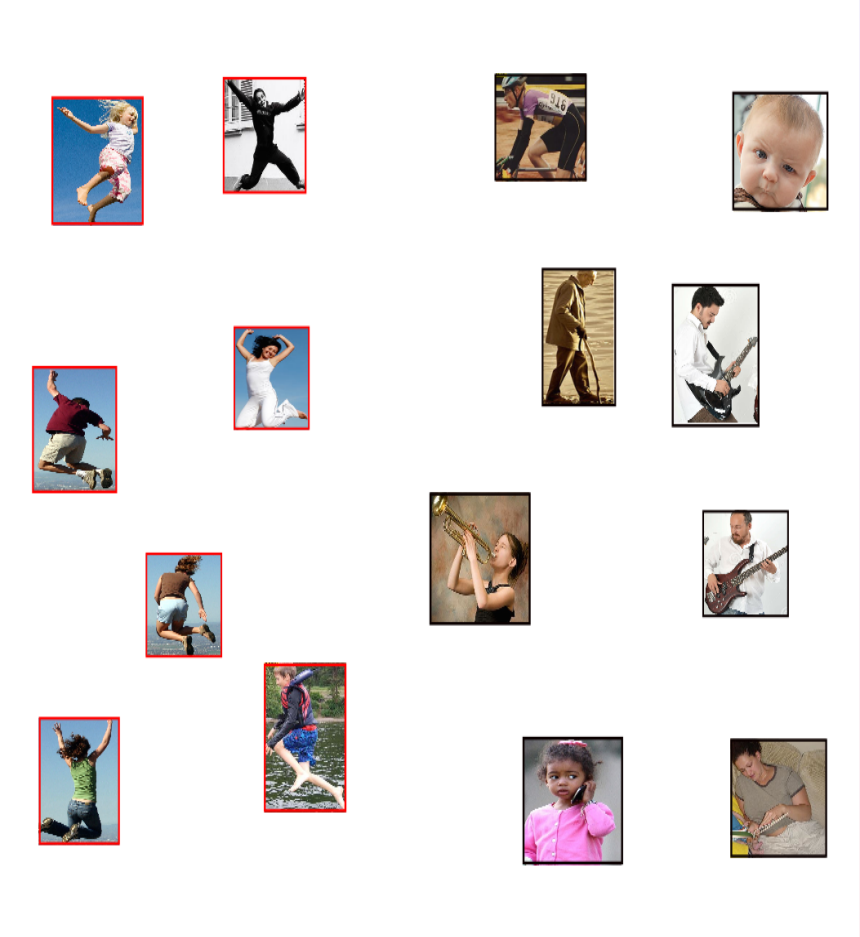
- Improves our prediction confidence
- We must use it if available
  - people strike similar poses
  - objects are of same/similar size
  - Friends have similar habit
  - Many more . . .
- More Examples

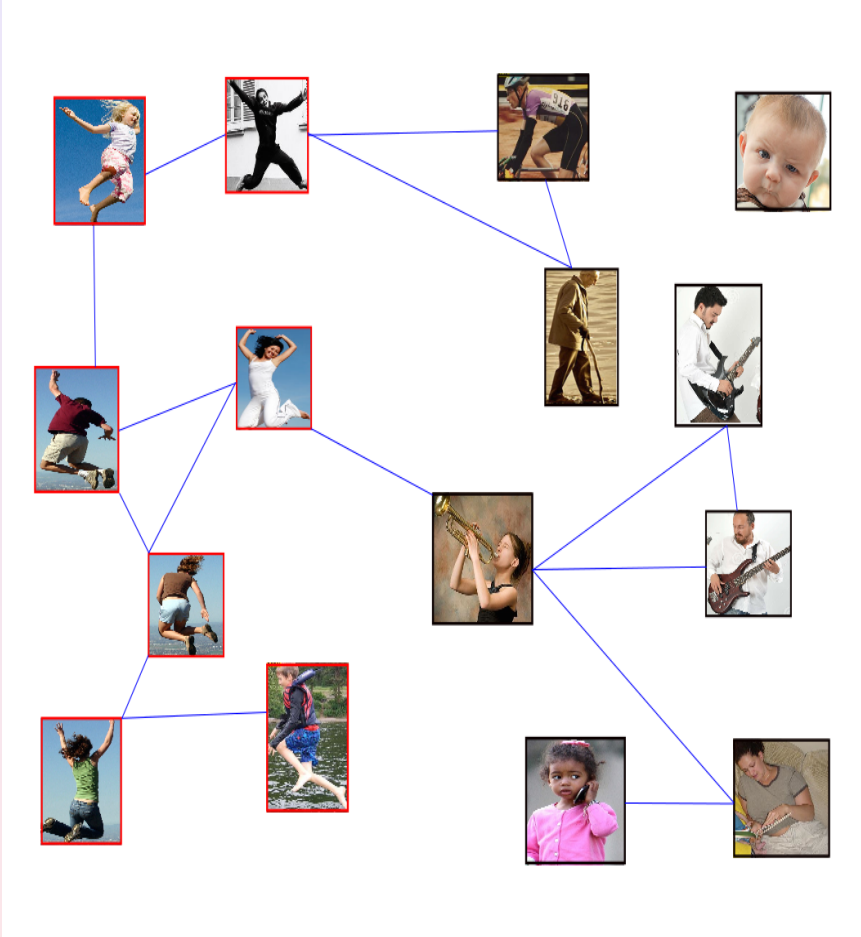
Presentation Outline

Structure (Jumping vs Others)

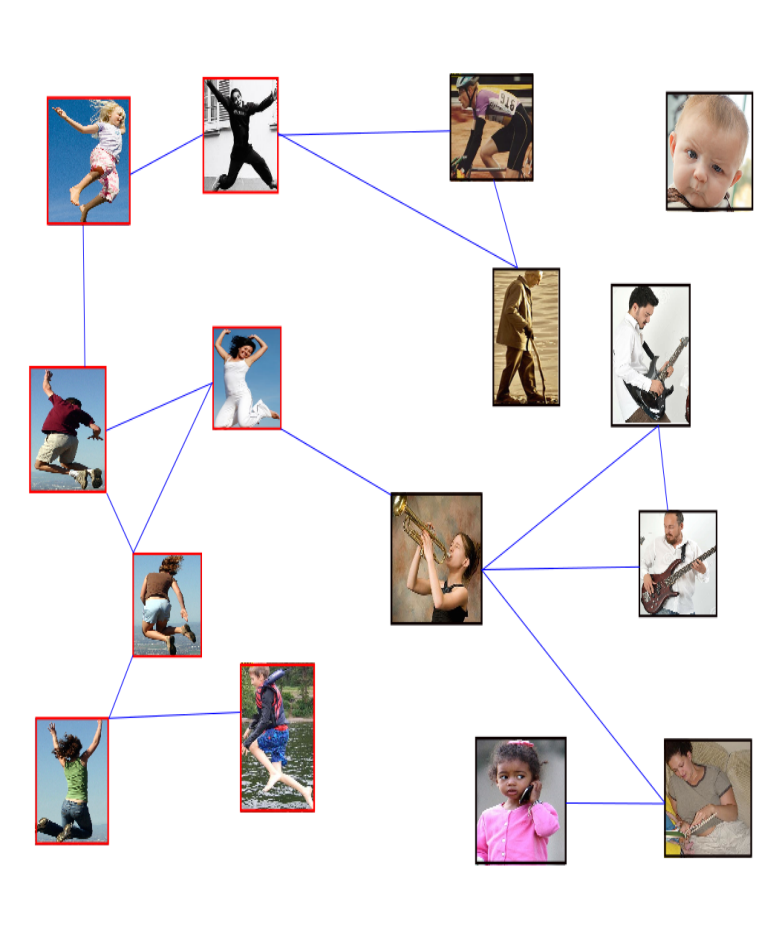


Structure (Jumping vs Others)

