Multi-Instance Classification by Max-Margin Training of Cardinality-Based Markov Networks

by Hossein Hajimirsadeghi and Greg Mori

Presented by Denis Baručić

Biomedical Imaging Algorithms
Faculty of Electrical Engineering
Czech Technical University in Prague

October 23 , 2020

Multiple Instance Learning

- A form of supervised learning
- $\mathcal{B} = \{\mathcal{I}_1, \ldots, \mathcal{I}_m\}$ is a bag of m instances with a binary bag label $Y \in \{-1, 1\}$
- Instance $\mathcal{I}_i$ is represented by a feature vector $\mathbf{x}_i \in \mathbb{R}^D$ and has a hidden binary instance label $y_i \in \{-1, 1\}$
- Bag $\mathcal{B}$ might be described by another feature vector $\mathbf{X}$

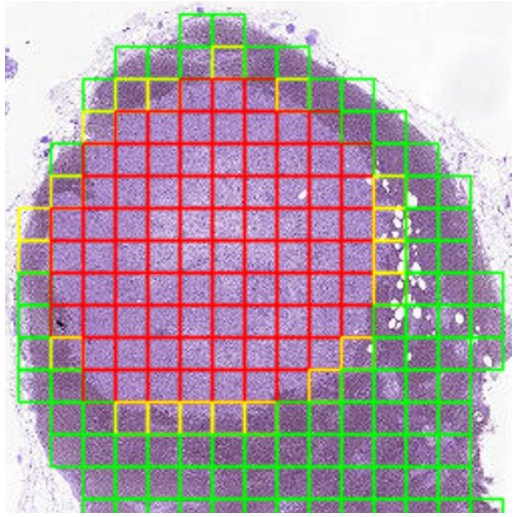
Multiple Instance Learning

- A form of supervised learning
- $\mathcal{B} = \{\mathcal{I}_1, \ldots, \mathcal{I}_m\}$ is a bag of m instances with a binary bag label $Y \in \{-1, 1\}$
- Instance $\mathcal{I}_i$ is represented by a feature vector $\mathbf{x}_i \in \mathbb{R}^D$ and has a hidden binary instance label $y_i \in \{-1, 1\}$
- Bag $\mathcal{B}$ might be described by another feature vector $\mathbf{X}$

- Assumption: $Y = \max\{y_1, \ldots, y_m\}$
- Given a training set $\mathcal{T}^N = \{(\mathcal{B}^n, Y^n) \mid n = 1, ..., N\}$, learn to predict the bag labels
  - ... and perhaps the instance labels too

Multiple Instance Learning: use cases

- Instance labels are unknown or too laborious to obtain
- Molecule prediction (Dietterich, 1997)
  - Molecule represented as a bag of feature vectors describing low-energy configurations
  - Predict if a molecule will smell "musky"
- Image classification
  - Histology image represented as a bag of image patches
  - Decide if an image contains cancer



Generalized assumption

What if we knew that a positive bag always contains at least a given portion of positive instances?

Generalized assumption

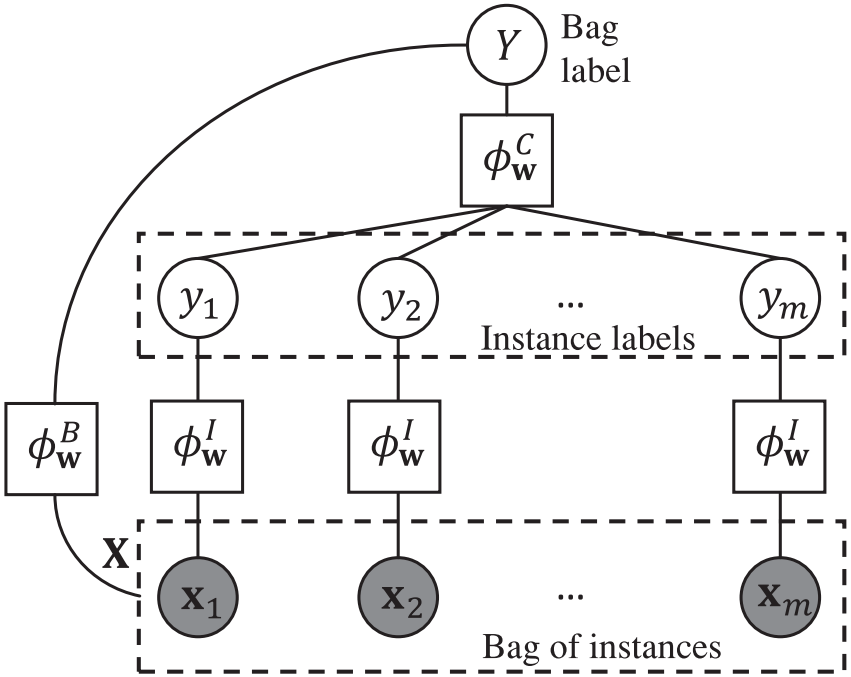
What if we knew that a positive bag always contains at least a given portion of positive instances?

