Bayesian multiple instance learning: automatic feature selection and inductive transfer

Vikas Chandrakant Raykar
(joint with Balaji Krishnapuram, Jinbo Bi, Murat Dundar, R. Bharat Rao)

Siemens Medical Solutions Inc., USA

July 8, 2008
Outline of the talk

1. Multiple Instance Learning

2. Proposed algorithm
   - Training Data
   - Classifier form
   - Model
   - Estimator
   - Regularization
   - Optimization

3. Feature Selection

4. Experiments

5. Multi-task Learning
Outline of the talk

1. Multiple Instance Learning

2. Proposed algorithm
   - Training Data
   - Classifier form
   - Model
   - Estimator
   - Regularization
   - Optimization

3. Feature Selection

4. Experiments

5. Multi-task Learning
Binary Classification
Predict whether an example belongs to class '1' or class '0'

Computer Aided Diagnosis
Given a region in a mammogram predict whether it is cancer(1) or not(0).
Binary Classification
Predict whether an example belongs to class '1' or class '0'

Computer Aided Diagnosis
Given a region in a mammogram predict whether it is cancer(1) or not(0).

Text Categorization
Given a text predict whether it pertains to a given topic(1) or not(0).
Binary Classification
Predict whether an example belongs to class '1' or class '0'

Computer Aided Diagnosis
Given a region in a mammogram predict whether it is cancer(1) or not(0).

Text Categorization
Given a text predict whether it pertains to a given topic(1) or not(0).

Binary Classifier
Given a feature vector $x \in \mathbb{R}^d$ predict the class label $y \in \{1, 0\}$. 
Linear Binary Classifier

Given a feature vector $x \in \mathbb{R}^d$ and a weight vector $w \in \mathbb{R}^d$
Given a feature vector $x \in \mathbb{R}^d$ and a weight vector $w \in \mathbb{R}^d$

$$y = \begin{cases} 
1 & \text{if } w^T x > \theta \\
0 & \text{if } w^T x < \theta
\end{cases}$$
Given a feature vector $x \in \mathbb{R}^d$ and a weight vector $w \in \mathbb{R}^d$

$$y = \begin{cases} 
1 & \text{if } w^T x > \theta \\
0 & \text{if } w^T x < \theta 
\end{cases}.$$ 

- The *threshold* $\theta$ determines the operating point of the classifier.
- The ROC curve is obtained as $\theta$ is swept from $-\infty$ to $\infty$. 
Given a feature vector $x \in \mathbb{R}^d$ and a weight vector $w \in \mathbb{R}^d$

$$y = \begin{cases} 
1 & \text{if } w^T x > \theta \\
0 & \text{if } w^T x < \theta 
\end{cases}.$$

- The threshold $\theta$ determines the operating point of the classifier.
- The ROC curve is obtained as $\theta$ is swept from $-\infty$ to $\infty$.

Training/Learning a classifier implies

- Given training data $\mathcal{D}$ consisting of $N$ examples $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$
- Choose the weight vector $w$. 
Labels for the training data

**Single Instance Learning**

*every example* $x_i$ *has a label* $y_i \in \{0, 1\}$
Labels for the training data

**Single Instance Learning**

Every example $x_i$ has a label $y_i \in \{0, 1\}$

**Multiple Instance Learning**

A group of examples (bag) $x_i = \{x_{ij} \in \mathbb{R}^d\}_{j=1}^{K_i}$ share a common label
Single 'vs' Multiple Instance Learning

Single Instance Learning

Multiple Instance Learning
MIL applications

A natural framework for many applications and often found to be superior than a conventional supervised learning approach.

- Drug Activity Prediction.
- Face Detection.
- Stock Selection
- Content based image retrieval.
- Text Classification.
- Protein Family Modeling.
- **Computer Aided Diagnosis.**
Computer Aided Diagnosis as a MIL problem

Digital Mammography
Computer Aided Diagnosis as a MIL problem

Pulmonary Embolism Detection
Our notion of Bags

Bag

A bag contains many instances.
All the instances in a bag share the same label.
Our notion of Bags

**Bag**
A **bag** contains many instances. All the instances in a bag share the same label.

**Positive Bag**
A bag is labeled positive if it contains **at least** one positive instance.

**For a radiologist**
A lesion is detected if at least one of the candidate which overlaps with it is detected.
Our notion of Bags

**Bag**

A *bag* contains many instances.
All the instances in a bag share the same label.