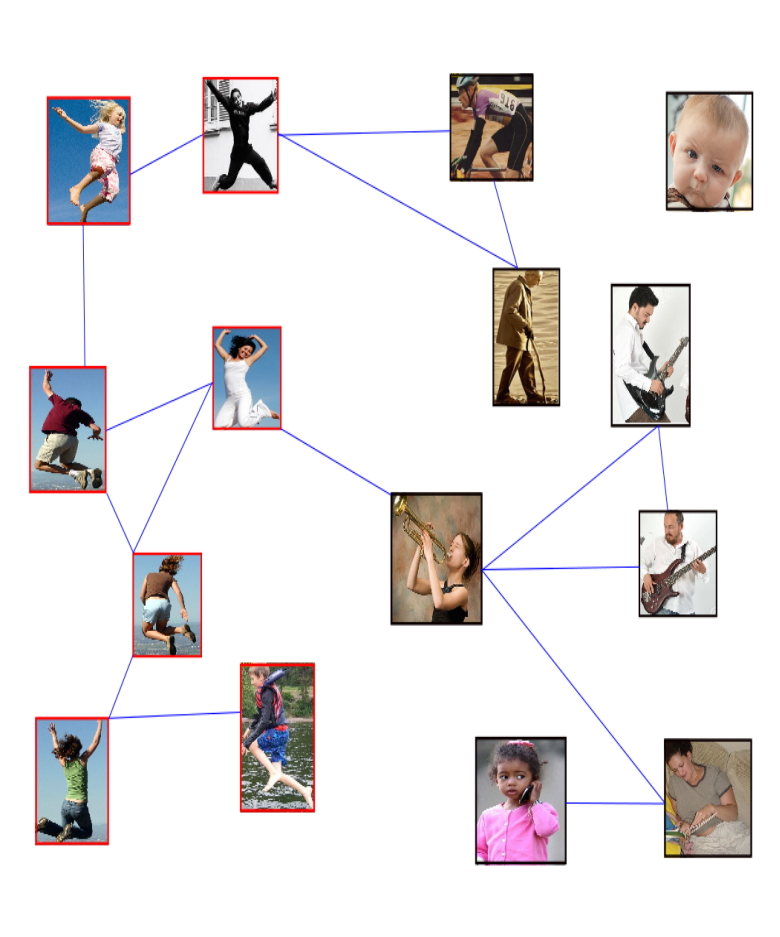




Problem Formulation



Problem Formulation

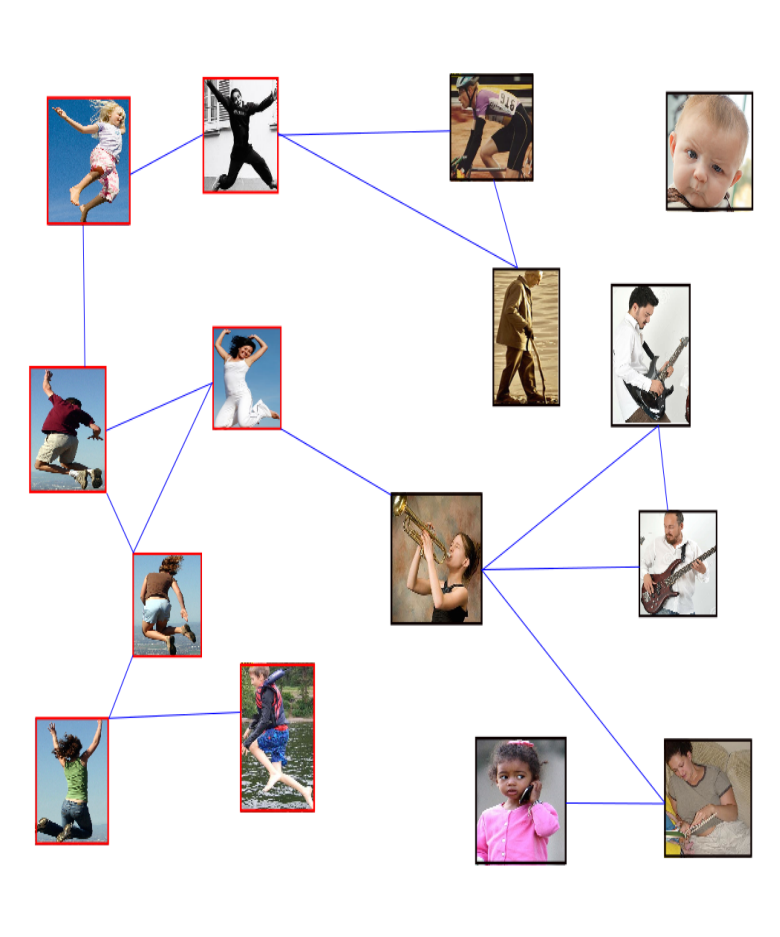


- Define Joint Feature Map (encodes the structure)

$$\Psi(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \sum_i \psi_1(x_i, y_i) \\ \sum_{i,j} \psi_2(x_i, y_i, x_j, y_j) \end{pmatrix}$$

  - ψ_1 - first-order information
  - ψ_2 - high-order information
- $\mathbf{y} \in \{-1, +1\}^n$

Problem Formulation



- Define Joint Feature Map (encodes the structure)

$$\Psi(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \sum_i \psi_1(x_i, y_i) \\ \sum_{i,j} \psi_2(x_i, y_i, x_j, y_j) \end{pmatrix}$$

  - ψ_1 - first-order information
  - ψ_2 - high-order information
- $\mathbf{y} \in \{-1, +1\}^n$
- Prediction: $y = argmax_{\bar{y}} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$

Prediction - closer look

$$\begin{aligned}
\mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) &= \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}^\top \begin{pmatrix} \sum_i \psi_1(x_i, y_i) \\ \sum_{i,j} \psi_2(x_i, y_i, x_j, y_j) \end{pmatrix} \\
&= \sum_i \mathbf{w}_1^\top \psi_1(x_i, y_i) + \sum_{i,j} \mathbf{w}_2^\top \psi_2(x_i, y_i, x_j, y_j) \qquad (6)
\end{aligned}$$

Prediction - closer look

$$\begin{aligned}
\mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) &= \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}^\top \begin{pmatrix} \sum_i \psi_1(x_i, y_i) \\ \sum_{i,j} \psi_2(x_i, y_i, x_j, y_j) \end{pmatrix} \\
&= \sum_i \mathbf{w}_1^\top \psi_1(x_i, y_i) + \sum_{i,j} \mathbf{w}_2^\top \psi_2(x_i, y_i, x_j, y_j) \qquad (6)
\end{aligned}$$

- Prediction (or Inference): $y = argmax_{\bar{y}} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$

Prediction - closer look

$$\begin{aligned}
\mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) &= \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}^\top \begin{pmatrix} \sum_i \psi_1(x_i, y_i) \\ \sum_{i,j} \psi_2(x_i, y_i, x_j, y_j) \end{pmatrix} \\
&= \sum_i \mathbf{w}_1^\top \psi_1(x_i, y_i) + \sum_{i,j} \mathbf{w}_2^\top \psi_2(x_i, y_i, x_j, y_j) \qquad (6)
\end{aligned}$$

- Prediction (or Inference): $y = argmax_{\bar{y}} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$
  - NP-HARD in general

Prediction - closer look

$$\begin{aligned} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) &= \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix}^\top \begin{pmatrix} \sum_i \psi_1(x_i, y_i) \\ \sum_{i,j} \psi_2(x_i, y_i, x_j, y_j) \end{pmatrix} \\ &= \sum_i \mathbf{w}_1^\top \psi_1(x_i, y_i) + \sum_{i,j} \mathbf{w}_2^\top \psi_2(x_i, y_i, x_j, y_j) \end{aligned} \quad (6)$$

- Prediction (or Inference): $y = argmax_{\bar{y}} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$
  - NP-HARD in general
  - Graph Cut if High-Order **supermodular** ($\mathbf{w}_2 \leq 0$)
  - Approximate solution using LP relaxations if not supermodular

Optimization

Optimization

- Loss function $\Delta(\mathbf{y}, \mathbf{y}^*)$: fraction of misclassification

Optimization

- Loss function $\Delta(\mathbf{y}, \mathbf{y}^*)$: fraction of misclassification
- Define Score $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$

Optimization

- Loss function $\Delta(\mathbf{y}, \mathbf{y}^*)$: fraction of misclassification
- Define Score $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$
- Optimization (ERM)

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{7}$$

Optimization

- Loss function $\Delta(\mathbf{y}, \mathbf{y}^*)$: fraction of misclassification
- Define Score $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$
- Optimization (ERM)

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \\ \text{s.t.} \quad & S(\mathbf{x}, \mathbf{y}^*; \mathbf{w}) - S(\mathbf{x}, \mathbf{y}; \mathbf{w}) \geq \Delta(\mathbf{y}, \mathbf{y}^*) - \xi, \\ & \mathbf{w}_2 \leq 0, \xi \geq 0, \forall \mathbf{y} \end{aligned} \tag{7}$$

Optimization

- Loss function $\Delta(\mathbf{y}, \mathbf{y}^*)$: fraction of misclassification
- Define Score $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$
- Optimization (ERM)

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \\ \text{s.t.} \quad & S(\mathbf{x}, \mathbf{y}^*; \mathbf{w}) - S(\mathbf{x}, \mathbf{y}; \mathbf{w}) \geq \Delta(\mathbf{y}, \mathbf{y}^*) - \xi, \\ & \mathbf{w}_2 \leq 0, \xi \geq 0, \forall \mathbf{y} \end{aligned} \tag{7}$$

