

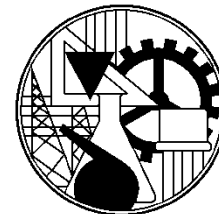
# Unsupervised feature discretization and selection for sparse data

Priberam Machine Learning Lunch Seminar  
Lisbon, 1<sup>st</sup> March 2011

Artur Ferreira

([arturj@isel.pt](mailto:arturj@isel.pt))

Prof. at ISEL &



PhD Student at IST-IT (Supervisor: Prof. Mário Figueiredo)

1. High-dimensional datasets
  - Some datasets with sparse data
  - Text categorization
2. Feature Selection (FS) / Feature Reduction (FR)
  - Unsupervised (and supervised) approaches
  - Compressed Learning theory
3. Feature Discretization (FD)
  - Unsupervised (and supervised) approaches
4. Analysis of FS methods
  - Experimental results and discussion
5. Analysis of FD and FD + FS methods
  - Experimental results and discussion
6. Concluding Remarks

# 1.1 High-dimensional datasets

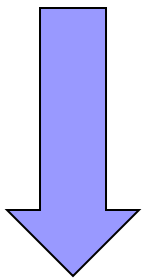
- In many machine learning problems we deal with high-dimensional datasets (  $p$  features,  $n$  patterns )

Dataset	$p$ features	$n$ patterns	Type of data	Problem
Arcene	10000	900	Dense integer	Cancer detection
Gisette	5000	13500	Dense integer	Distinguish between confusable handwritten digits 4 and 9
Dexter	20000	2600	Sparse - Bag of Words	Text classification
Dorothea	<u>100000</u>	1950	Sparse binary input variables	Detection of Thrombin compound
Example 1 (Reuters)	9947	2600	Sparse - Bag of Words	Text classification

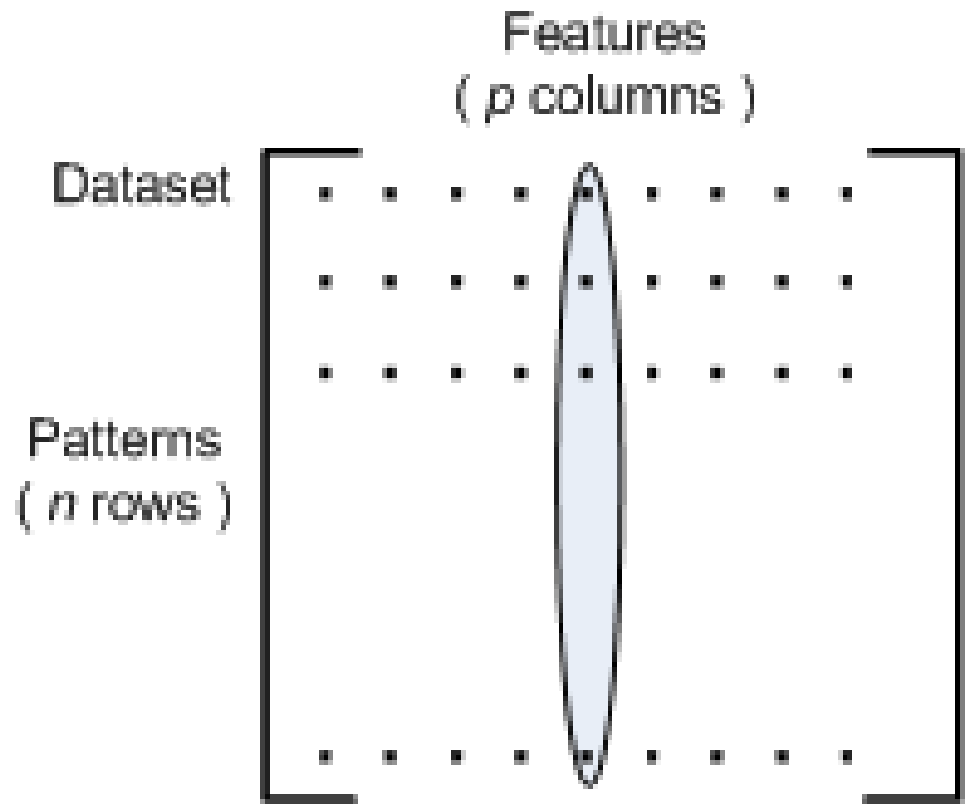
# 1.2 Sparse data

- In high dimensional spaces (large  $p$ ), many datasets have *sparse* features
- A sparse feature has a high occurrence of zeros
- The  $L_0$  norm of a vector is defined as the number of non-zero occurrences

Sparse Features



Small  $L_0$  norm in the columns of the dataset



## 1.2 Sparse data

*Sparse* data is commonly found in

- Biological datasets
- Gene expression datasets
- Text Categorization (TC) datasets
  - Reuters21578, RCV1, 20NewsGroups, ...
  
