Efficient Feature Selection in the Presence of Multiple Feature Classes

Paramveer Dhillon [1] Dean Foster [2] Lyle Ungar [1]

[1]Computer and Information Science
[2]Statistics, Wharton School

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

International Conference on Data Mining (ICDM), 2008

Outline

1. Features come in Classes
2. Our Approach
   - Feature Selection using Information Theory
   - Three Part Coding (TPC) scheme
3. Experimental Results
4. Conclusion

Feature Classes are Everywhere...

- Genes may be divided into different Gene Families or Pathways.
- In WSD tasks, one can have feature classes like adjacent words (*word-1, word+1*), the part of speech of the word (*pos*), the topic of the document the word is in (*tp*) etc.

Or...

Feature Classes are Everywhere...

- Genes may be divided into different Gene Families or Pathways.
- In WSD tasks, one can have feature classes like adjacent words (*word-1, word+1*), the part of speech of the word (*pos*), the topic of the document the word is in (*tp*) etc.

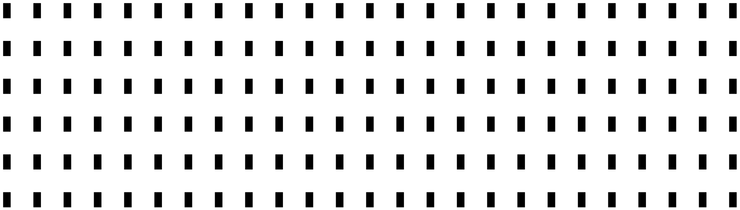
Or...

- You can use your favorite NIPS / UCI datasets and get new feature classes by doing PCA, NNMF, Square, Square Root or any other algebraic manipulation of the original features!

Feature Classes in Data

X Matrix with Feature Classes

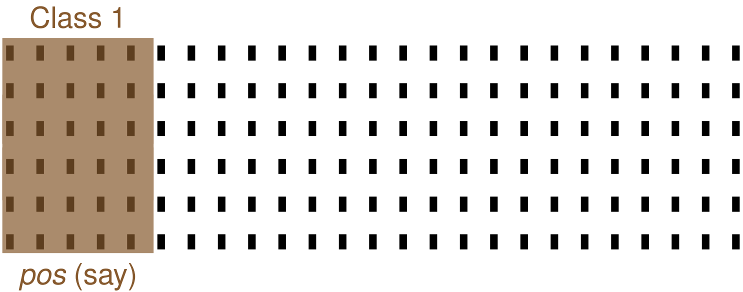
(Standard Setting)



Feature Classes in Data

X Matrix with Feature Classes

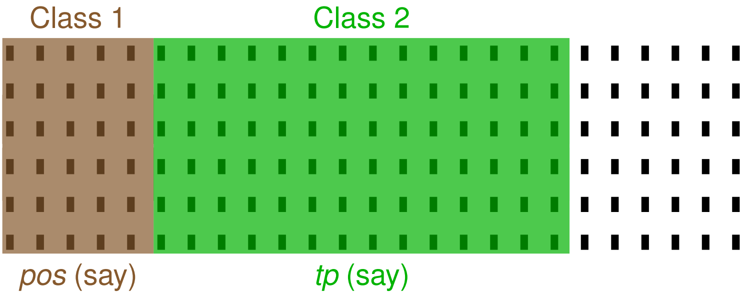
(Standard Setting)



Feature Classes in Data

X Matrix with Feature Classes

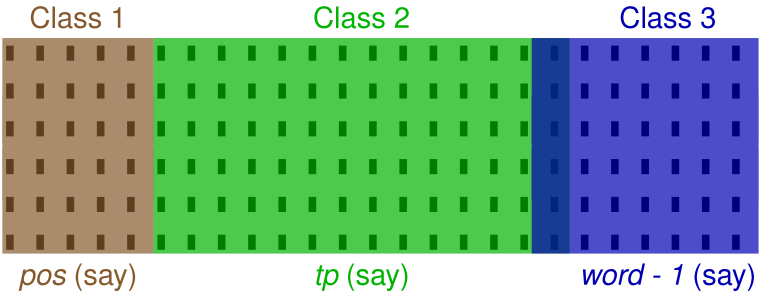
(Standard Setting)



Feature Classes in Data

X Matrix with Feature Classes

(Standard Setting)



Feature Classes contd . . .