- Exponentially many constraints - 2^n for binary

Optimization

- Loss function $\Delta(\mathbf{y}, \mathbf{y}^*)$: fraction of misclassification
- Define Score $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$
- Optimization (ERM)

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \\ \text{s.t.} \quad & S(\mathbf{x}, \mathbf{y}^*; \mathbf{w}) - S(\mathbf{x}, \mathbf{y}; \mathbf{w}) \geq \Delta(\mathbf{y}, \mathbf{y}^*) - \xi, \\ & \mathbf{w}_2 \leq 0, \xi \geq 0, \forall \mathbf{y} \end{aligned} \tag{7}$$

- Exponentially many constraints - 2^n for binary
- Cutting plane - Loss augmented inference (most violated constraint)

$$y = argmax_{\bar{y}} \{\mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}^*)\}$$

Optimization

- Loss function $\Delta(\mathbf{y}, \mathbf{y}^*)$: fraction of misclassification
- Define Score $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$
- Optimization (ERM)

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \\ \text{s.t.} \quad & S(\mathbf{x}, \mathbf{y}^*; \mathbf{w}) - S(\mathbf{x}, \mathbf{y}; \mathbf{w}) \geq \Delta(\mathbf{y}, \mathbf{y}^*) - \xi, \\ & \mathbf{w}_2 \leq 0, \xi \geq 0, \forall \mathbf{y} \end{aligned} \tag{7}$$

- Exponentially many constraints - 2^n for binary
- Cutting plane - Loss augmented inference (most violated constraint)

$$y = argmax_{\bar{y}} \{\mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}^*)\}$$

- $\Delta(y, y^*)$ is decomposable, therefore, loss augmented inference is similar to prediction.

Ranking?

- So far, the problem formulation is a trivial application of SSVM

Ranking?

- So far, the problem formulation is a trivial application of SSVM
- Single score for the whole dataset $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$

Ranking?

- So far, the problem formulation is a trivial application of SSVM
- Single score for the whole dataset $S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$
- Need score for each sample x_i to get the ranking
- We also need to keep the structure intact
- No trivial method to resolve these issues

How to get the Ranking

- We propose to use difference of max-marginals

How to get the Ranking

- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i(+) - m_i(-)$,

How to get the Ranking

- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i(+) - m_i(-)$, where, $m_i(+)$ is the max-marginal score such that sample x_i takes label of $+1$.

How to get the Ranking

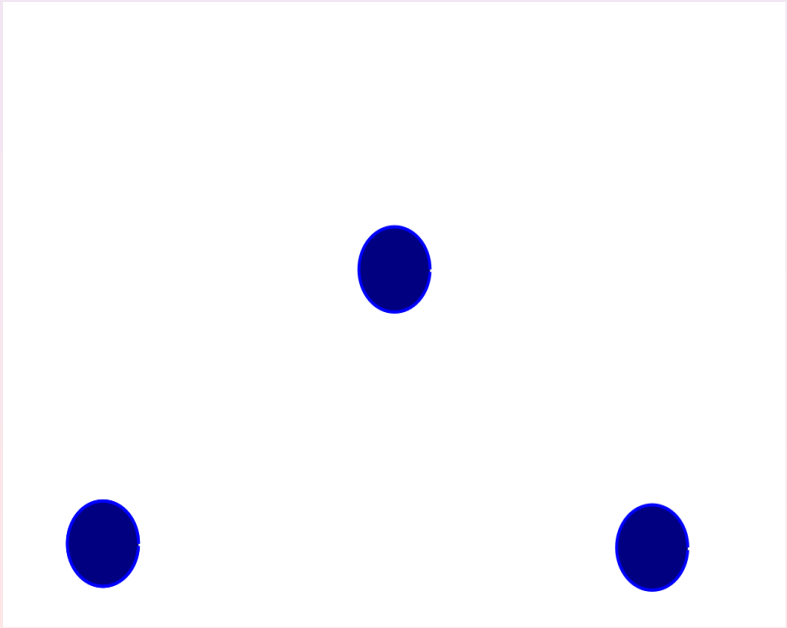
- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i(+) - m_i(-)$, where, $m_i(+)$ is the max-marginal score such that sample x_i takes label of $+1$.

$$m_i(+) = argmax_{\mathbf{y}, y_i=+1} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$$

How to get the Ranking

- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i(+) - m_i(-)$, where, $m_i(+)$ is the max-marginal score such that sample x_i takes label of $+1$.

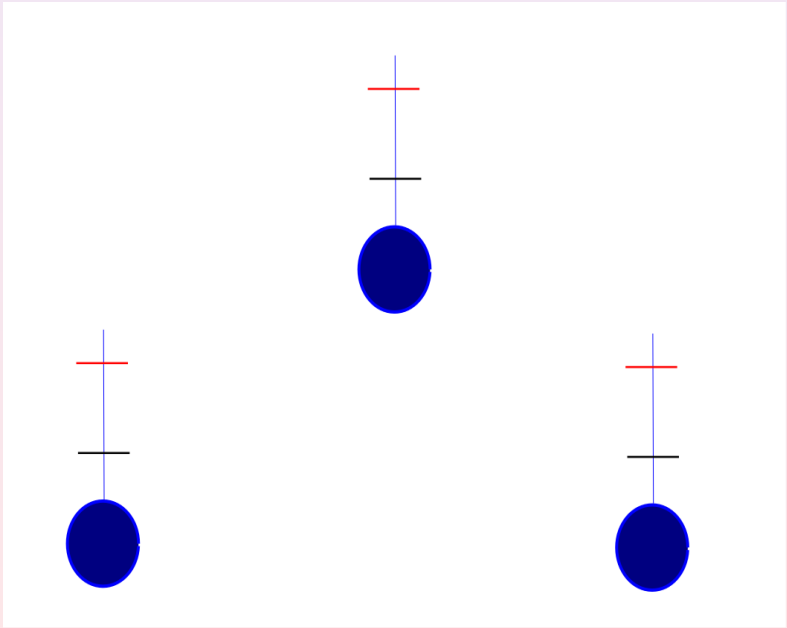
$$m_i(+) = argmax_{\mathbf{y}, y_i=+1} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$$



How to get the Ranking

- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i(+) - m_i(-)$, where, $m_i(+)$ is the max-marginal score such that sample x_i takes label of $+1$.

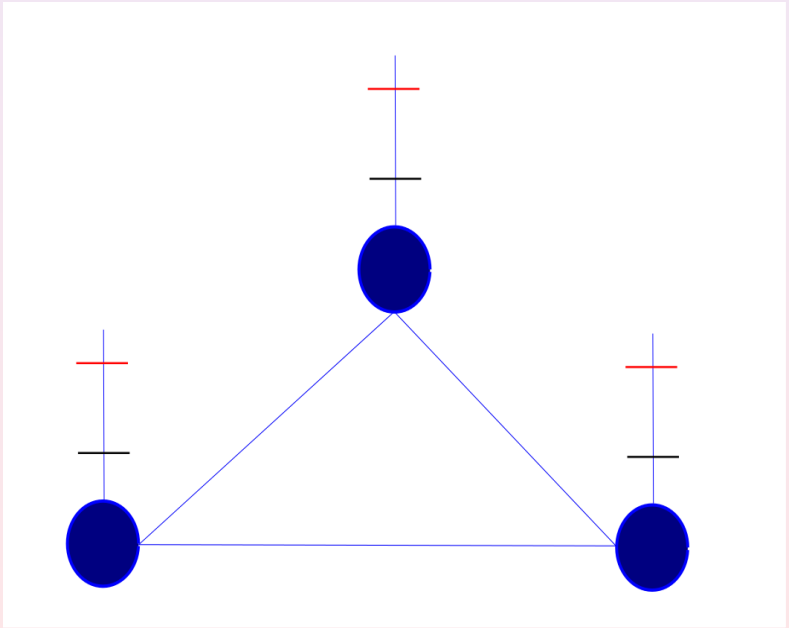
$$m_i(+) = argmax_{\mathbf{y}, y_i=+1} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$$



How to get the Ranking

- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i(+) - m_i(-)$, where, $m_i(+)$ is the max-marginal score such that sample x_i takes label of $+1$.

