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

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 $x \in \mathbf{R}^d$ predict the class label $y \in \{1, 0\}$.

Linear Binary Classifier

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

Linear Binary Classifier

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

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

Linear Binary Classifier

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

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

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

Linear Binary Classifier

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

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

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

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

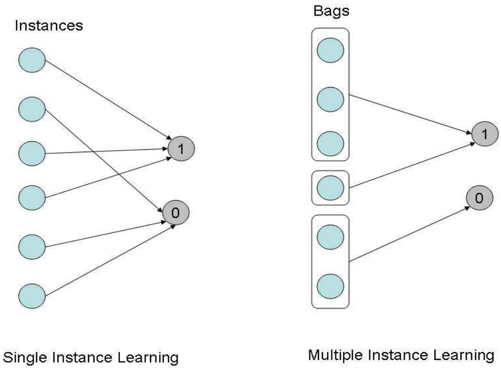
Single Instance Learning

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

Multiple Instance Learning

a group of examples (bag) $\mathbf{x}_i = \{x_{ij} \in \mathbf{R}^d\}_{j=1}^{K_i}$ **share a common label**

Single 'vs' Multiple Instance Learning



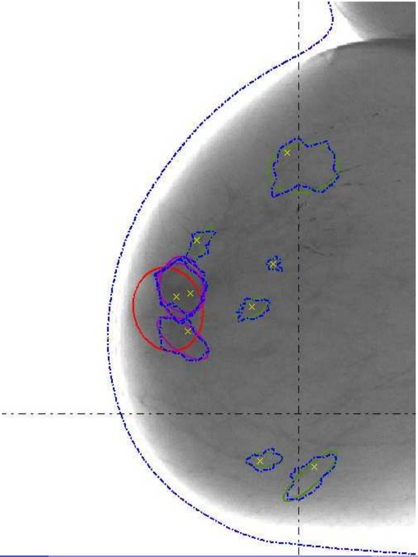
MIL applications

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.**

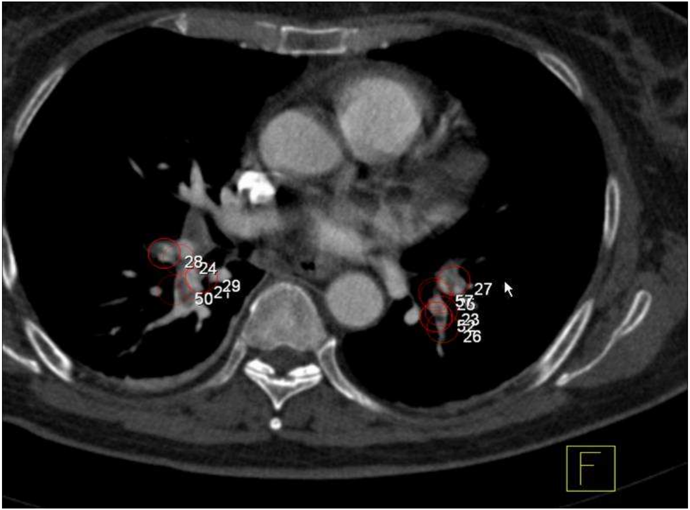
Computer Aided Diagnosis as a MIL problem

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

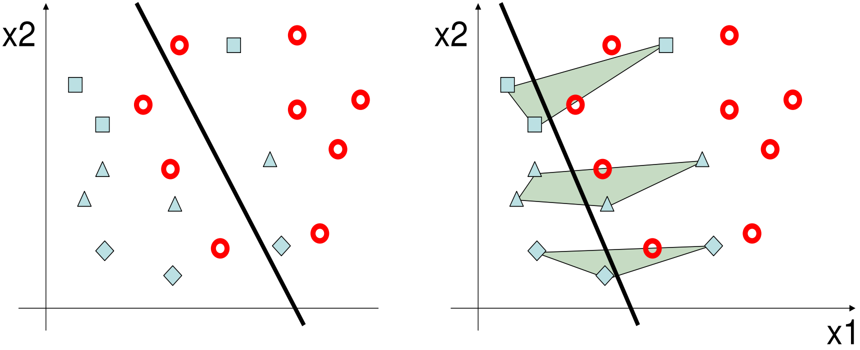
A lesion is detected if at least one of the candidate which overlaps with it is detected.

