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 $x_i \in \mathbb{R}^D$ and has a hidden binary instance label $y_i \in \{-1, 1\}$
- Bag $\mathcal{B}$ might be described by another feature vector $X$
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 $x_i \in \mathbb{R}^D$ and has a hidden binary instance label $y_i \in \{-1, 1\}$
- Bag $\mathcal{B}$ might be described by another feature vector $X$

- Assumption: $Y = \max \{y_1, \ldots, y_m\}$
- Given a training set $\mathcal{T}^N = \{(\mathcal{B}^n, Y^n) \mid n = 1, \ldots, N\}$, 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 \((X, x = \{x_i\}_{i=1}^m, Y, y = \{y_i\}_{i=1}^m)\)

and the scoring function

\[
f_w(X, x, Y, y) = \sum_i \phi^I_w(x_i, y_i) + \phi^C_w(Y, Y) + \phi^B_w(X, Y)
\]

instance \hspace{2cm} labels \hspace{2cm} bag
Potentials

- **Instance-label**

\[
\phi_w^I(x_i, y_i) = w_I^T x_i \mathbb{1}(y_i = 1) = w_I^T \Psi_I(x_i, y_i)
\]

- **Labels**

\[
\phi_w^C(y, Y) = C_w(m^+, m^-, Y) = C_w^+(m^+, m^-) \mathbb{1}(Y = 1) + C_w^-(m^+, m^-) \mathbb{1}(Y = -1)
\]

- **Bag-label**

\[
\phi_w^B(X, Y) = w_B^T X \mathbb{1}(Y = 1) = w_B^T \Psi_B(X, Y)
\]

Functions \(C_w^+\) and \(C_w^-\) will encode the assumption. Notice: the instance-label and bag-label potentials are linear functions of \(w\).
Multiclass version

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

\[
f_w(X, x, Y, y) = \sum_{l=1}^{L} \left( \sum_i \phi^{I}_{wl}(x_i, y_{li}) + \phi^{C}_{wl}(y_l, Y_l) + \phi^{B}_{wl}(X, Y_l) \right)
\]
Assumption encoding

Recall:

$$\phi_{wl}^C(y, Y) = C_{wl}^+(m^+, m^-) \mathbf{1}(Y = 1) + C_{wl}^-(m^+, m^-) \mathbf{1}(Y = -1)$$

- The standard MIL assumption

$$C_{wl}^+(m_i^+, m_i^-) = \begin{cases} -\infty & \text{if } m_i^+ = 0, \\ w_{Cl}^+ & \text{otherwise.} \end{cases} \quad C_{wl}^-(m_i^+, m_i^-) = \begin{cases} w_{Cl}^- & \text{if } 0 \leq \frac{m_i^+}{m_i^+ + m_i^-} \leq \rho, \\ -\infty & \text{otherwise.} \end{cases}$$

- The generalized assumption
The ratio can be learned

Divide the bag size into $K$ equal parts, and define

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

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

Higher $K$ leads to finer “resolution of $\rho$”.
The model is linear in $w$ for any of the definitions of the label potentials $\phi^C_{wl}$, and can be expressed as

$$f_w(X, x, Y, y) = w^T \Psi(X, x, Y, y) + \sum_l g_C(y_l, Y_l),$$

where $\Psi$ aggregates the functions $\Psi_I, \Psi_C, \Psi_B$ and $g_C$ (I think) the negative infinities.
Inference

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

Define the scoring function for assigning the bag label $Y$ as 

$$F_w(X, x, Y) = \max_y f_w(X, x, Y, y),$$

and then set $Y^* = \arg \max Y F_w(X, x, Y)$. 

In general, evaluating $F_w$ is NP-complete. Can we do better?
Inference

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

Define the scoring function for assigning the bag label $Y$ as

$$F_w(X, x, Y) = \max_y f_w(X, x, Y, y),$$

and then set

$$Y^* = \arg \max_Y F_w(X, x, Y).$$
Goal: predict the bag label \( Y^* \) for a given bag \( B \) described by \( X \) and \( x \).