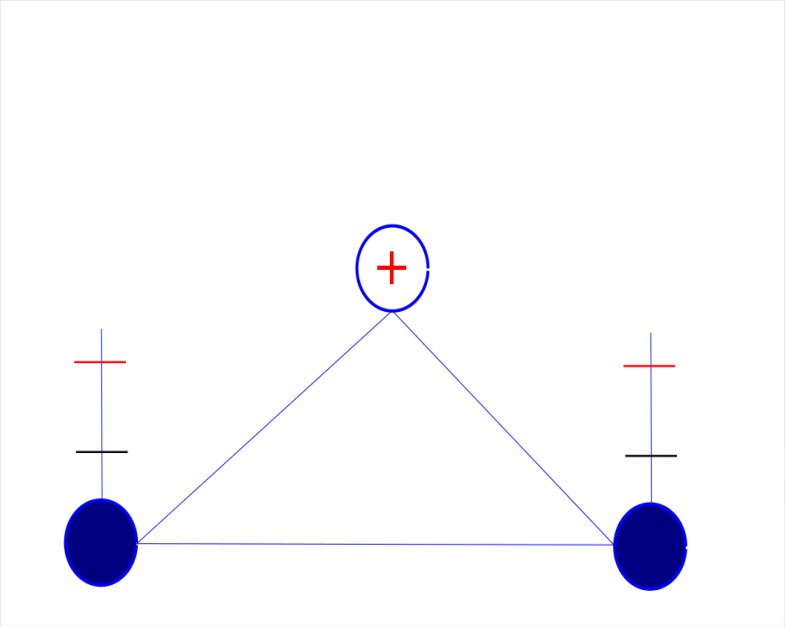
$$m_i(+) = argmax_{\mathbf{y}, y_i=+1} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$$



How to get the Ranking

- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i(+) - m_i(-)$, where, $m_i(+)$ is the max-marginal score such that sample x_i takes label of $+1$.

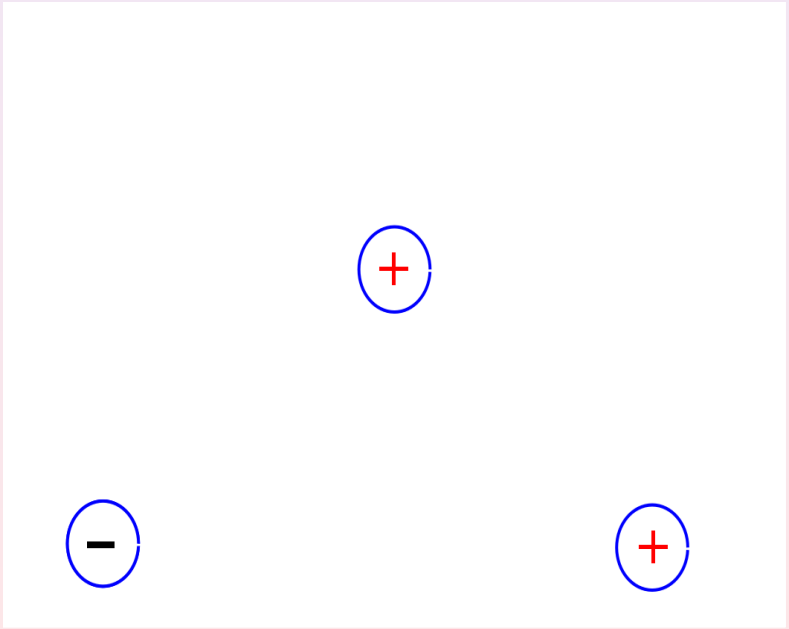
$$m_i(+) = argmax_{\mathbf{y}, y_i=+1} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$$



How to get the Ranking

- We propose to use difference of max-marginals
- $s(x_i; \mathbf{w}) = m_i(+) - m_i(-)$, where, $m_i(+)$ is the max-marginal score such that sample x_i takes label of $+1$.

$$m_i(+) = argmax_{\mathbf{y}, y_i=+1} \mathbf{w}^\top \Psi(\mathbf{x}, \mathbf{y})$$



Computational Complexity

Computational Complexity

- Max-marginals for n samples, n times Graph-Cut ?

Computational Complexity

- Max-marginals for n samples, n times Graph-Cut ?
- Dynamic Graph-Cut - Very Efficient

Presentation Outline

- We must optimize AP in order to avoid suboptimal solutions

- We must optimize AP in order to avoid suboptimal solutions
- We must use High-Order information

- We must optimize AP in order to avoid suboptimal solutions
- We must use High-Order information
- Summary table

Method/Properties	Loss	High-Order	Convex	Ranking
SVM	0-1	No	Yes	Yes
AP-SVM	AP	No	Yes	Yes
HOB-SVM	0-1	Yes	Yes	Yes

- We must optimize AP in order to avoid suboptimal solutions
- We must use High-Order information
- Summary table

Method/Properties	Loss	High-Order	Convex	Ranking
SVM	0-1	No	Yes	Yes
AP-SVM	AP	No	Yes	Yes
HOB-SVM	0-1	Yes	Yes	Yes

- No method uses High-Order information and optimizes AP in a single unified framework

Presentation Outline

HOAP-SVM

HOAP-SVM

- Uses High-Order Information

HOAP-SVM

- Uses High-Order Information
- Optimizes AP based loss function (thus improve ranking)

Problem Formulation

Problem Formulation

Similar to AP-SVM

- Single input $\mathbf{x}$, Positive Set $\mathcal{P}$, Negative Set $\mathcal{N}$
- $\phi(x_i), \forall i \in \mathcal{P}$, $\phi(x_j), \forall j \in \mathcal{N}$
- Ranking

$$R_{ij} = \begin{cases} +1, & \text{if i is better ranked than j} \\ -1, & \text{if j is better ranked than i} \end{cases}$$

- Loss function $\Delta(R, R^*) = 1 - AP(R, R^*)$

- Define score for the given ranking as

$$S(\mathbf{x}, R; \mathbf{w}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} R_{ij}(s_i(\mathbf{w}) - s_j(\mathbf{w}))$$

Encodes Ranking

- Define score for the given ranking as

$$S(\mathbf{x}, R; \mathbf{w}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} R_{ij}(s_i(\mathbf{w}) - s_j(\mathbf{w}))$$

Encodes Ranking

- Note that the scores s_i are the same that we defined in HOB-SVM, difference of max-marginals

$$s_i(\mathbf{w}) = m_i(+) - m_i(-)$$

Encodes High-Order Information

Optimization

Optimization

- Objective Function

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{8}$$

Optimization

- Objective Function

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{8}$$
$$\text{s.t.} \quad S(\mathbf{x}, R^*; \mathbf{w}) - S(\mathbf{x}, R; \mathbf{w}) \geq \Delta(R, R^*) - \xi, \forall R$$

Optimization

- Objective Function

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{8}$$
$$\text{s.t.} \quad S(\mathbf{x}, R^*; \mathbf{w}) - S(\mathbf{x}, R; \mathbf{w}) \geq \Delta(R, R^*) - \xi, \forall R$$

- Each Max-Marginal is a convex function (max over affine functions)

Optimization

- Objective Function

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi \tag{8}$$
$$\text{s.t.} \quad S(\mathbf{x}, R^*; \mathbf{w}) - S(\mathbf{x}, R; \mathbf{w}) \geq \Delta(R, R^*) - \xi, \forall R$$

- Each Max-Marginal is a convex function (max over affine functions)
- Above objective function is a difference of convex program

Difference of Convex

Difference of Convex

- DC objective function of HOAP-SVM

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi,$$

Difference of Convex

- DC objective function of HOAP-SVM

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi,$$

$$\text{s.t. } \xi \geq \frac{1}{|\mathcal{P}||\mathcal{N}|}\sum_{i\in P}\sum_{j\in N}(R^*_{ij} - R_{ij})\Big(m_i(-) + m_j(+)\Big) +$$