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

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 $x \in \mathbf{R}^d$.

Training Data

Consists of N bags

Notation

- We represent an instance as a feature vector $x \in \mathbf{R}^d$.
- A bag which contains K instances is denoted by boldface $\mathbf{x} = \{x_j \in \mathbf{R}^d\}_{j=1}^{K}$.

Training Data

Consists of N bags

Notation

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

Training Data

The training data $\mathcal{D}$ consists of N bags $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where

- $\mathbf{x}_i = \{x_{ij} \in \mathbf{R}^d\}_{j=1}^{K_i}$ 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** $f_w(x) = w^T x$

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

Single Instance Model

Logistic regression

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.

$$
\begin{aligned}
p(y=1|\mathbf{x}) &= 1 - p(\text{all instances are negative}) \\
&= 1 - \prod_{j=1}^{K} [1 - p(y=+1|x_j)] = 1 - \prod_{j=1}^{K} \left[1 - \sigma(w^\top x_j)\right],
\end{aligned}
$$

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

Multiple Instance Model

Logistic regression

Positive Bag

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

$$\begin{aligned} p(y=1|\mathbf{x}) &= 1 - p(\text{all instances are negative}) \\ &= 1 - \prod_{j=1}^{K}[1 - p(y=+1|x_j)] = 1 - \prod_{j=1}^{K}\left[1 - \sigma(w^\top x_j)\right], \end{aligned}$$

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

Negative Bag

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

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

Maximum Likelihood (ML) Estimator

ML estimate

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

$$\widehat{w}_{\text{ML}} = \arg\max_{w} \left[\log p(\mathcal{D}|w)\right].$$

Maximum Likelihood (ML) Estimator

ML estimate

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

$$\widehat{w}_{\text{ML}} = \arg\max_{w} \left[\log p(\mathcal{D}|w)\right].$$

Log-likelihood

Assuming that the training bags are independent

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

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 $\mathbf{x}_i$ is positive.

MAP estimator

Regularization

ML estimator can exhibit severe over-fitting especially for high-dimensional data.

MAP estimator

Regularization

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.

$$\begin{aligned} \widehat{w}_{\text{MAP}} &= \arg\max_{w} p(w/\mathcal{D}) \\ &= \arg\max_{w} \left[\log p(\mathcal{D}/w) + \log p(w)\right]. \end{aligned}$$

Our prior

Gaussian Prior

Zero mean Gaussian with inverse variance (precision) α_i.

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

We assume that individual weights are independent.

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

$\mathbf{A} = diag(\alpha_1 \ldots \alpha_d)$-also called **hyper-parameters**.

The final MAP Estimator

The optimization problem

Substituting for the log likelihood and the prior we have

$$\widehat{w}_{\text{MAP}} = \arg\max_{w} L(w).$$

where

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

The final MAP Estimator

The optimization problem

Substituting for the log likelihood and the prior we have

$$\widehat{w}_{\text{MAP}} = \arg\max_{w} L(w).$$

where

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

Newton-Raphson method

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

where $\mathbf{g}$ is the gradient vector, $\mathbf{H}$ is the Hessian matrix, and η is the step length.

Outline of the talk

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 $\mathbf{A} = diag(\alpha_1 \ldots \alpha_d)$.

