

# Multi-Task Feature Selection using the Multiple Inclusion Criterion (MIC)

Paramveer S. Dhillon<sup>1</sup>   Brian Tomasik<sup>2</sup>   Dean Foster<sup>3</sup>  
Lyle Ungar<sup>1</sup>

<sup>1</sup>Computer and Information Science

<sup>3</sup>Statistics, Wharton School

University of Pennsylvania, Philadelphia, PA, U.S.A

<sup>2</sup>Computer Science Department

Swarthmore College, Swarthmore, PA, U.S.A

ECML-PKDD 2009



# Outline

- 1 Introduction
- 2 Previous Work
- 3 MIC: The Model
- 4 Experiments
- 5 Conclusion



# Multiple Inclusion Criterion (MIC)

- Addresses Joint feature selection for related tasks
  - A set of related tasks, with shared feature space
  - Large number of available features, only a handful are finally relevant.
  - Goal is better predictive accuracy and interpretability of selected features
  - Example tasks: predict size of the tumor and prognosis, using gene expression values.



# Multiple Inclusion Criterion (MIC)

- Addresses Joint feature selection for related tasks
  - A set of related tasks, with shared feature space
  - Large number of available features, only a handful are finally relevant.
  - Goal is better predictive accuracy and interpretability of selected features
  - Example tasks: predict size of the tumor and prognosis, using gene expression values.
- MIC imposes sparsity at two levels ( $\ell_0 - \ell_0$ )
  - Small number of features selected for each task (first  $\ell_0$ )
  - Small number of tasks associated with each feature (second  $\ell_0$ )



# Previous Work

- BBLasso [*Obozinski, Taskar and Jordan '09*]
  - $l_1/l_2$  penalty to enforce “Block sparsity”, feature added to all or none of the tasks.
  - Convex problem.
  - **Exact solution to approximate problem**



# Previous Work

- BBLasso [*Obozinski, Taskar and Jordan '09*]
  - $l_1/l_2$  penalty to enforce “Block sparsity”, feature added to all or none of the tasks.
  - Convex problem.
  - **Exact solution to approximate problem**
  
- Proposed method: MIC
  - uses  $l_0 - l_0$  penalty
  - non-convex problem, but greedy algorithm yields good approximation in practice.
  - **Approximate solution to exact problem**



# General Methodology

- MIC uses MDL based coding scheme to specify a penalized likelihood method.
- The Total Description Length (TDL) can be written as:

$$S = S_E + S_M$$

$S_E$   $\mapsto$  # Bits for encoding the residual errors given the model.

$S_M$   $\mapsto$  # Bits for encoding the model



# General Methodology

- MIC uses MDL based coding scheme to specify a penalized likelihood method.
- The Total Description Length (TDL) can be written as:

$$S = S_E + S_M$$

$S_E \mapsto$  # Bits for encoding the residual errors given the model.

$S_M \mapsto$  # Bits for encoding the model

- Maximize reduction in Description Length when adding a feature 'j' to the model:

$$\max \Delta S_j = \Delta S_{jE} - \Delta S_{jM}$$



# Simple $\ell_0$ regression - Independent MIC (baseline)

- Assume a simple linear regression model:  $Y = WX + \epsilon$



# Simple $\ell_0$ regression - Independent MIC (baseline)

- Assume a simple linear regression model:  $Y = WX + \epsilon$
- In this case:  $S_E = -\log\left(\exp\left(-\frac{\sum_{i=1}^n (y_i - wx_i)^2}{2\sigma^2}\right)\right)$
- $S_M = \log(m) + 2$  i.e. bits to code the feature (RIC Penalty) and its coefficient (AIC Penalty).



# Coding Schemes for MIC

- The goal is to maximize  $\Delta S_j^k = \Delta S_{jE}^k - \Delta S_{jM}^k$  i.e. the reduction in TDL by adding a feature 'j' to a subset  $k$  of  $h$  tasks.