$$\Delta(R, R^*) -$$

Difference of Convex

- DC objective function of HOAP-SVM

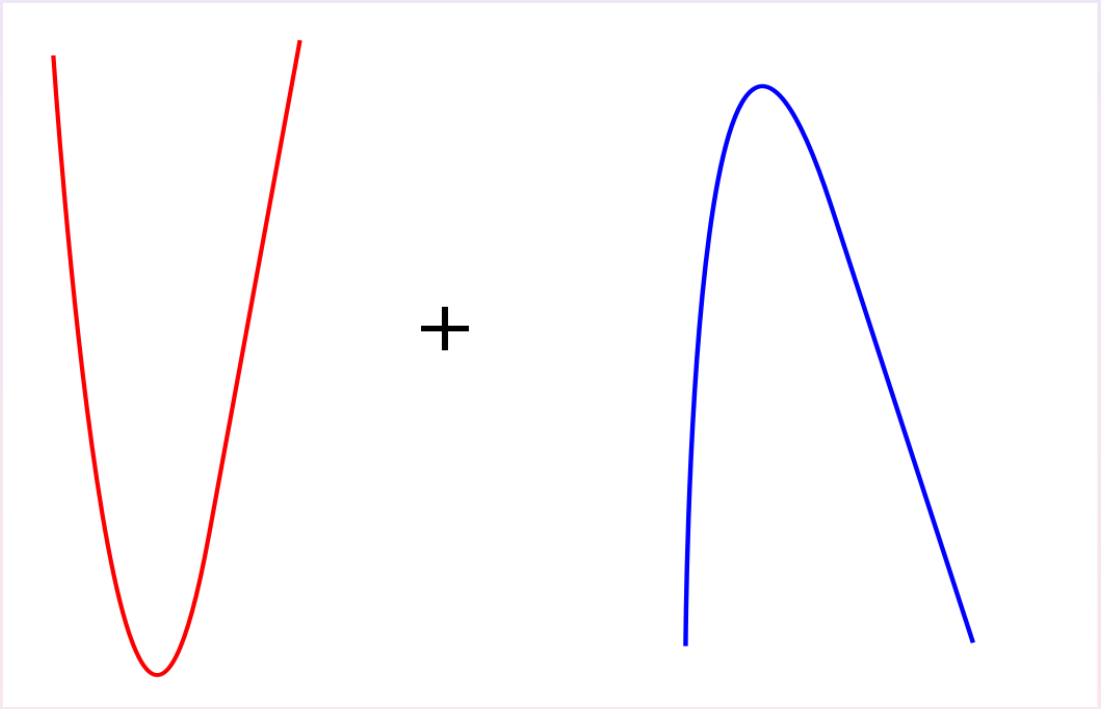
$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi, \\ \text{s.t. } \xi \quad \geq \quad & \frac{1}{|\mathcal{P}||\mathcal{N}|}\sum_{i\in P}\sum_{j\in N}(R^*_{ij} - R_{ij})\Big(m_i(-) + m_j(+)\Big) + \\ \Delta(R, R^*) \quad - \quad & \frac{1}{|\mathcal{P}||\mathcal{N}|}\sum_{i\in P}\sum_{j\in N}(R^*_{ij} - R_{ij})\Big(m_i(+) + m_j(-)\Big), \\ & \forall R, \xi \geq 0, \mathbf{w}_2 \leq 0. \end{aligned} \tag{9}$$

CCCP

Difference of convex functions can be optimized using CCCP algorithm

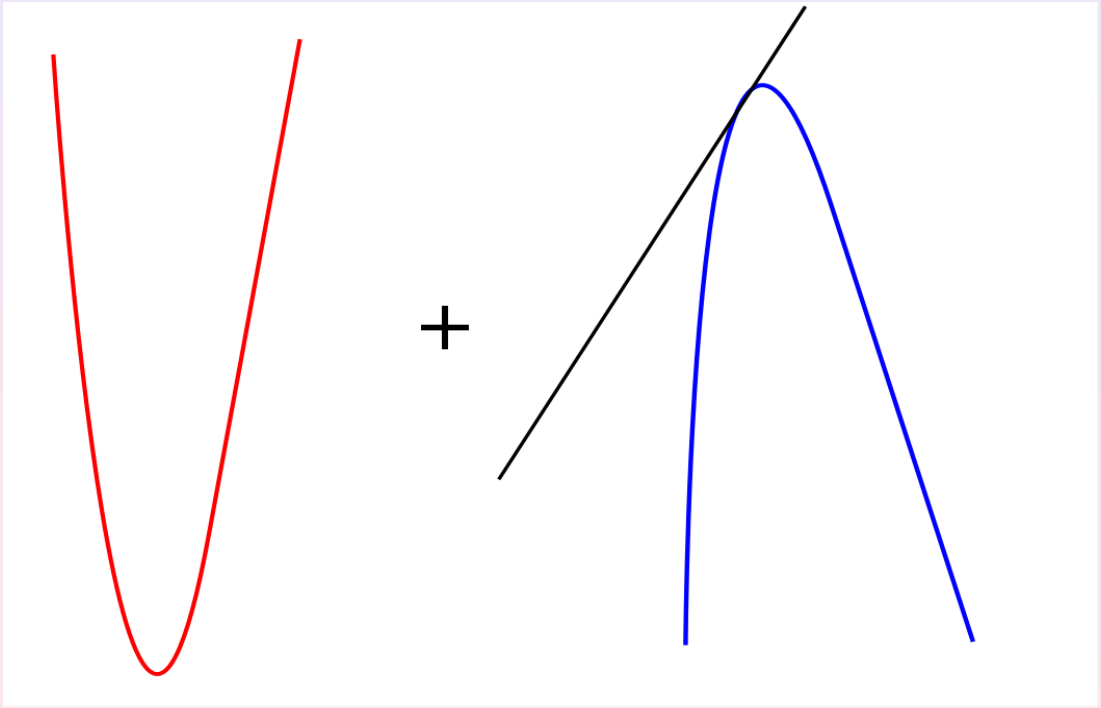
CCCP

Difference of convex functions can be optimized using CCCP algorithm



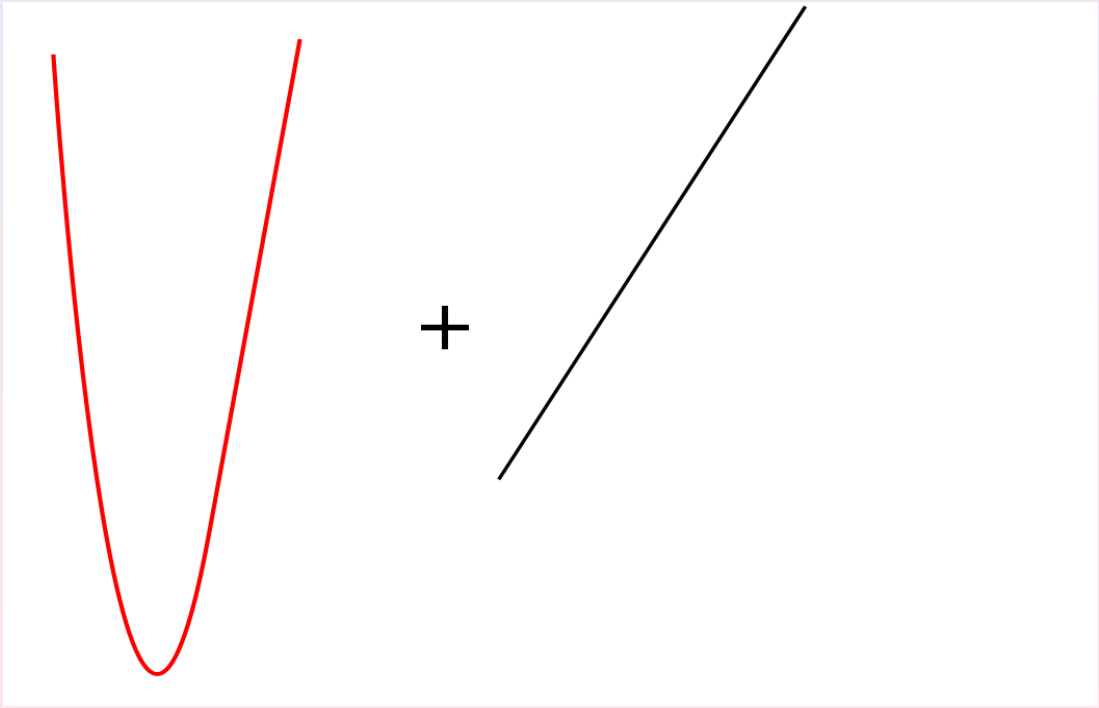
CCCP

Difference of convex functions can be optimized using CCCP algorithm



CCCP

Difference of convex functions can be optimized using CCCP algorithm



Computation

Computation

- Linearization step using Dynamic Graph-Cut - Very Efficient

Computation

- Linearization step using Dynamic Graph-Cut - Very Efficient
- Loss augmented inference using Greedy approach (Yue et.al.)