Feature Selection

Choosing the hyper-parameters

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

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 $\mathbf{A} = diag(\alpha_1 \ldots \alpha_d)$.
- 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 $p(w) = \mathcal{N}(w|0, \mathbf{A}^{-1})$, parameterized by d hyper-parameters $\mathbf{A} = diag(\alpha_1 \ldots \alpha_d)$.
- 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 $p(w) = \mathcal{N}(w|0, \mathbf{A}^{-1})$, parameterized by d hyper-parameters $\mathbf{A} = diag(\alpha_1 \ldots \alpha_d)$.
- 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 to maximize the marginal likelihood

Type-II marginal likelihood approach for prior selection

$$\widehat{\mathbf{A}} = \arg\max_{\mathbf{A}} p(\mathcal{D}|\mathbf{A}) = \arg\max_{\mathbf{A}} \int p(\mathcal{D}|w)p(w|\mathbf{A})dw.$$

Feature Selection

Choosing the hyper-parameters to maximize the marginal likelihood

Type-II marginal likelihood approach for prior selection

$$\widehat{\mathbf{A}} = \arg\max_{\mathbf{A}} p(\mathcal{D}|\mathbf{A}) = \arg\max_{\mathbf{A}} \int p(\mathcal{D}|w)p(w|\mathbf{A})dw.$$

- 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

$$\widehat{\mathbf{A}} = \arg\max_{\mathbf{A}} p(\mathcal{D}|\mathbf{A}) = \arg\max_{\mathbf{A}} \int p(\mathcal{D}|w)p(w|\mathbf{A})dw.$$

- 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

$$\widehat{\mathbf{A}} = \arg\max_{\mathbf{A}} p(\mathcal{D}|\mathbf{A}) = \arg\max_{\mathbf{A}} \int p(\mathcal{D}|w)p(w|\mathbf{A})dw.$$

- 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 $\log p(\mathcal{D}|\mathbf{A})$

$$\log p(\mathcal{D}|\widehat{w}_{\text{MAP}}) - \frac{1}{2}\widehat{w}_{\text{MAP}}^{\top}\mathbf{A}\widehat{w}_{\text{MAP}} + \frac{1}{2}\log|\mathbf{A}| - \frac{1}{2}\log|-\mathbf{H}(\widehat{w}_{\text{MAP}},\mathbf{A})|.$$

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.

$$\alpha_i^{\text{new}} = \frac{1}{w_i^2 + \Sigma_{ii}},$$

where Σ_{ii} is the i^{th} diagonal element of $\mathbf{H}^{-1}(\widehat{w}_{\text{MAP}}, \mathbf{A})\mathbf{I}$.

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.

$$\alpha_i^{\text{new}} = \frac{1}{w_i^2 + \Sigma_{ii}},$$

where Σ_{ii} is the i^{th} diagonal element of $\mathbf{H}^{-1}(\widehat{w}_{\text{MAP}}, \mathbf{A})\mathbf{I}$.

Relevance vector Machine for MIL

- In an outer loop we update the hyperparameters $\mathbf{A}$.
- In an inner loop we find the MAP estimator $\widehat{w}_{\text{MAP}}$ given $\mathbf{A}$.
- 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

Benchmark Experiments

Datasets

Dataset	Features	positive		negative	
		examples	bags	examples	bags
Musk1	166	207	47	269	45
Musk2	166	1017	39	5581	63
Elephant	230	762	100	629	100
Tiger	230	544	100	676	100

Experiments

Methods compared

- **MI RVM** Proposed method.
- **MI** Proposed method without feature selection.
- **RVM** Proposed method without MIL.
- **MI LR** MIL variant of Logistic Regression. (Settles et al., 2008)
- **MI SVM** MIL variant of SVM. (Andrews et al., 2002)
- **MI Boost** MIL variant of AdaBoost. (Xin and Frank, 2004)

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

Set	MIRVM	RVM	MIBoost	MILR	MISVM	MI
Musk1	0.942	**0.951**	0.899	0.846	0.899	0.922
Musk2	**0.987**	0.985	0.964	0.795	-	0.982
Elephant	0.962	**0.979**	0.828	0.814	0.959	0.953
Tiger	**0.980**	0.970	0.890	0.890	0.945	0.956