The original assumption $Y = \max\{y_1, \ldots, y_n\}$ still holds but is weak.

Use the apriori information!

Proposed model

Markov network over the tuple $(\mathbf{X}, \mathbf{x} = \{\mathbf{x}_i\}_{i=1}^m, Y, \mathbf{y} = \{y_i\}_{i=1}^m)$



and the scoring function

$$f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) = \sum_i \underbrace{\phi_{\mathbf{w}}^I(\mathbf{x}_i, y_i)}_{\text{instance}} + \underbrace{\phi_{\mathbf{w}}^C(\mathbf{y}, Y)}_{\text{labels}} + \underbrace{\phi_{\mathbf{w}}^B(\mathbf{X}, Y)}_{\text{bag}}$$

Potentials

- *Instance-label*

$$\phi^I_{\mathbf{w}}(\mathbf{x}_i, y_i) = \mathbf{w}_I^T \mathbf{x}_i \, \mathbb{1}(y_i = 1) = \mathbf{w}_I^T \Psi_I(\mathbf{x}_i, y_i)$$

- *Labels*

$$\begin{aligned}\phi^C_{\mathbf{w}}(\mathbf{y}, Y) &= C_{\mathbf{w}}(m^+, m^-, Y) \\ &= C^+_{\mathbf{w}}(m^+, m^-) \, \mathbb{1}(Y = 1) + C^-_{\mathbf{w}}(m^+, m^-) \, \mathbb{1}(Y = -1)\end{aligned}$$