- Well-know problems on high-dimensional datasets:
  - “*large  $p$ , small  $n$* ”
  - “*curse of dimensionality*”

## 1.2 Sparse data

- TC datasets SpamBase, Example1, and Dexter
  - Spam Base – classify email as SPAM or not
  - Example1 and Dexter – subset of Reuters, classify if a text is about “corporate acquisitions” or not

Dataset	p	Subset	n	+1	-1	Avg. $L_0$	Avg. $L_0$ +1	Avg. $L_0$ -1
SpamBase	54	-----	4601	1813	2788	841.2	411.8	429.4
Example1	9947	Train	2000	1000	1000	9.5	4.5	5.0
		Test	600	300	300	2.4	1.1	1.3
Dexter	20000	Train	300	150	150	1.4	0.7	0.7
		Test	2000	1000	1000	9.6	-----	-----
		Validation	300	150	150	1.4	0.7	0.7

# 1.3 Text Categorization

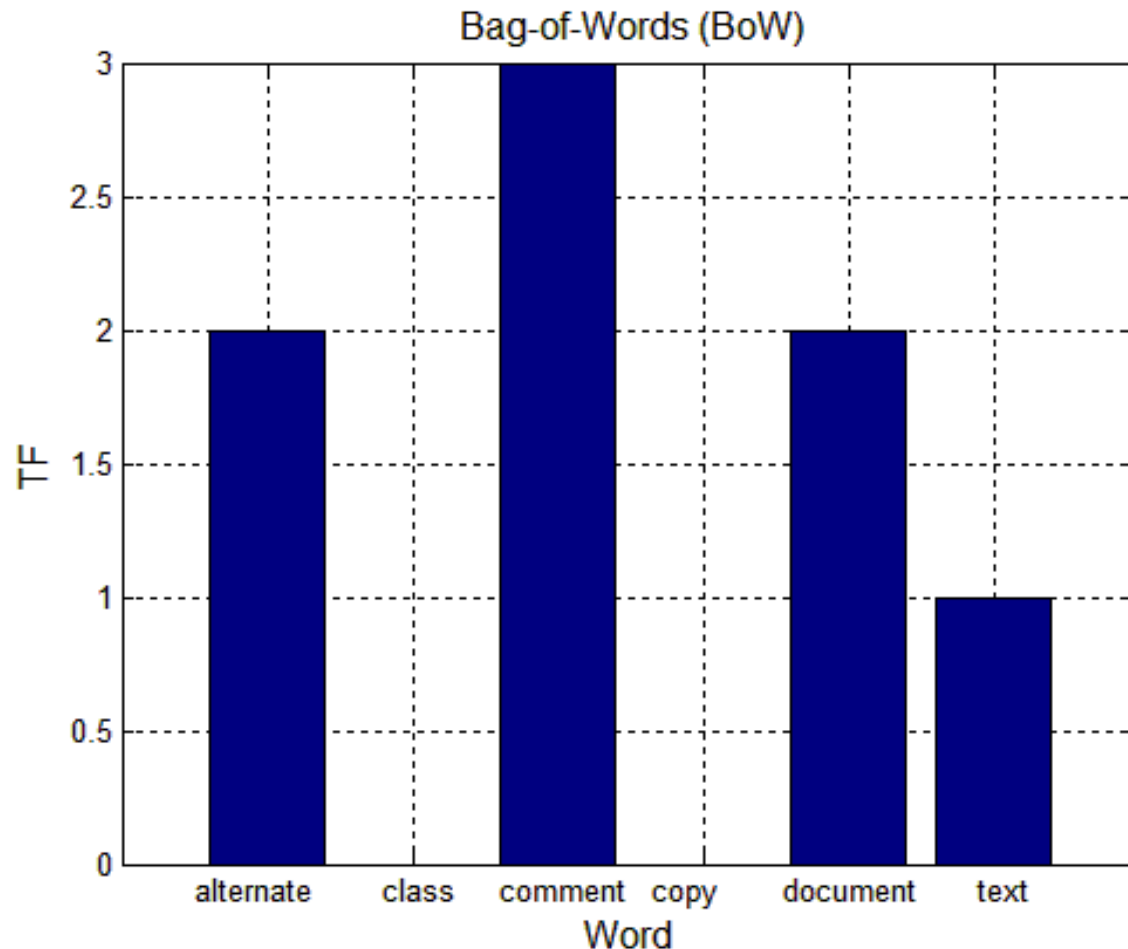
- Text categorization arises in many information retrieval problems
- Each text/document is assigned to one (or more):
  - *class*, in supervised or semi-supervised learning
  - *cluster*, in unsupervised learning
- For machine learning algorithms, each document is represented by a *Bag-of-Words* (BoW) vector