**Positive Bag**

A bag is labeled positive if it contains *at least* one positive instance.

**For a radiologist**

A lesion is detected if at least one of the candidate which overlaps with it is detected.

**Negative Bag**

A negative bag means that *all* instances in the bag are negative.
MIL Illustration
Single instance Learning 'vs' Multiple instance learning
Outline of the talk

1. Multiple Instance Learning

2. Proposed algorithm
   - Training Data
   - Classifier form
   - Model
   - Estimator
   - Regularization
   - Optimization

3. Feature Selection

4. Experiments

5. Multi-task Learning
Proposed algorithm

Key features

**MIRVM—Multiple Instance Relevance Vector Machine**

- Logistic Regression classifier which handles MIL scenario.
- Joint feature selection and classifier learning in a Bayesian paradigm.
- Extension to multi-task learning.
- Very fast.
- Easy to use. No tuning parameters.
Training Data
Consists of $N$ bags

Notation
- We represent an instance as a feature vector $x \in \mathbb{R}^d$. 
Training Data
Consists of $N$ bags

Notation
- We represent an instance as a feature vector $x \in \mathbb{R}^d$.
- A bag which contains $K$ instances is denoted by boldface $x = \{x_j \in \mathbb{R}^d\}_{j=1}^K$. 
Training Data
Consists of $N$ bags

Notation

- We represent an instance as a feature vector $x \in \mathbb{R}^d$.
- A bag which contains $K$ instances is denoted by boldface
  $\mathbf{x} = \{x_j \in \mathbb{R}^d\}_{j=1}^K$.
- The label of a bag is denoted by $y \in \{0, 1\}$. 
Training Data
Consists of $N$ bags

Notation
- We represent an instance as a feature vector $x \in \mathbb{R}^d$.
- A bag which contains $K$ instances is denoted by boldface $x = \{x_j \in \mathbb{R}^d\}_{j=1}^K$.
- The label of a bag is denoted by $y \in \{0, 1\}$.

Training Data
The training data $\mathcal{D}$ consists of $N$ bags $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, where
- $x_i = \{x_{ij} \in \mathbb{R}^d\}_{j=1}^{K_i}$ is a bag containing $K_i$ instances
- and share the same label $y_i \in \{0, 1\}$. 
Classifier form
We consider linear classifiers

**Linear Binary Classifier**
Acts on a given instance  \( f_w(x) = w^T x \)
Classifier form

We consider linear classifiers

**Linear Binary Classifier**

Acts on a given instance \( f_w(x) = w^T x \)

\[
y = \begin{cases} 
1 & \text{if } w^T x > \theta \\
0 & \text{if } w^T x < \theta
\end{cases}
\]
Link function

The probability for the positive class is modeled as a **logistic sigmoid** acting on the linear classifier $f_w$, *i.e.*,

$$p(y = 1|x) = \sigma(w^\top x),$$

where $\sigma(z) = 1/(1 + e^{-z})$.

We modify this for the multiple instance learning scenario.
Multiple Instance Model

Logistic regression

**Positive Bag**

A bag is labeled positive if it contains at least one positive instance.

\[
p(y = 1 | x) = 1 - p(\text{all instances are negative})
\]

\[
= 1 - \prod_{j=1}^{K} [1 - p(y = +1 | x_j)]
= 1 - \prod_{j=1}^{K} \left[ 1 - \sigma(w^\top x_j) \right],
\]

where the bag \( x = \{x_j\}_{j=1}^{K} \) contains \( K \) examples.
Multiple Instance Model
Logistic regression

**Positive Bag**
A bag is labeled positive if it contains at least one positive instance.

\[ p(y = 1 | x) = 1 - p(\text{all instances are negative}) \]

\[ = 1 - \prod_{j=1}^{K} [1 - p(y = +1 | x_j)] = 1 - \prod_{j=1}^{K} \left[ 1 - \sigma(w^\top x_j) \right], \]

where the bag \( x = \{x_j\}_{j=1}^{K} \) contains \( K \) examples.