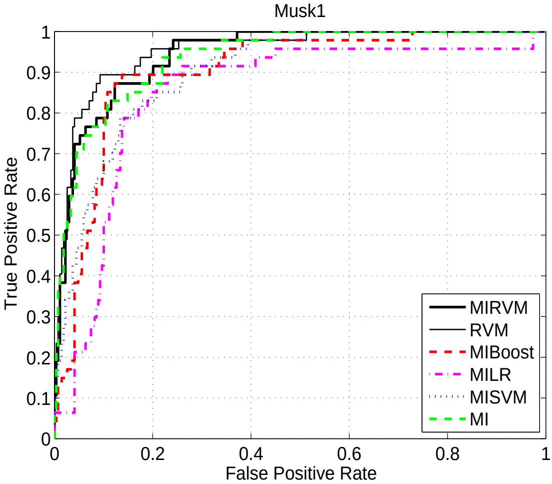
Observations

(1) The proposed method MIRVM and RVM clearly perform better.

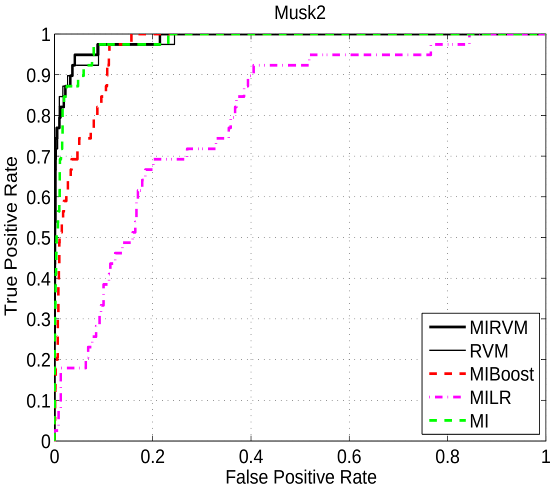
(2) For some datasets RVM is better, *i.e*, MIL does not help.

(3) Feature selection helps (MIRVM is better than MI).