# Coding Schemes for MIC

- The goal is to maximize  $\Delta S_j^k = \Delta S_{jE}^k - \Delta S_{jM}^k$  i.e. the reduction in TDL by adding a feature 'j' to a subset  $k$  of  $h$  tasks.
- The cost to code  $S_{jE}^k$  i.e. the decrease in error by adding the feature to the model of  $k$  tasks is:
- $S_E = -\log(P(Y|X, w))$

$$P(Y|X, w) = \frac{1}{((2\pi)^h |\Sigma|)^{\frac{n}{2}}} \exp\left(-\frac{1}{2} \sum_{i=1}^n [(y_i - wx_i)^T \Sigma^{-1} (y_i - wx_i)]\right)$$

where  $\Sigma$  is the  $h \times h$  covariance matrix.



# Coding Schemes for MIC contd ...

- The cost to code the model is  $S_{jM}^k$  is  $I_l + I_H + I_\theta$
- Cost to code:
  - The feature being included:  $I_l = \log(m)$ .
  - How many and which of the  $h$  tasks have that feature  $I_H = \log(h) + \log\binom{h}{k}$
  - The coefficient of the feature being included:  $I_\theta = 2 \times k$ .



# Variations of MIC

- **Full MIC**

- A feature is added to all or none of the tasks
- Similar to BBLasso [*Obozinski, Taskar & Jordan '09*]
- $I_H$  bits saved in the coding



# Variations of MIC

## ● Full MIC

- A feature is added to all or none of the tasks
- Similar to BBLasso [*Obozinski, Taskar & Jordan '09*]
- $I_H$  bits saved in the coding

## ● Independent MIC

- Each task is modeled in isolation.
- $I_H$  bits saved in the coding



# Variations of MIC

## ● Full MIC

- A feature is added to all or none of the tasks
- Similar to BBLasso [*Obozinski, Taskar & Jordan '09*]
- $I_H$  bits saved in the coding

## ● Independent MIC

- Each task is modeled in isolation.
- $I_H$  bits saved in the coding

## ● Partial MIC

- A feature can be added to all, none or a subset of the tasks.



# Experimental Setup

Name	# Tasks $h$	# Obs. $n$	# Features $m$	Source
Yeast Dataset	20	104	6715	[Litvin et. al. '09]
Breast Cancer	5	100	5000	[van't Veer et. al. '02]

- We compared the three versions of MIC (Partial, Full and Independent) against BBLasso [Obozinski et. al. '09] and AndoZhang [Ando et. al. '05]
- For BBLasso and AndoZhang we used their standard implementations from Berkeley Transfer Learning Toolkit.



## Experiments contd...

**Table:** 5 fold CV accuracies. *Note:* AndoZhang's NA values are due to the fact that it does not explicitly select features.

Method	Test Error	# Features Selected	# Active Coefs.
Yeast Dataset			
Partial MIC	<b><math>0.38 \pm 0.04</math></b>	$4 \pm 0$	$22 \pm 4$
Full MIC	$0.39 \pm 0.04$	$3 \pm 0$	$64 \pm 4$
Independent	$0.41 \pm 0.05$	$9 \pm 1$	$9 \pm 1$
AndoZhang	$0.39 \pm 0.03$	NA	NA
BBLasso	$0.43 \pm 0.03$	$63 \pm 14$	$1268 \pm 279$



## Experiments contd...

**Table:** 5 fold CV accuracies. *Note:* AndoZhang's NA values are due to the fact that it does not explicitly select features.

Method	Test Error	# Features Selected.	# Active Coefs.
Breast Cancer Dataset			
Partial MIC	<b>0.33</b> $\pm$ <b>0.08</b>	2 $\pm$ 0	3 $\pm$ 0
Full MIC	0.37 $\pm$ 0.08	2 $\pm$ 0	11 $\pm$ 1
Independent	0.36 $\pm$ 0.08	2 $\pm$ 0	2 $\pm$ 0
AndoZhang	0.44 $\pm$ 0.03	NA	NA
BBLasso	<b>0.33</b> $\pm$ <b>0.08</b>	12 $\pm$ 4	61 $\pm$ 19



# Conclusion

- MIC gives flexible coding schemes for doing “joint feature selection” in related tasks.
- Coding schemes can easily be customized to fit the problem at hand
- They capture the spirit of Bayesian priors
- Significantly (5% level, paired t-test) better than AndoZhang, on Yeast and Breast Cancer datasets.
- Comparable in accuracy to BBLasso, but provides simpler and sparser models.



# Thanks