**Negative Bag**
A negative bag means that all instances in the bag are negative.

\[ p(y = 0 | x) = \prod_{j=1}^{K} p(y = 0 | x_j) = \prod_{j=1}^{K} \left[ 1 - \sigma(w^\top x_j) \right]. \]
Maximum Likelihood (ML) Estimator

**ML estimate**

Given the training data $\mathcal{D}$ the ML estimate for $w$ is given by

$$\hat{w}_{\text{ML}} = \arg \max_w \left[ \log p(\mathcal{D}|w) \right].$$
Maximum Likelihood (ML) Estimator

**ML estimate**

Given the training data $\mathcal{D}$ the ML estimate for $w$ is given by

$$\hat{w}_{ML} = \arg \max_w [\log p(\mathcal{D}|w)].$$

**Log-likelihood**

Assuming that the training bags are independent

$$\log p(\mathcal{D}|w) = \sum_{i=1}^{N} y_i \log p_i + (1 - y_i) \log(1 - p_i).$$

where $p_i = 1 - \prod_{j=1}^{K_i} \left[ 1 - \sigma(w^\top x_{ij}) \right]$ is the probability that the $i^{th}$ bag $x_i$ is positive.
ML estimator can exhibit severe over-fitting especially for high-dimensional data.
ML estimator can exhibit severe over-fitting especially for high-dimensional data.

MAP estimator

Use a prior on $w$ and then find the maximum a-posteriori (MAP) solution.

$$
\hat{w}_{\text{MAP}} = \arg\max_w p(w/D) \\
= \arg\max_w [\log p(D/w) + \log p(w)] .
$$
Our prior

Gaussian Prior

Zero mean Gaussian with inverse variance (precision) \( \alpha_i \).

\[
p(w_i | \alpha_i) = \mathcal{N}(w_i | 0, 1/\alpha_i).
\]

We assume that individual weights are independent.

\[
p(w) = \prod_{i=1}^{d} p(w_i | \alpha_i) = \mathcal{N}(w | 0, A^{-1}).
\]

\( A = \text{diag}(\alpha_1 \ldots \alpha_d) \)-also called hyper-parameters.
The final MAP Estimator

The optimization problem

Substituting for the log likelihood and the prior we have

$$\hat{w}_{\text{MAP}} = \arg \max_w L(w).$$

where

$$L(w) = \left[ \sum_{i=1}^{N} y_i \log p_i + (1 - y_i) \log(1 - p_i) \right] - \frac{w^\top A w}{2},$$
The final MAP Estimator

The optimization problem

Substituting for the log likelihood and the prior we have

$$\hat{w}_{\text{MAP}} = \arg \max_w L(w).$$

where

$$L(w) = \sum_{i=1}^{N} y_i \log p_i + (1 - y_i) \log(1 - p_i) \right] - \frac{w^\top A w}{2},$$

Newton-Raphson method

$$w^{t+1} = w^t - \eta H^{-1} g,$$

where $g$ is the gradient vector, $H$ is the Hessian matrix, and $\eta$ is the step length.
Outline of the talk

1. Multiple Instance Learning
2. Proposed algorithm
   - Training Data
   - Classifier form
   - Model
   - Estimator
   - Regularization
   - Optimization
3. Feature Selection
4. Experiments
5. Multi-task Learning
Feature Selection
Choosing the hyper-parameters

- We imposed a prior of the form $p(w) = \mathcal{N}(w|0, \mathbf{A}^{-1})$, parameterized by $d$ hyper-parameters $\mathbf{A} = \text{diag}(\alpha_1 \ldots \alpha_d)$. 
Feature Selection
Choosing the hyper-parameters

- We imposed a prior of the form \( p(w) = \mathcal{N}(w|0, A^{-1}) \), parameterized by \( d \) hyper-parameters \( A = \text{diag}(\alpha_1 \ldots \alpha_d) \).
- If we know the hyper-parameters we can compute the MAP estimate.
Feature Selection