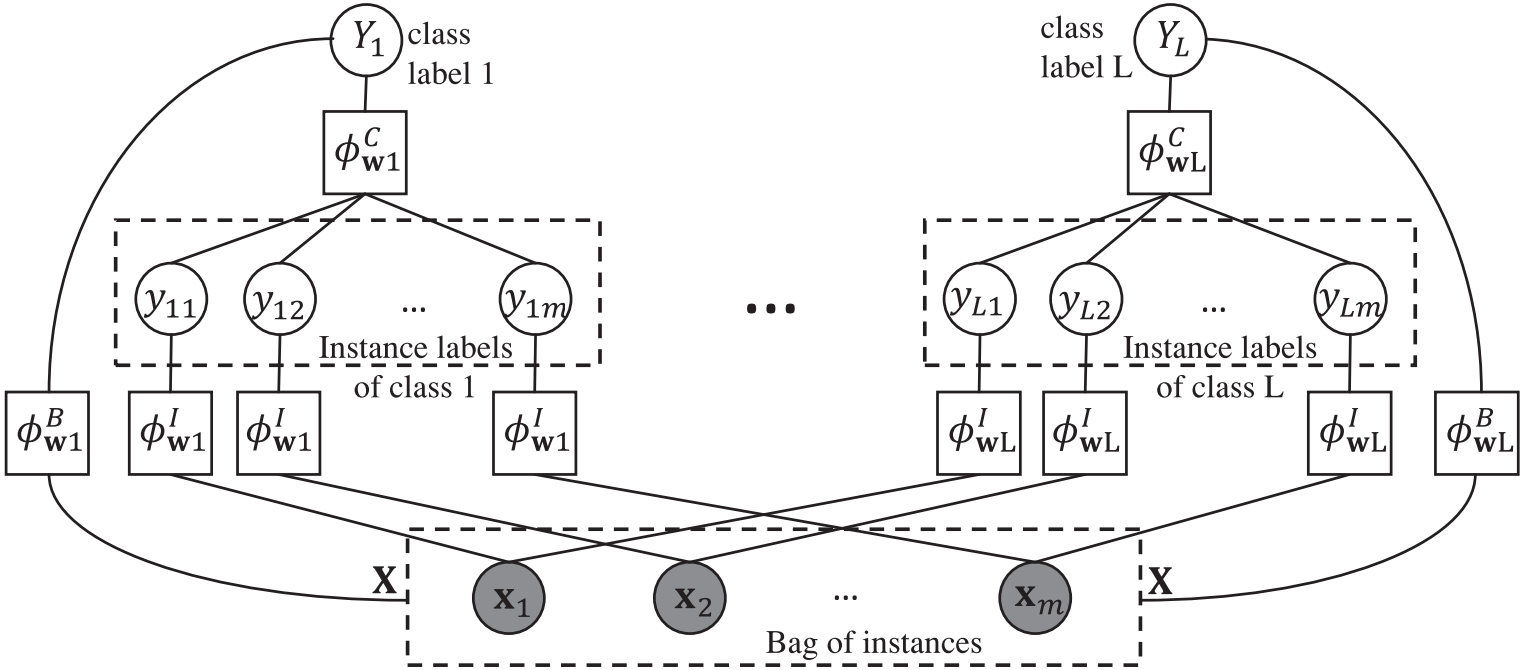
- *Bag-label*

$$\phi^B_{\mathbf{w}}(\mathbf{X}, Y) = \mathbf{w}_B^T \mathbf{X} \, \mathbb{1}(Y = 1) = \mathbf{w}_B^T \Psi_B(\mathbf{X}, Y)$$

Functions $C^+_{\mathbf{w}}$ and $C^-_{\mathbf{w}}$ will encode the assumption.
Notice: the instance-label and bag-label potentials are linear functions of $\mathbf{w}$.

Multiclass version

Now, $Y \in \{1, 2, \ldots, L\}$ is represented by a binary vector $(Y_1, Y_2, \ldots, Y_L)$, $\mathbf{y}_l$ are binary instance labels for class l, and $\mathbf{y}$ is a collection of all instance labels of all classes.



$$f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) = \sum_{l=1}^{L} \left(\sum_i \phi^I_{\mathbf{w}^l}(\mathbf{x}_i, y_{li}) + \phi^C_{\mathbf{w}^l}(\mathbf{y}_l, Y_l) + \phi^B_{\mathbf{w}^l}(\mathbf{X}, Y_l) \right)$$

Assumption encoding

Recall:

$$\phi^C_{\mathbf{w}_l}(\mathbf{y}, Y) = C^+_{\mathbf{w}_l}(m^+, m^-)\,\mathbb{1}(Y = 1) + C^-_{\mathbf{w}_l}(m^+, m^-)\,\mathbb{1}(Y = -1)$$

- The standard MIL assumption

$$C^+_{\mathbf{w}_l}(m^+_l, m^-_l) = \begin{cases} -\infty \\ w^+_{Cl} \end{cases} \qquad C^-_{\mathbf{w}_l}(m^+_l, m^-_l) = \begin{cases} w^-_{Cl} & \text{if } m^+_l = 0, \\ -\infty & \text{otherwise.} \end{cases}$$

- The generalized assumption

$$C^+_{\mathbf{w}_l}(m^+_l, m^-) = \begin{cases} -\infty \\ w^+_{Cl} \end{cases} \qquad C^-_{\mathbf{w}_l}(m^+_l, m^-) = \begin{cases} w^-_{Cl} & \text{if } 0 \leq \frac{m^+_l}{m^+_l + m^-_l} \leq \rho, \\ -\infty & \text{otherwise.} \end{cases}$$