Computation

- Linearization step using Dynamic Graph-Cut - Very Efficient
- Loss augmented inference using Greedy approach (Yue et.al.)
- Ranking - sort individual scores s_i, $\mathcal{O}(nlogn)$

Presentation Outline

Action Recognition

Action Recognition

- PASCAL VOC 2011 Dataset
- 10 Action Classes

Action Recognition

- PASCAL VOC 2011 Dataset
- 10 Action Classes
- Unary Feature - POSELET and GIST concatenated
- High-Order Feature -POSELET

Action Recognition

- PASCAL VOC 2011 Dataset
- 10 Action Classes
- Unary Feature - POSELET and GIST concatenated
- High-Order Feature -POSELET
- High-Order Information

Action Recognition

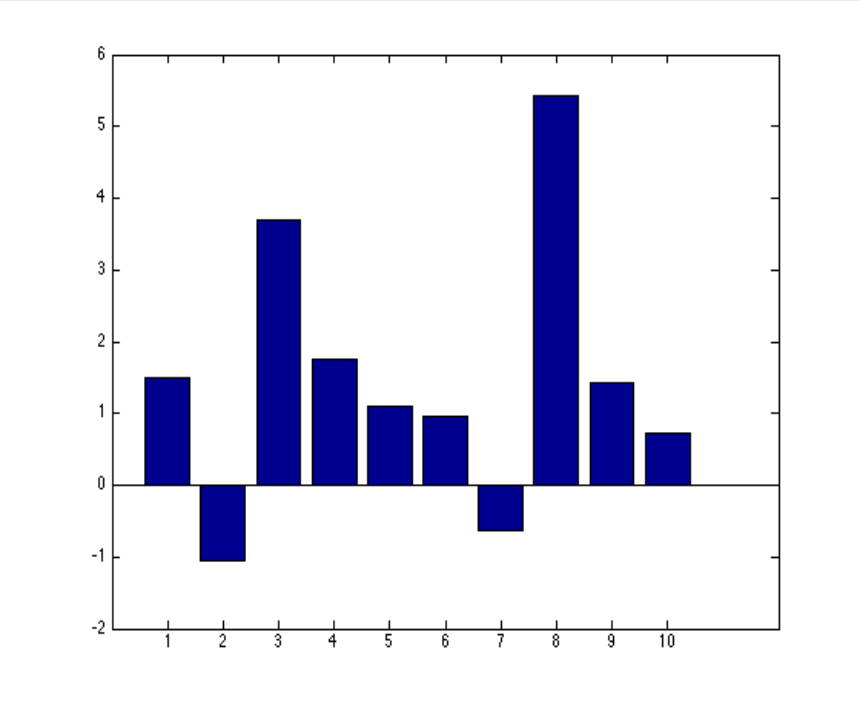
- PASCAL VOC 2011 Dataset
- 10 Action Classes
- Unary Feature - POSELET and GIST concatenated
- High-Order Feature -POSELET
- High-Order Information
  - Hypothesis: Persons in the same image are more likely to perform same action

Action Recognition

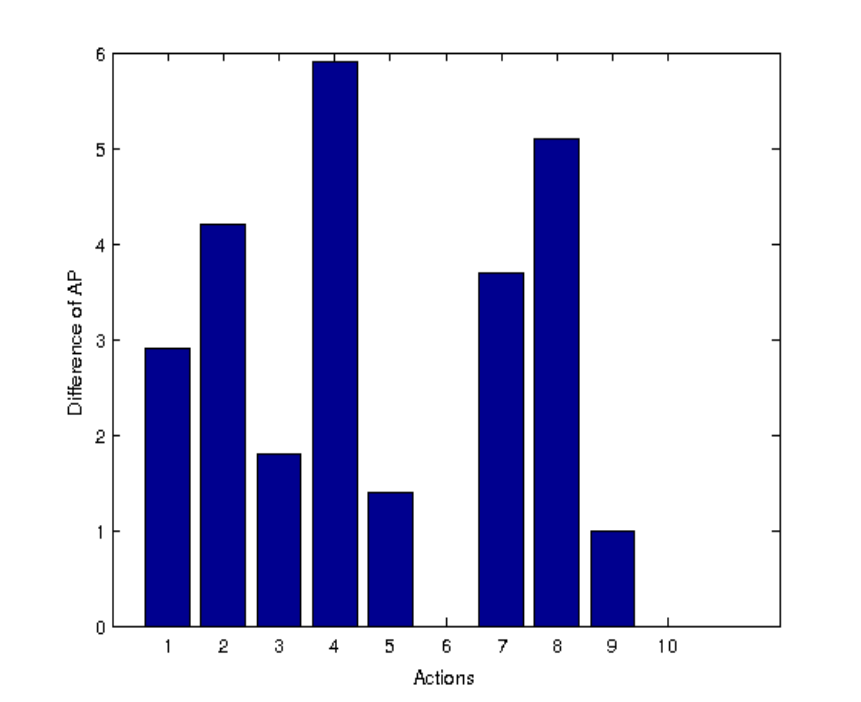
- PASCAL VOC 2011 Dataset
- 10 Action Classes
- Unary Feature - POSELET and GIST concatenated
- High-Order Feature -POSELET
- High-Order Information
  - Hypothesis: Persons in the same image are more likely to perform same action
  - Connected bounding boxes coming from the same image

AP-SVM vs SVM

Difference of AP



Trainval

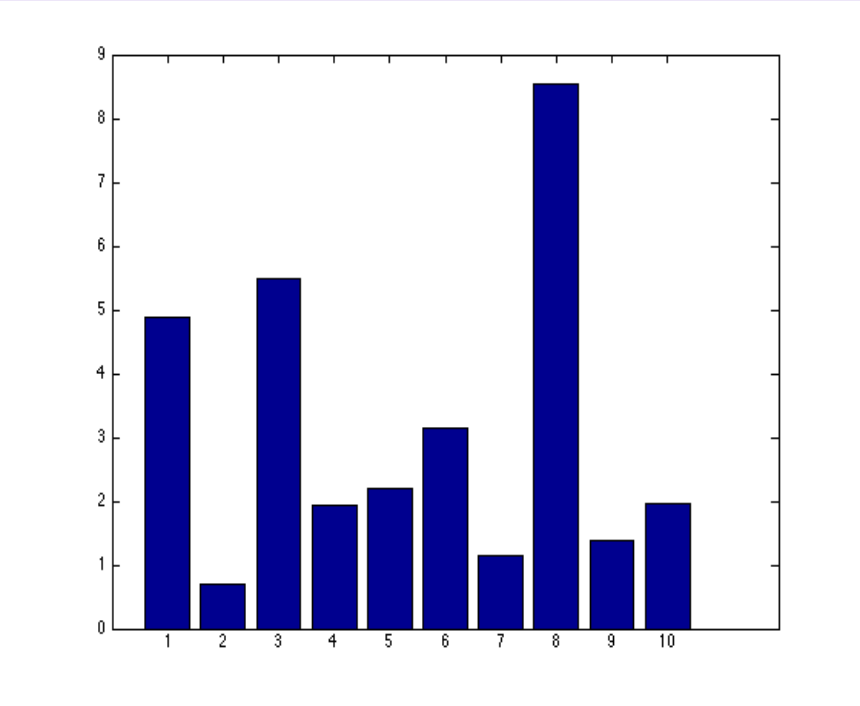


Test (Evaluation Server)

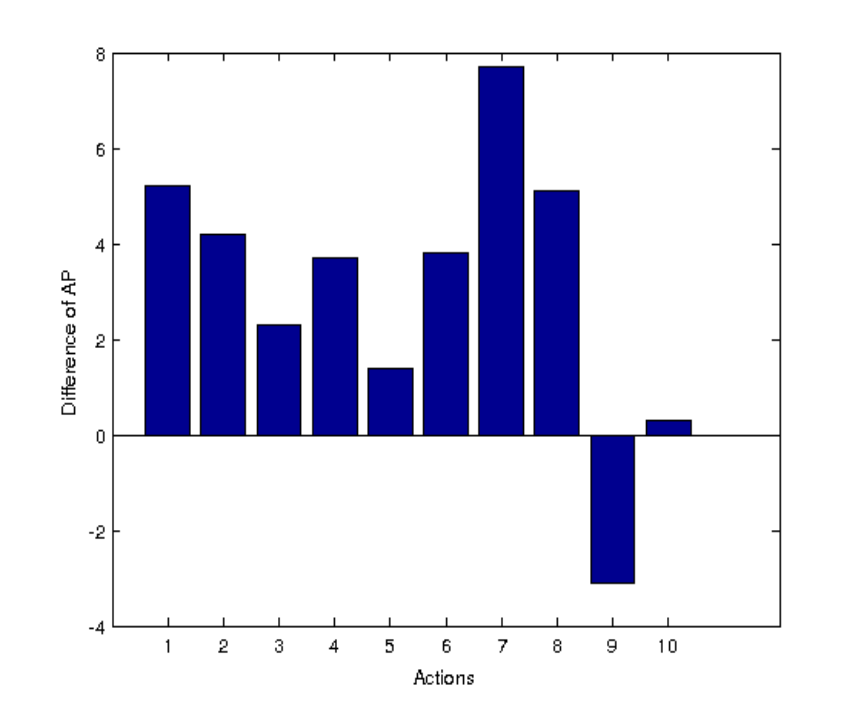
AP-SVM statistically better than SVM in 3 action classes