Choosing the hyper-parameters

- We imposed a prior of the form $p(w) = \mathcal{N}(w|0, \mathbf{A}^{-1})$, parameterized by $d$ hyper-parameters $\mathbf{A} = \text{diag}(\alpha_1 \ldots \alpha_d)$.
- If we know the hyper-parameters we can compute the MAP estimate.
- As the precision $\alpha_k \to \infty$, i.e., the variance for $w_k$ tends to zero (thus concentrating the prior sharply at zero).
Feature Selection
Choosing the hyper-parameters

- We imposed a prior of the form $p(w) = \mathcal{N}(w|0, A^{-1})$, parameterized by $d$ hyper-parameters $A = diag(\alpha_1 \ldots \alpha_d)$.
- If we know the hyper-parameters we can compute the MAP estimate.
- As the precision $\alpha_k \rightarrow \infty$, i.e, the variance for $w_k$ tends to zero (thus concentrating the prior sharply at zero).
- posterior $\propto$ likelihood $\times$ prior
- Hence, regardless of the evidence of the training data, the posterior for $w_k$ will also be sharply concentrated on zero.
Feature Selection
Choosing the hyper-parameters

- We imposed a prior of the form $p(w) = \mathcal{N}(w|0, A^{-1})$, parameterized by $d$ hyper-parameters $A = \text{diag}(\alpha_1 \ldots \alpha_d)$.
- If we know the hyper-parameters we can compute the MAP estimate.
- As the precision $\alpha_k \to \infty$, i.e., the variance for $w_k$ tends to zero (thus concentrating the prior sharply at zero).
- posterior $\propto$ likelihood $\times$ prior
- Hence, regardless of the evidence of the training data, the posterior for $w_k$ will also be sharply concentrated on zero.
- Thus that feature will not affect the classification result—hence, it is effectively removed out via feature selection.
Feature Selection

Choosing the hyper-parameters

- We imposed a prior of the form $p(w) = \mathcal{N}(w|0, A^{-1})$, parameterized by $d$ hyper-parameters $A = \text{diag}(\alpha_1 \ldots \alpha_d)$.
- If we know the hyper-parameters we can compute the MAP estimate.
- As the precision $\alpha_k \to \infty$, i.e., the variance for $w_k$ tends to zero (thus concentrating the prior sharply at zero).
- posterior $\propto$ likelihood $\times$ prior
- Hence, regardless of the evidence of the training data, the posterior for $w_k$ will also be sharply concentrated on zero.
- Thus that feature will not affect the classification result-hence, it is effectively removed out via feature selection.
- Therefore, the discrete optimization problem corresponding to feature selection, can be more easily solved via an easier continuous optimization over hyper-parameters.
Feature Selection
Choosing the hyper-parameters

- We imposed a prior of the form $p(w) = \mathcal{N}(w|0, A^{-1})$, parameterized by $d$ hyper-parameters $A = \text{diag}(\alpha_1 \ldots \alpha_d)$.
- If we know the hyper-parameters we can compute the MAP estimate.
- As the precision $\alpha_k \rightarrow \infty$, i.e., the variance for $w_k$ tends to zero (thus concentrating the prior sharply at zero).
- posterior $\propto$ likelihood $\times$ prior
- Hence, regardless of the evidence of the training data, the posterior for $w_k$ will also be sharply concentrated on zero.
- Thus that feature will not affect the classification result—hence, it is effectively removed out via feature selection.
- Therefore, the discrete optimization problem corresponding to feature selection, can be more easily solved via an easier continuous optimization over hyper-parameters.
Feature Selection
Choosing the hyper-parameters to maximize the marginal likelihood

Type-II marginal likelihood approach for prior selection

\[ \hat{A} = \arg \max_A p(D | A) = \arg \max_A \int p(D | w) p(w | A) dw. \]
Feature Selection
Choosing the hyper-parameters to maximize the marginal likelihood

Type-II marginal likelihood approach for prior selection

\[
\hat{A} = \arg \max_A p(D|A) = \arg \max_A \int p(D|w)p(w|A)dw.
\]