# 1.3 Text Categorization

- Bag-of-Words (BoW) is a high-dimensional feature vector
- These features:
  - represent the relative frequency of occurrence of a given word/term in each document
  - usually are stored as floating point values
  - usually are sparse; for a given document, many features are zero

# 1.3 Text Categorization

- Bag-of-Words (BoW) - toy example
- Contains some measure of terms in a document



Common Measures

**TF**  
Term-Frequency

**TF-IDF**  
Term-Frequency  
Inverse-Document-Frequency

# 1.3 Text Categorization

- A collection of documents is usually represented by a ***Term-Document*** (TD) matrix
  - columns/rows hold the BoW for each document
  - rows/columns correspond to the terms in the collection

**Each feature is usually a floating point value**

- An alternative representation is the binary ***Term-Document-Incidence*** (TDI) matrix
  - holds the information, for each document, if a given term is present or absent

**Binary Features !**

**Large collections imply large matrices !**

## 2.Feature Selection (FS)

- FS is a central problem in Machine Learning and Pattern Recognition: *to find the best subset of features for a given problem*
  - *How many features should we choose?*
  - *What features?*
- There are many approaches for supervised and unsupervised for **Feature Selection** (FS) and **Feature Reduction** (FR)
- Unsupervised approaches do not use class labels
- Supervised approaches use labels
- Semi-supervised approaches can use available labels

## 2.Feature Selection: benefits

Four well-known benefits of FS and FR techniques:

1. attains reduced dimension datasets
2. achieves lower memory requirements for dataset representation
3. attains a smaller training time for the machine learning method at hand (e.g. classifier)
4. improves the classification accuracy

All of these benefits are also shared by FD methods

There is by far much less research on FD than on FS

## 2.Feature Selection on TC

- For TC, several techniques have been proposed for FS and FR
  - A collection of documents occupies a large amount of memory!
  - It takes a long time to train/learn a classifier
- The majority of these techniques is applied directly on the floating-point BoW representations (the TD matrix)
- There are many supervised and unsupervised methods
- Supervised methods usually apply some information-theoretic criterion (e.g. *mutual information*)