- Feature Classes differ in how dense they are in beneficial features.

- For e.g. when disambiguating various senses of word Paris (i.e. either Paris (France) or Paris Hilton), a feature class like topic of the document would contain more beneficial features than a feature class like # words in the document.

Feature Classes contd . . .

- Feature Classes differ in how dense they are in beneficial features.

- For e.g. when disambiguating various senses of word Paris (i.e. either Paris (France) or Paris Hilton), a feature class like topic of the document would contain more beneficial features than a feature class like # words in the document.

- Standard L_0 and L_1 penalty based feature selection methods are oblivious to this structure in data.

The Big Picture of our Model

- The main idea:
  Once you have found good feature classes, preferentially draw features from them.

The Big Picture of our Model

- The main idea:
  Once you have found good feature classes, preferentially draw features from them.

- We provide an information theoretic approach called Three Part Coding to exploit the structure in data.

The Big Picture of our Model

- The main idea:
  Once you have found good feature classes, preferentially draw features from them.
- We provide an information theoretic approach called Three Part Coding to exploit the structure in data.
- TPC is a penalized likelihood method based on the MDL principle.
- TPC assumes a setting in which $n << p$ i.e. lots of features and few observations.

Formulation of TPC scheme

- The Total Description Length (TDL) can be written as:

$$S = S_E + S_M$$

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

$S_M \longmapsto$ # Bits for encoding the model

Formulation of TPC scheme

- The Total Description Length (TDL) can be written as:

  $$S = S_E + S_M$$

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

- Reduction in TDL by adding feature 'i' to the model:

  $$\Delta S^i = \Delta S^i_E - \Delta S^i_M$$

- The goal is to maximize ΔS i.e. paying less for adding a new feature, while at the same time getting more accurate model due to increased likelihood.

Formulation of TPC scheme contd . . .

- $\Delta S_E^i = (log_2(likelihood)|_{i \bigcup model}) - (log_2(likelihood)|_{model \setminus i})$

Formulation of TPC scheme contd ...

- $\Delta S_E^i = (log_2(likelihood)|_{i \bigcup model}) - (log_2(likelihood)|_{model \setminus i})$

- We assume a Gaussian model for linear regression:

$$\text{likelihood} \sim e^{-(\frac{\sum_{i=1}^{n}(y_i - wx_i)^2}{2\sigma^2})}$$

Formulation of TPC scheme contd ...

- $\Delta S_E^i = (log_2(likelihood)|_{i \bigcup model}) - (log_2(likelihood)|_{model \setminus i})$

- We assume a Gaussian model for linear regression:

$$\text{likelihood} \sim e^{-\left(\frac{\sum_{i=1}^{n}(y_i - wx_i)^2}{2\sigma^2}\right)}$$

- $\Delta S_M = l_c + l_i + l_\theta$ (Three Part Coding)

  $l_c \longmapsto$ # Bits to code the index of the feature class of the feature.

  $l_i \longmapsto$ # Bits to code the index of the feature within its feature class.

  $l_\theta \longmapsto$ # Bits to code the coefficient of the feature.

Formulation of TPC scheme contd ...

- If there are 'K' feature classes and
'Q' of them are currently in the model, then

$$l_C = \begin{cases} log(K) & \textit{if the feature class is not in the model} \\ log(Q) & \textit{if the feature class is already in the model} \end{cases}$$

Formulation of TPC scheme contd . . .

- If there are 'K' feature classes and 'Q' of them are currently in the model, then

$$l_C = \begin{cases} log(K) & \textit{if the feature class is not in the model} \\ log(Q) & \textit{if the feature class is already in the model} \end{cases}$$

- $l_I = log(p_k)$, where p_k = # Features in the k^{th} Feature Class (RIC or Bonferroni Penalty)

Formulation of TPC scheme contd . . .