The ratio can be learned

Divide the bag size into K equal parts, and define

$$C^{+}_{\mathbf{w}_l}(m_l^+, m - m_l^+) = \begin{cases} -\infty & \text{if } m_l^+ = 0, \\ \sum_{k=1}^{K} w_{kl}^{+} \, \mathbb{1}\left(\frac{k-1}{K} < \frac{m_l^+}{m} \leq \frac{k}{K}\right) & \text{otherwise.} \end{cases}$$

$$C^{-}_{\mathbf{w}_l}(m_l^+, m - m_l^+) = \begin{cases} \sum_{k=1}^{K} w_{kl}^{-} \, \mathbb{1}\left(\frac{k-1}{K} \leq \frac{m_l^+}{m} < \frac{k}{K}\right) & \text{if } m_l^+ = 0, \\ -\infty & \text{otherwise.} \end{cases}$$

Higher K leads to finer "resolution of ρ".

Linearity

The model is linear in $\mathbf{w}$ for any of the definitions of the label potentials $\phi^C_{\mathbf{w}_l}$, and can be expressed as

$$f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) = \mathbf{w}^T \Psi(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) + \sum_l g_C(\mathbf{y}_l, Y_l),$$

where Ψ aggregates the functions Ψ_I, Ψ_C, Ψ_B and g_C (I think) the negative infinities.

Inference

Goal: predict the bag label Y^* for a given bag $\mathcal{B}$ described by $\mathbf{X}$ and $\mathbf{x}$.

Inference

Goal: predict the bag label Y^* for a given bag $\mathcal{B}$ described by $\mathbf{X}$ and $\mathbf{x}$.

Define the scoring function for assigning the bag label Y as

$$F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y) = \max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}),$$

and then set

$$Y^* = \arg\max_{Y} F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y).$$

Inference

Goal: predict the bag label Y^* for a given bag $\mathcal{B}$ described by $\mathbf{X}$ and $\mathbf{x}$.

Define the scoring function for assigning the bag label Y as

$$F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y) = \max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}),$$

and then set

$$Y^* = \arg\max_{Y} F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y).$$

In general, evaluating $F_{\mathbf{w}}$ is NP-complete. Can we do better?

Inference decomposition

Consider $F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y)$ for a fixed $Y \in \{1, \ldots, L\}$.

Inference decomposition

Consider $F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y)$ for a fixed $Y \in \{1, \ldots, L\}$.
Then

$$
\begin{aligned}
F_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y) &= \max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}, \mathbf{x}, Y, \mathbf{y}) \\
&= \max_{\mathbf{y}} \sum_{l=1}^{L} \left(\sum_{i} \phi^{I}_{\mathbf{w}_l}(\mathbf{x}_i, y_{li}) + \phi^{C}_{\mathbf{w}_l}(\mathbf{y}_l, Y_l) + \phi^{B}_{\mathbf{w}_l}(\mathbf{X}, Y_l) \right) \\
&= \max_{\mathbf{y}} \sum_{l=1}^{L} \left(\sum_{i} \phi^{I}_{\mathbf{w}_l}(\mathbf{x}_i, y_{li}) + C_{\mathbf{w}_l}(m_l^{+}, m_l^{-}, Y) \right) \\
&= \sum_{l=1}^{L} \max_{\mathbf{y}_l} \left(\sum_{i} \phi^{I}_{\mathbf{w}_l}(\mathbf{x}_i, y_{li}) + C_{\mathbf{w}_l}(m_l^{+}, m_l^{-}, Y) \right),
\end{aligned}
$$

which decomposes into L independent optimization tasks

$$
\max_{\mathbf{y}_l} \left(\sum_{i} \phi^{I}_{\mathbf{w}_l}(\mathbf{x}_i, y_{li}) + C_{\mathbf{w}_l}(m_l^{+}, m_l^{-}, Y) \right), l = 1, \ldots, L.
$$

Inference algorithm

The problem

$$\max_{\mathbf{y}_l} \left(\sum_i \phi^I_{\mathbf{w}_l}(\mathbf{x}_i, y_{li}) + C_{\mathbf{w}_l}(m_l^+, m_l^-, Y) \right).$$

can be solved exactly for binary instance labels y_{li}.