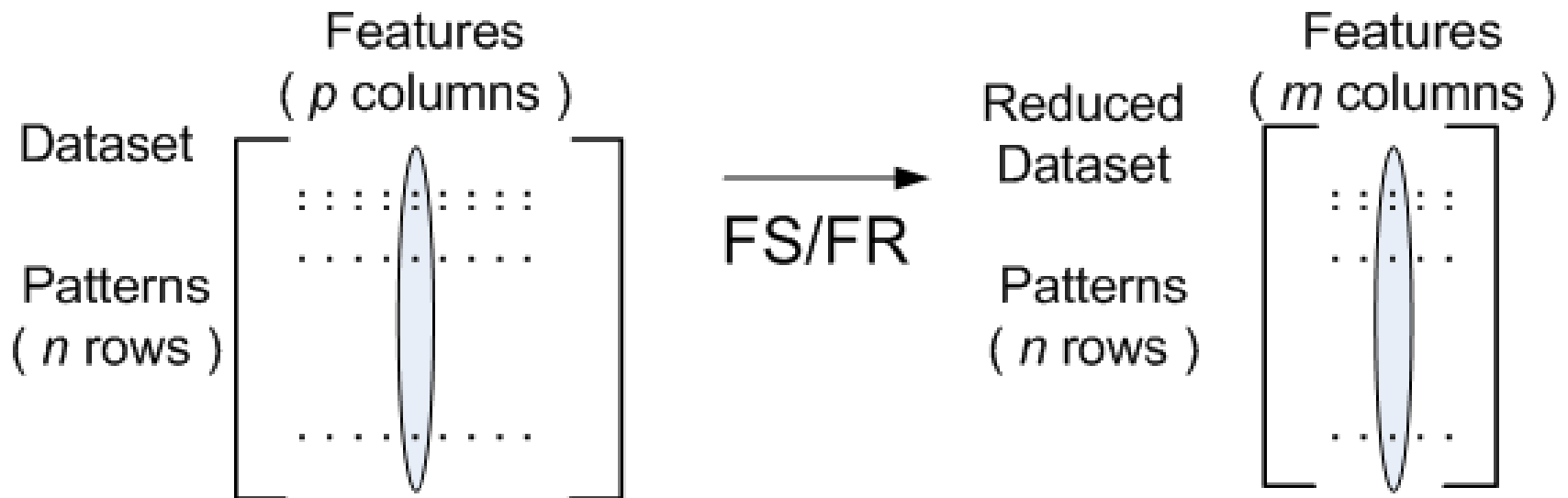
## 2.1 Key Issues on FS

- Two key issues for FS: *relevance* and *redundancy*
- **Relevance** is computed with some criterion
  - Measures how important/discriminative a given feature is
  - Supervised methods use the class label to computing relevance
  - Unsupervised methods can only look at the values of the feature
- **Redundancy** is a measure of dependency (common information) between features
  - Redundant features must be identified (and deleted), even if they have high relevance

## 2.2 Filter approach to FS

### Filter approach for FS/FR

Select  $m$  ( $< p$ ) features from the set of patterns



## 2.2 Filter approach to FS

### Advantages

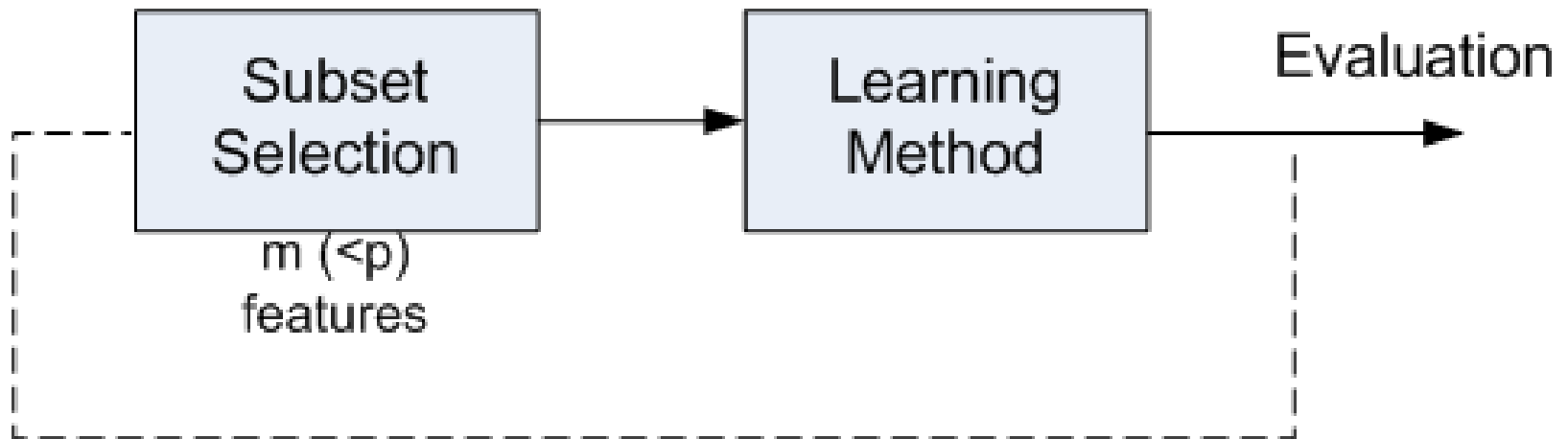
- Uses some statistical measure
- Assigns a rank to each feature
- Fast and efficient
- *Agnostic* – independent of the learning algorithm

### Drawbacks

- Measures only the *relevance* of each feature
- Ignores the learning bias of the learning method
- Selected features can be correlated among themselves

## 2.2 Wrapper approach

- Learning method is trained for each candidate subset
- Usually finds *better* features than the filter approach
- Computationally expensive



- Evolutionary and parallel genetic algorithms have been used ( small and medium  $p$  )
- Drawback: hard to apply directly to high-dimensional datasets (**processing time!**)

## 2.3 Some supervised FS methods

- Many methods rely on information theory measures, like *mutual information* and *entropy* between features and class label
- Some common methods:
  1. Fishers Ratio (FiR)
  2. mrMR - Minimum Redundancy Maximum Relevance
  3. Branch and Bound, Pudil's Method
  4. Forward Selection, Backward Selection, ...
  5. ....
  6. MIM – Mutual Information Maximization
  7. CMIM – Conditional Mutual Information Maximization
  8. FIRM - Feature Importance Ranking Measure