- What hyper-parameters best describe the observed data?
Feature Selection
Choosing the hyper-parameters to maximize the marginal likelihood

Type-II marginal likelihood approach for prior selection

\[ \hat{A} = \arg \max_A \ p(D|A) = \arg \max_A \ \int p(D|w) p(w|A) \, dw. \]

- What hyper-parameters best describe the observed data?
- Not easy to compute.
- We use an approximation to the marginal likelihood via the Taylor series expansion around the MAP estimate.
Feature Selection
Choosing the hyper-parameters to maximize the marginal likelihood

**Type-II marginal likelihood approach for prior selection**

\[
\hat{A} = \arg \max_A p(D|A) = \arg \max_A \int p(D|w)p(w|A)dw.
\]

- What hyper-parameters best describe the observed data?
- Not easy to compute.
- We use an approximation to the marginal likelihood via the Taylor series expansion around the MAP estimate.

**Approximation to log marginal likelihood log \( p(D|A) \)**

\[
\log p(D|\hat{w}_{MAP}) - \frac{1}{2} \hat{w}_{MAP}^\top A \hat{w}_{MAP} + \frac{1}{2} \log |A| - \frac{1}{2} \log | - H(\hat{w}_{MAP}, A)|.
\]
Feature Selection

Choosing the hyper-parameters

Update Rule for hyperparameters

A simple update rule for the hyperparameters can be written by equating the first derivative to zero.

\[
\alpha_i^{\text{new}} = \frac{1}{w_i^2 + \sum_{ii}},
\]

where \(\sum_{ii}\) is the \(i^{th}\) diagonal element of \(H^{-1}(\hat{w}_{\text{MAP}}, A)^I\).
Feature Selection
Choosing the hyper-parameters

Update Rule for hyperparameters

A simple update rule for the hyperparameters can be written by equating the first derivative to zero.

\[ \alpha_{i}^{\text{new}} = \frac{1}{w_i^2 + \sum_{ii}} \]

where \( \sum_{ii} \) is the \( i^{th} \) diagonal element of \( H^{-1}(\hat{w}_{MAP}, A)I \).

Relevance vector Machine for MIL

- In an outer loop we update the hyperparameters \( A \).
- In an inner loop we find the MAP estimator \( \hat{w}_{MAP} \) given \( A \).
- After a few iterations we find that the hyperparameters for several features tend to infinity.
- This means that we can simply remove those irrelevant features.
Outline of the talk

1. Multiple Instance Learning

2. Proposed algorithm
   - Training Data
   - Classifier form
   - Model
   - Estimator
   - Regularization
   - Optimization

3. Feature Selection

4. Experiments

5. Multi-task Learning
## Benchmark Experiments

### Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Features</th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>examples</td>
<td>bags</td>
</tr>
<tr>
<td>Musk1</td>
<td>166</td>
<td>207</td>
<td>47</td>
</tr>
<tr>
<td>Musk2</td>
<td>166</td>
<td>1017</td>
<td>39</td>
</tr>
<tr>
<td>Elephant</td>
<td>230</td>
<td>762</td>
<td>100</td>
</tr>
<tr>
<td>Tiger</td>
<td>230</td>
<td>544</td>
<td>100</td>
</tr>
</tbody>
</table>
## Experiments

<table>
<thead>
<tr>
<th>Methods compared</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MI RVM</strong> Proposed method.</td>
</tr>
<tr>
<td><strong>MI</strong> Proposed method without feature selection.</td>
</tr>
<tr>
<td><strong>RVM</strong> Proposed method without MIL.</td>
</tr>
<tr>
<td><strong>MI LR</strong> MIL variant of Logistic Regression. (Settles et al., 2008)</td>
</tr>
<tr>
<td><strong>MI SVM</strong> MIL variant of SVM. (Andrews et al., 2002)</td>
</tr>
<tr>
<td><strong>MI Boost</strong> MIL variant of AdaBoost. (Xin and Frank, 2004)</td>
</tr>
</tbody>
</table>
Experiments

Evaluation Procedure

- 10-fold stratified cross-validation.

- We plot the Receiver Operating Characteristics (ROC) curve for various algorithms.

- The True Positive Rate is computed on a bag level.