HOB-SVM vs SVM

Difference of AP



Trainval

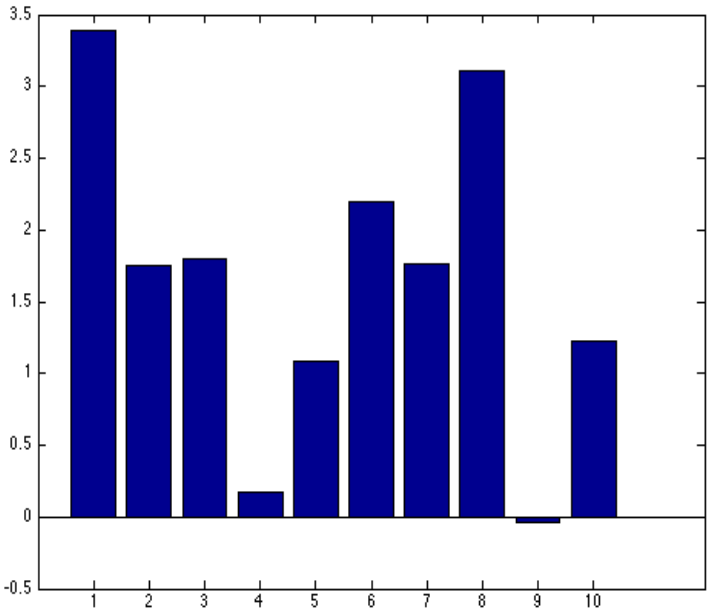


Test (Evaluation Server)

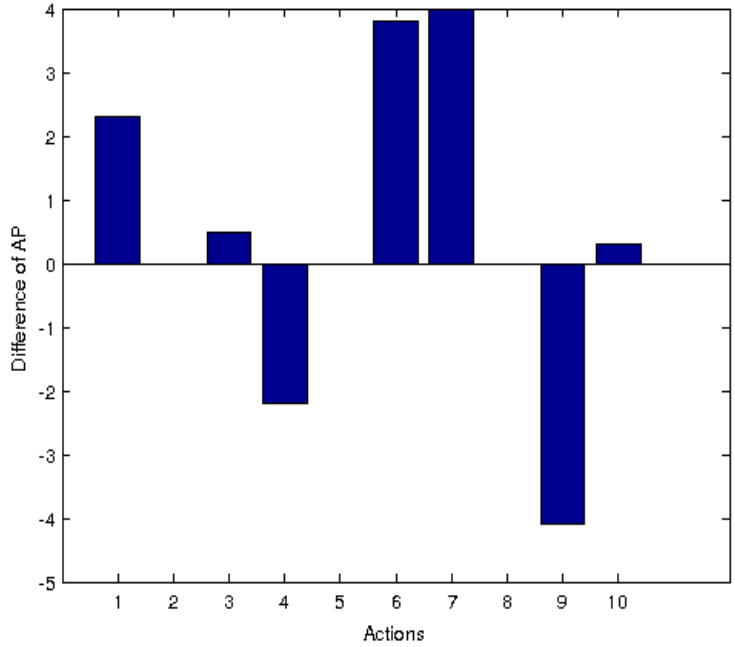
HOB-SVM statistically better than SVM in 6 action classes

HOB-SVM vs AP-SVM

Difference of AP



Trainval

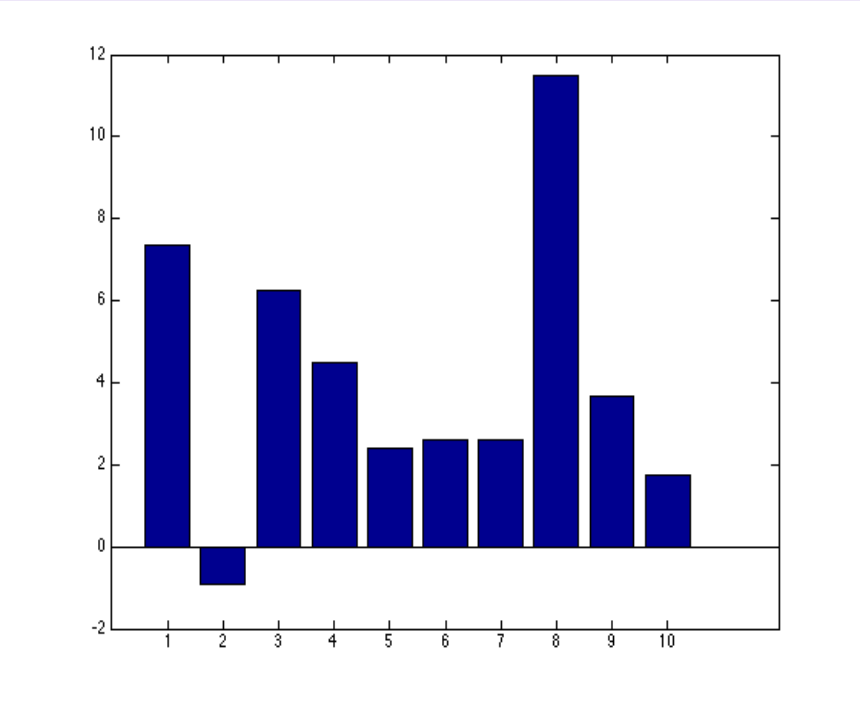


Test (Evaluation Server)

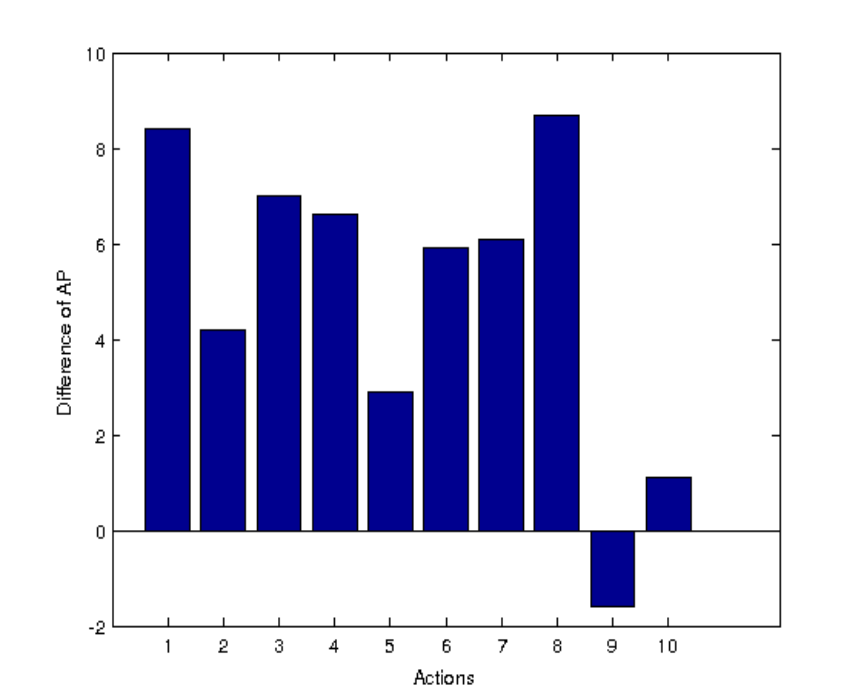
HOB-SVM is not statistically better than AP-SVM in any class

HOAP-SVM vs SVM

Difference of AP



Trainval

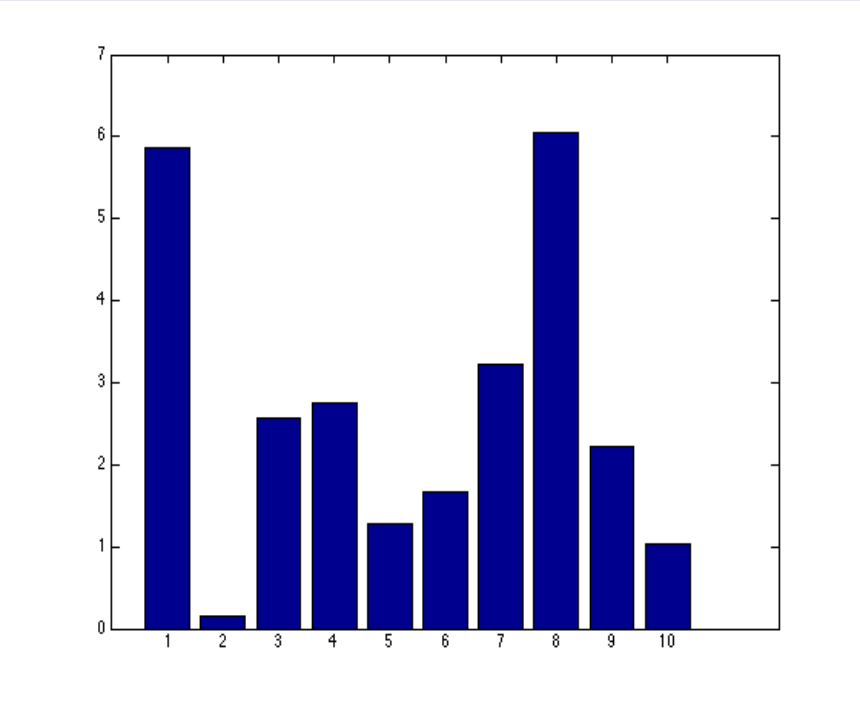


Test (Evaluation Server)