1. Sort $W^l = \left(\phi^I_{\mathbf{w}_l}(\mathbf{x}_i, +1) - \phi^I_{\mathbf{w}_l}(\mathbf{x}_i, -1)\right)_{i=1}^m$ in decreasing order
2. For $k = 0, \ldots, m$, compute $s_k^l = \sum_{j=1}^k W_j^l + C_{\mathbf{w}_l}(k, m-k, Y_l)$
3. Find $k_l^* \in \arg\max_k s_k^l$
4. Set positive labels for top k_l^* instances from W^l, negative for the rest

Doing so for L classes takes $O(Lm \log m)$ time.

Learning

Consider the training set $\mathcal{T}^N = \{(\mathbf{X}^n, \mathbf{x}^n, Y^n) \mid n = 1, ..., N\}$.
The goal is to learn the parameters $\mathbf{w}$.
The learning is formulated as minimizing the hinge loss:

$$\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathcal{L}^n - \mathcal{R}^n) + \frac{\lambda}{2}\|\mathbf{w}\|^2,$$

$$\text{where } \mathcal{L}^n = \max_{Y} \max_{\mathbf{y}} \left(\mathbb{1}(Y \neq Y^n) + f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y, \mathbf{y}) \right),$$

$$\mathcal{R}^n = \max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y^n, \mathbf{y}).$$

Notice the hinge loss:

$$\mathcal{L}^n - \mathcal{R}^n =$$

$$\max\left(0, 1 + \max_{Y \neq Y^n} F_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y, \mathbf{y}) - F_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y^n, y)\right)$$

Learning algorithm

Non-convex Regularized Bundle Method (NRBM), a cutting plane algorithm, employed for solving the optimization problem.

NRBM solves problems in the form

$$\min_{\mathbf{w}} f(\mathbf{w})$$
$$\text{where } f(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + R(\mathbf{w}),$$

for a (not necessarily convex or smooth) risk $R(\mathbf{w})$. The method needs $\partial_{\mathbf{w}} f(\mathbf{w})$.

Here, we have

$$R(\mathbf{w}) = \sum_{n=1}^{N} \left(\mathcal{L}^n - \mathcal{R}^n\right),$$

and thus, we need to compute $\partial_{\mathbf{w}} \mathcal{L}^n$ and $\partial_{\mathbf{w}} \mathcal{R}^n$.

Learning: subgradient computation

Recall:

$$\mathcal{L}^n = \max_Y \max_{\mathbf{y}} \left(\mathbb{1}(Y \neq Y^n) + f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y, \mathbf{y}) \right),$$
$$\mathcal{R}^n = \max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y^n, \mathbf{y})$$

Computing $\partial_{\mathbf{w}} \mathcal{L}^n$ amounts to solving the inference problem

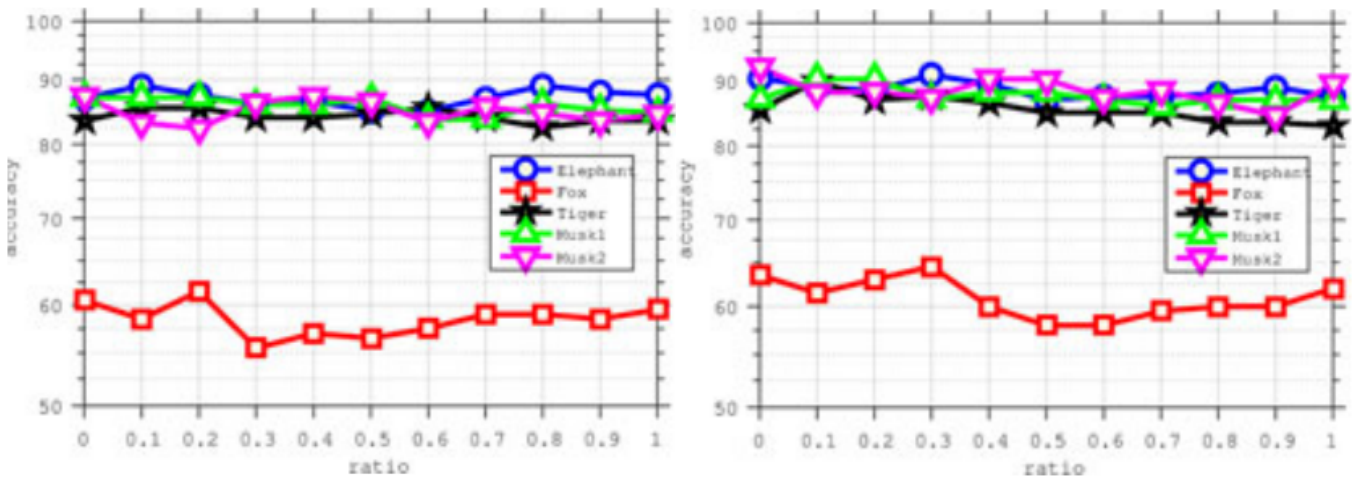
$$(Y^*, \mathbf{y}^*) \in \arg\max_{Y,\mathbf{y}} \left(\mathbb{1}(Y, Y^n) + f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y, \mathbf{y}) \right)$$