- The ROC curve is plotted by pooling the prediction of the algorithm across all folds.

- We also report the area under the ROC curve (AUC).
### AUC Comparison

#### Area under the ROC Curve

<table>
<thead>
<tr>
<th>Set</th>
<th>MIRVM</th>
<th>RVM</th>
<th>MIBoost</th>
<th>MILR</th>
<th>MISVM</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musk1</td>
<td>0.942</td>
<td><strong>0.951</strong></td>
<td>0.899</td>
<td>0.846</td>
<td>0.899</td>
<td>0.922</td>
</tr>
<tr>
<td>Musk2</td>
<td><strong>0.987</strong></td>
<td>0.985</td>
<td>0.964</td>
<td>0.795</td>
<td>-</td>
<td>0.982</td>
</tr>
<tr>
<td>Elephant</td>
<td>0.962</td>
<td><strong>0.979</strong></td>
<td>0.828</td>
<td>0.814</td>
<td>0.959</td>
<td>0.953</td>
</tr>
<tr>
<td>Tiger</td>
<td><strong>0.980</strong></td>
<td>0.970</td>
<td>0.890</td>
<td>0.890</td>
<td>0.945</td>
<td>0.956</td>
</tr>
</tbody>
</table>

### Observations

1. The proposed method MIRVM and RVM clearly perform better.
2. For some datasets RVM is better, i.e., MIL does not help.
3. Feature selection helps (MIRVM is better than MI).
ROC Comparison

Musk2

False Positive Rate

True Positive Rate

MIRVM
RVM
MIBoost
MILR
MI

0 0.2 0.4 0.6 0.8 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

0 0.2 0.4 0.6 0.8 1
ROC Comparison

![ROC curves for different classifiers](image)

- MIRVM
- RVM
- MIBoost
- MILR
- MISVM
- MI

Vikas C. Raykar (Siemens)

ICML 2008

July 8, 2008
ROC Comparison

Elephant

False Positive Rate

True Positive Rate

MIRVM
RVM
MIBOost
MILR
MISVM
MI

Vikas C. Raykar (Siemens)
ICML 2008
July 8, 2008 35 / 42
### Features selected

#### The average number of features selected

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of features</th>
<th>selected by RVM</th>
<th>selected by MI RVM</th>
<th>selected by MI Boost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musk1</td>
<td>166</td>
<td>39</td>
<td>14</td>
<td>33</td>
</tr>
<tr>
<td>Musk2</td>
<td>166</td>
<td>90</td>
<td>17</td>
<td>32</td>
</tr>
<tr>
<td>Elephant</td>
<td>230</td>
<td>42</td>
<td>16</td>
<td>33</td>
</tr>
<tr>
<td>Tiger</td>
<td>230</td>
<td>56</td>
<td>19</td>
<td>37</td>
</tr>
</tbody>
</table>

**Observation**

Multiple instance learning (MIRVM) selects much less features than single instance learning (RVM).
PECAD Experiments
Selected 21 out of 134 features.

PECAD bag level FROC Curve

False Positives/ Volume

Sensitivity

MI RVM
RVM
MI Boost

Vikas C. Raykar (Siemens)
Outline of the talk

1. Multiple Instance Learning

2. Proposed algorithm
   - Training Data
   - Classifier form
   - Model
   - Estimator
   - Regularization
   - Optimization

3. Feature Selection

4. Experiments

5. Multi-task Learning
Multi-task Learning

Learning multiple related classifiers.
May have a shortage of training data for learning classifiers for a task.
Multi-task learning can exploit information from other datasets.
The classifiers share a common prior.
A separate classifier is trained for each task.
However the optimal hyper-parameters of the shared prior are estimated from all the data sets simultaneously.
Multi-task Learning
LungCAD nodule (solid and GGOs) detection
Multi-task Learning Experiments

The bag level FROC curve for the solid validation set.
Conclusion

**MIRVM—Multiple Instance Relevance Vector Machine**

- Joint feature selection and classifier learning in the MIL scenario.
- MIL selects much sparser models.
- More accurate and faster than some competing methods.
- Extension to multi-task learning.