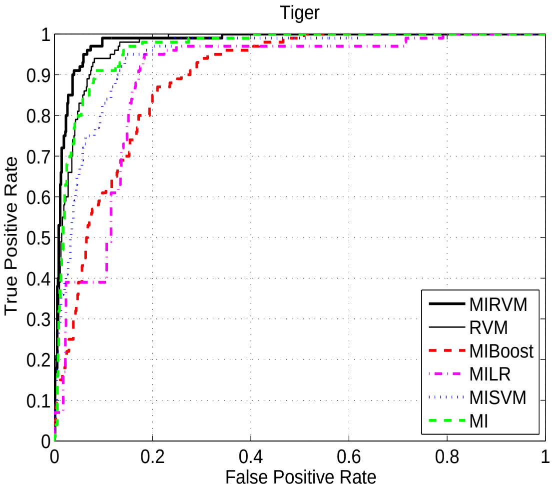
ROC Comparison



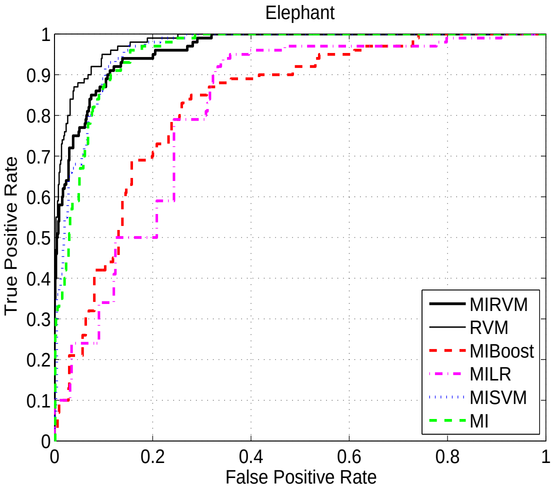
ROC Comparison



ROC Comparison



ROC Comparison



Features selected

The average number of features selected

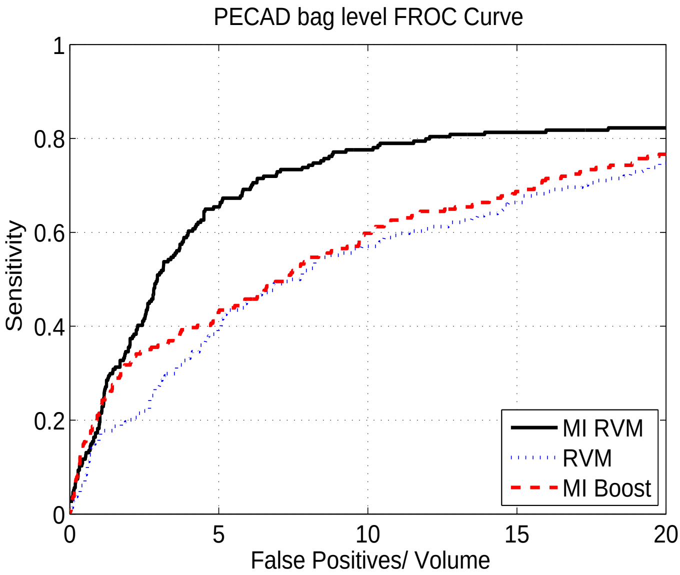
Dataset	Number of features	selected by RVM	selected by MI RVM	selected by MI Boost
Musk1	166	39	**14**	33
Musk2	166	90	**17**	32
Elephant	230	42	**16**	33
Tiger	230	56	**19**	37

Observation

Multiple instance learning (MIRVM) selects much less features than single instance learning (RVM).

PECAD Experiments

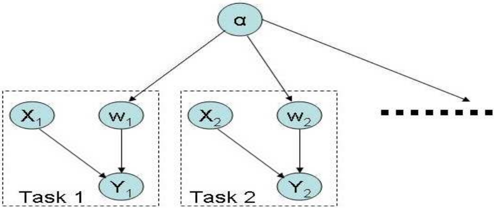
Selected 21 out of 134 features.



Outline of the talk

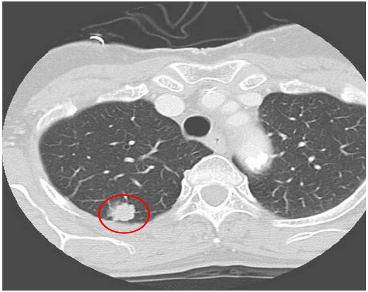
Multi-task Learning

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.



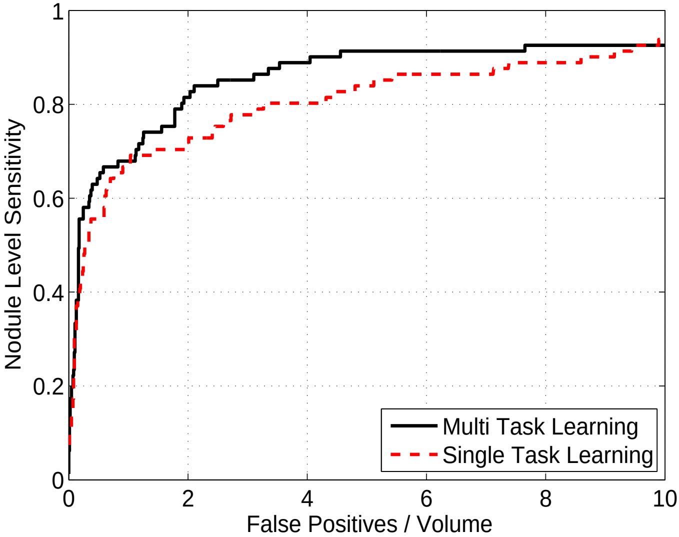
Multi-task Learning

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.