and putting $\partial_{\mathbf{w}} \mathcal{L}^n = \Psi(\mathbf{X}^n, \mathbf{x}^n, Y^*, \mathbf{y}^*)$.

Similarly, $\partial_{\mathbf{w}} \mathcal{R}^n = \Psi(\mathbf{X}^n, \mathbf{x}^n, Y^n, \mathbf{y}^*)$, where

$$\mathbf{y}^* \in \arg\max_{\mathbf{y}} f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y^n, \mathbf{y}).$$

Experiments: standard datasets



Results on the standard datasets for different ratios ρ without (left) and with (right) bag features.
Prediction score of the MI-Kernel method used as the bag features.

Observation: works *approximately* the same for any ρ

Experiments: standard datasets – comparison

Method	Elephant	Fox	Tiger	Musk1	Musk2
MIMN	*89*	**64**	*86*	87	**92**
RMIMN ($\rho = 0.5$)	87	59	85	88	**92**
GMIMN ($K = 5$)	**90**	63	**89**	89	**92**
mi-SVM [11]	82	58	79	87	84
MI-SVM [11]	81	59	84	78	84
MI-Kernel [27]	84	60	84	88	*89*
γ-rule SVM [44]	84	63	81	88	85
SetMaxRBM^{XOR} [23]	88	60	83	84	84
MIRealBoost [19]	83	63	73	**91**	77
MIForest [15]	84	**64**	82	85	82
SVR-SVM [45]	85	63	80	88	85
MIGraph [30]	85	61	82	*90*	90
miGraph [30]	87	62	*86*	*90*	90
MILES [14]	81	62	80	88	83
AW-SVM [17]	82	**64**	83	86	84
AL-SVM [17]	79	63	78	86	83
EM-DD [10]	78	56	72	85	85

Bold = best, italic = second best.

Observation: always among the best, better than MI-Kernel.

Experiments: cyclist helmet

- 24 cyclist tracks (bags), 12 wearing helmets and 12 not
- head location estimated
- region around the head described using texton histograms (instances)
- some instances in a positive bag might be negative due to tracking imperfections (not suitable for standard learning)
- positive tracks very likely to have more than one positive instance (the standard MIL assumption too weak)
- goal is to recognize if a cyclist wears a helmet

Experiments: cyclist helmet – comparison

Method	Accuracy %
SVM-AtLeastOne	58.33
SVM-Majority	79.17
mi-SVM	62.50
MIMN	58.33
RMIMN ($\rho = 0.5$)	91.67
GMIMN ($K = 5$)	87.50

SVM-AtLeastOne and SVM-Majority represent supervised learning

Deep learning

Can we employ deep learning in combination with this model?

- Use a deep net as the instance-label potentials $\phi^I_{\mathbf{w}}$?
  - How to train such a model?
  - Some iterative algorithm (alternate between training the net and the Markov model)?
  - Or switch from the risk minimization to MLE and do gradient descent together with the net?
- Use a deep net for feature extraction and pass the extracted features to the Markov model
  - Train the network before the Markov model
  - What network? What should be its goal?

Conclusion

- Besides image segmentation, Markov networks can tackle other problems too
- The authors managed to include the generalized MIL assumption into the model in a very explicit and transparent way
- Although the model looks complex, there is an efficient inference algorithm (thanks to the binary labels)
- There is a non-convex version for cutting planes