Define the scoring function for assigning the bag label \( Y \) as

\[
F_w(X, x, Y) = \max_y f_w(X, x, Y, y),
\]

and then set

\[
Y^* = \arg \max_Y F_w(X, x, Y).
\]

In general, evaluating \( F_w \) is NP-complete. Can we do better?
Inference decomposition

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

\[ F_w(X, x, Y) = \max_y f_w(X, x, Y, y) = \max_y L \sum_{l=1} f_{wl}(x_i, y_{li}) + c_{wl}(m+l,m-l,Y_l) \]

which decomposes into $L$ independent optimization tasks

\[ \max_{y_l} \sum_{i} f_{wl}(x_i, y_{li}) + c_{wl}(m+l,m-l,Y_l), l=1,\ldots,L. \]
Inference decomposition

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

$$F_w(X, x, Y) = \max_y f_w(X, x, Y, y)$$

$$= \max_y \sum_{l=1}^{L} \left( \sum_i \phi^I_{wl}(x_i, y_{li}) + \phi^C_{wl}(y_l, Y_l) + \phi^B_{wl}(X, Y_l) \right)$$

$$= \max_y \sum_{l=1}^{L} \left( \sum_i \phi^I_{wl}(x_i, y_{li}) + C_{wl}(m^+_l, m^-_l, Y) \right)$$

$$= \sum_{l=1}^{L} \max_{y_l} \left( \sum_i \phi^I_{wl}(x_i, y_{li}) + C_{wl}(m^+_l, m^-_l, Y) \right),$$

which decomposes into $L$ independent optimization tasks

$$\max_{y_l} \left( \sum_i \phi^I_{wl}(x_i, y_{li}) + C_{wl}(m^+_l, m^-_l, Y) \right), l = 1, \ldots, L.$$
Inference algorithm

The problem

$$\max_{y_l} \left( \sum_i \phi_{wl}^I (x_i, y_{li}) + C_{wl} (m_i^+, m_i^-, Y) \right).$$

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

1. Sort $W^l = (\phi_{wl}^I (x_i, +1) - \phi_{wl}^I (x_i, -1))_{i=1}^m$ in decreasing order

2. For $k = 0, \ldots, m$, compute
   $$s^l_k = \sum_{j=1}^k W^l_j + C_{wl} (k, m - k, Y_l)$$

3. Find $k^*_l \in \arg \max_k s^l_k$

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

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

where $\mathcal{L}^n = \max_Y \max_y (1(Y \neq Y^n) + f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y, y))$, $\mathcal{R}^n = \max_y f_{\mathbf{w}}(\mathbf{X}^n, \mathbf{x}^n, Y^n, y)$.

Notice the hinge loss:

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

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

NRBM solves problems in the form

$$\min_w f(w)$$

where

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

for a (not necessarily convex or smooth) risk $R(w)$. The method needs $\partial_w f(w)$.

Here, we have

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

and thus, we need to compute $\partial_w \mathcal{L}^n$ and $\partial_w \mathcal{R}^n$. 
Learning: subgradient computation

Recall:

\[ \mathcal{L}^n = \max_Y \max_y (1(Y \neq Y^n) + f_w(X^n, x^n, Y, y)), \]
\[ \mathcal{R}^n = \max_y f_w(X^n, x^n, Y^n, y) \]

Computing \( \partial_w \mathcal{L}^n \) amounts to solving the inference problem

\[ (Y^*, y^*) \in \arg \max_{Y, y} (1(Y, Y^n) + f_w(X^n, x^n, Y, y)) \]

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

Similarly, \( \partial_w \mathcal{R}^n = \Psi(X^n, x^n, Y^n, y^*) \), where

\[ y^* \in \arg \max_y f_w(X^n, x^n, Y^n, y). \]
Experiments: standard datasets

Results on the standard datasets for different ratios $\rho$ 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 $\rho$
Experiments: standard datasets – comparison

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

**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

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM-AtLeastOne</td>
<td>58.33</td>
</tr>
<tr>
<td>SVM-Majority</td>
<td>79.17</td>
</tr>
<tr>
<td>mi-SVM</td>
<td>62.50</td>
</tr>
<tr>
<td>MIMN</td>
<td>58.33</td>
</tr>
<tr>
<td>RMIMN ($\rho = 0.5$)</td>
<td>91.67</td>
</tr>
<tr>
<td>GMIMN ($K = 5$)</td>
<td>87.50</td>
</tr>
</tbody>
</table>

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 $\phi^I_w$?
  - 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.