- If there are 'K' feature classes and 'Q' of them are currently in the model, then

$$l_C = \begin{cases} log(K) & \textit{if the feature class is not in the model} \\ log(Q) & \textit{if the feature class is already in the model} \end{cases}$$

- $l_I = log(p_k)$, where p_k = # Features in the k^{th} Feature Class (RIC or Bonferroni Penalty)
- $l_\theta = 2$ (AIC like penalty)

Analysis of TPC Scheme

- Baseline= RIC Penalty for feature indices + AIC Penalty for coefficients
- RIC $\longmapsto$ Optimal penalty for the case n $<<$ p [Foster and George ‘94]

Analysis of TPC Scheme

- Baseline= RIC Penalty for feature indices + AIC Penalty for coefficients
- RIC $\longmapsto$ Optimal penalty for the case n $<<$ p [Foster and George '94]
- **TPC wins when all or most of the features lie in a small number of Feature Classes i.e. multiple 'good' features per Feature Class.**
- $[TPC - Baseline]_{bits} = (q - Q)log(\frac{K}{Q})$
  q $\longmapsto$ # Features Selected
  Q $\longmapsto$ # Feature Classes Selected
  K $\longmapsto$ Total # Feature Classes

Experimental Setup and Datasets Used

- WSD Datasets containing 6 verbs $\sim$ 7000 features each [Chen and Palmer '05].
- Feature Classes $\longmapsto$ *'tp', 'word-1', 'word+1' , 'pos' etc.* [# 75]

Experimental Setup and Datasets Used

- WSD Datasets containing 6 verbs $\sim$ 7000 features each [Chen and Palmer ‘05].
- Feature Classes $\longmapsto$ *‘tp’, ’word-1’, ’word+1’ , ‘pos’ etc.* [# 75]
- GSEA Datasets containing 5 different phenotypes and $\sim$ 10000 features each [Mootha et. al ‘03].
- Feature Classes $\longmapsto$ *Gene Classes* [# 318]

Experimental Setup and Datasets Used

- WSD Datasets containing 6 verbs $\sim$ 7000 features each [Chen and Palmer '05].
- Feature Classes $\longmapsto$ *'tp', 'word-1', 'word+1' , 'pos' etc.* [# 75]
- GSEA Datasets containing 5 different phenotypes and $\sim$ 10000 features each [Mootha et. al '03].
- Feature Classes $\longmapsto$ *Gene Classes* [# 318]
- We compare TPC to its standard counterparts i.e. Standard Stepwise and Streamwise Regressions and also to Lasso [Tibshirani '96] [L_1 penalty] and Elastic Nets [Zou and Hastie '05] [$L_1 + L_2$ penalty].

WSD Experimental Results

- 10 Fold CV Errors (RMS Value) for WSD Datasets

Dataset	TPC	RIC	Lasso	EN
ADD	0.39	0.42	0.42	0.40
BEGIN	0.27	0.31	0.32	0.32
CALL	0.15	0.16	0.24	0.30
CARRY	0.26	0.29	0.37	0.30
DEVELOP	0.41	0.42	0.52	0.48
DRAW	0.20	0.23	0.24	0.25

Genomic Experimental Results

- 10 Fold CV Errors (RMS Value) for GSEA Datasets

Dataset	TPC	RIC	Lasso	EN
Leukemia	0.41	0.43	0.71	0.71
Gender 1	0.21	0.24	0.73	0.73
Diabetes	0.51	0.53	0.59	0.66
Gender 2	0.21	0.26	0.90	0.84
P53	0.52	0.53	0.75	0.72

- TPC is always (significantly) better.
- Only $\sim$ 14 % Feature Classes and $<$ 1% features get selected on average. So, TPC induces sparsity at the level of Feature Classes as well at the level of features.

Conclusion

- TPC is a penalized likelihood method
  - based on the MDL principle.
  - exploits the presence of feature classes
    - unlike standard L_0 and L_1 methods
  - works best when good features are concentrated in a small number of features classes
  - outperforms competitors on WSD and GSEA datasets

Thanks