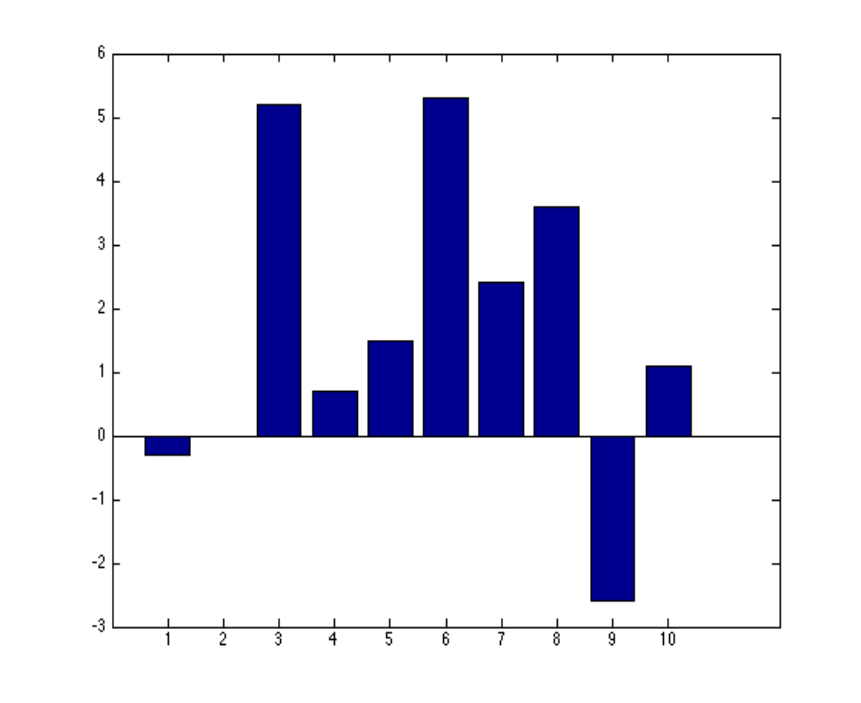
HOAP-SVM statistically better than SVM in 6 action classes

HOAP-SVM vs AP-SVM

Difference of AP



Trainval



Test (Evaluation Server)

HOAP-SVM statistically better than APSVM in 4 action classes

PASCAL VOC Results

Five fold cross validation results

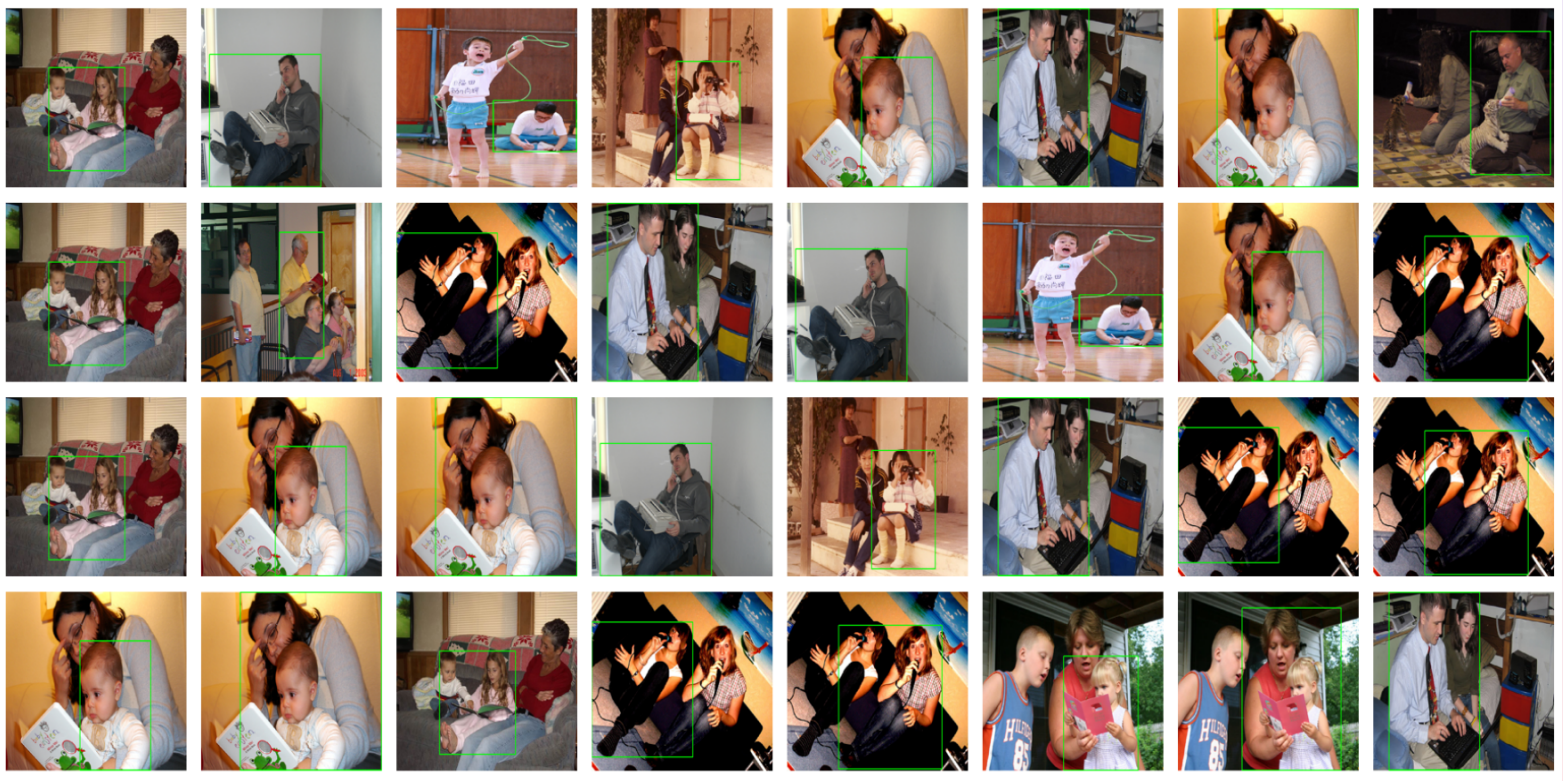
Actions/ Methods	Jump	Phone	Play inst	Read	Ride bike	Run	Take photo	Use comp	Walk	Ride horse
SVM	56.0	35.5	42.6	33.8	81.9	78.4	33.9	37.2	61.7	85.9
AP-SVM	57.5	34.4	46.3	35.5	83.0	79.3	33.3	42.7	63.1	86.6
HOB-SVM	60.9	**36.1**	48.1	35.7	84.1	**81.5**	35.1	45.8	63.0	**87.9**
HOAP-SVM	**63.4**	34.5	**48.8**	**38.3**	**84.3**	81.0	**36.5**	**48.7**	**65.3**	87.7

PASCAL Evaluation server results (Test)

Actions/ Methods	Jump	Phone	Play inst	Read	Ride bike	Run	Take photo	Use comp	Walk	Ride horse
SVM	51.1	29.7	40.5	20.6	81.1	76.7	20.0	27.7	56.7	84.2
AP-SVM	54.0	**33.8**	42.3	26.5	82.5	76.7	23.7	32.8	**57.7**	84.2
HOB-SVM	56.3	**33.8**	42.8	24.3	82.5	80.5	**27.7**	32.8	53.6	84.5
HOAP-SVM	**59.5**	**33.8**	**47.5**	27.2	**84.0**	**82.6**	26.1	36.4	55.1	**85.3**

Visualization - Usingcomputer top 6

- First row: SVM, Second row: AP-SVM
- Third row: HOB-SVM, Fourth row: HOAP-SVM



Presentation Outline

- More Applications to explore -

- More Applications to explore -
  - Likely to become smoker - based on social network

- More Applications to explore -
  - Likely to become smoker - based on social network
  - Link prediction

- More Applications to explore -
  - Likely to become smoker - based on social network
  - Link prediction
  - Medical Imaging - segmentation

- More Applications to explore -
  - Likely to become smoker - based on social network
  - Link prediction
  - Medical Imaging - segmentation
- Link with Manifold regularization . . .

Presentation Outline

http://cvn.ecp.fr/projects/ranking-highorder/

Presentation Outline

Thank You . . .