## 2.3 Some supervised FS methods

- **Fishers Ratio (FiR)**

$\mu_i^{(\pm 1)}$  and  $\text{var}_i^{(\pm 1)}$

$$FiR_i = \frac{|\mu_i^{(-1)} - \mu_i^{(+1)}|}{\sqrt{\text{var}_i^{(-1)} + \text{var}_i^{(+1)}}}$$

are the mean and variance of feature  $i$

- **mrMR - Minimum Redundancy Maximum Relevance**

- *Relevance* is the mutual information (MI) between features and class labels
- *Redundancy* is the MI between features

## 2.3 Some supervised FS methods

**Binary Features only (TDI matrices)**

### ■ **MIM - Mutual Information Maximization**

- Chooses (binary) features such that maximize MI with class label

### ■ **CMIM – Conditional Mutual Information Maximization**

Chooses (binary) features such that maximize both:

- MI with class label
- the conditional entropy with the previous selected features

### ■ **FIRM - Feature Importance Ranking Measure**

Takes the underlying correlation structure of the features into account as well as its prior class probabilities

## 2.4 Some unsupervised FS methods

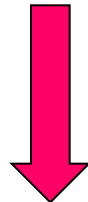
There are many methods; we address only three of them

1. **Term-Variance (TV)** – computes the variance of each feature; relevance = variance
2. **Random Subspaces (RS)**
  1. (pseudo)randomly selects a subset of components of the feature vector
  2. this procedure is repeated many times and the *q chosen feature subsets* are combined into a final list of selected features, to train some classifier
3. **Random Projections (RP)** [Feature Reduction]
  1. Compressed Learning theory - 2009

## 2.4 Some unsupervised FS methods

### ■ Random Projections (RP)

- Let  $\mathbf{A}$  be an  $m \times p$  random matrix, with  $m < p$
- Let  $\mathbf{x}$  be the feature (BoW) vector
- Then  $\mathbf{y} = \mathbf{A}\mathbf{x}$  is a reduced (BoW) vector



The entries of  $\mathbf{A}$  are randomly generated

The following distributions yield good RP matrices  $\mathbf{A}$ :

1. **Gaussian**  $N(0, 1/\sqrt{m})$
2. **Bernoulli** over  $\pm 1/\sqrt{m}$  with equal probability
3. **Achlioptas** probability mass function  $\{1/6, 2/3, 1/6\}$  over  $\{-\sqrt{3/m}, 0, \sqrt{3/m}\}$
4. **Li et al.** probability mass function  $\{1/(2s), 1 - 1/s, 1/(2s)\}$  over  $\{-\sqrt{s/m}, 0, \sqrt{s/m}\}$  with  $s = n$  or  $s = \log(n)$

**Achlioptas and Li distributions lead to sparse matrices**

## 2.4 Compressed Learning

- A *good*  $m \times n$  RP matrix  $\mathbf{A}$  must satisfy the  $(k, \varepsilon)$  **RIP** – **Restricted Isometry Property** if for any  $k$ -sparse vector  $x$  (up to  $k$  non-zeros) obeys

$$(1 - \varepsilon) \|x\|^2 \leq \|Ax\|^2 \leq (1 + \varepsilon) \|x\|^2 \quad \text{RIP}$$

This happens for small  $\varepsilon$ , with overwhelming probability, if

$$m = \Omega\left(k \log\left(\frac{p}{k}\right)\right)$$

Provides a good estimate for the number of reduced dimensions  $m$

Similar to the Johnson-Lindenstrauss lemma

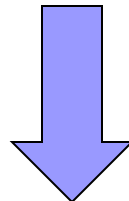
## 2.4 Compressed Learning

- The **generalized RIP (GRIP)** gives conditions under which the inner products are approximately preserved
- A satisfies  $(2k, \varepsilon)$ -RIP

**GRIP**

$$(1 + \varepsilon) \mathbf{x}^T \mathbf{x}' - 2R^2 \varepsilon \leq \mathbf{y}^T \mathbf{y}' \leq (1 - \varepsilon) \mathbf{x}^T \mathbf{x}' + 2R^2 \varepsilon$$

If the training patterns are  $k$ -sparse and  $A$  satisfies the  $(2k, \varepsilon)$ -RIP, a linear SVM learnt from the compressed patterns  $y$  will be very similar to one obtained from the original patterns  $x$



**Remark: A linear SVM classifier depends only on the inner products between input patterns!**

## 2.4 Compressed Learning

### Compressed Learning Theorem

A SVM learned from the compressed data is never much worse than the best linear classifier in the original high dimensional space

*R. Calderbank, S. Jafarpour, and R. Schapire. "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain", 2009 [Online] <http://dsp.rice.edu/files/cs/cl.pdf>*

# 3. Feature Discretization (FD)

- FD has been addressed in the literature
- There is by far much less research on FD than on FS
- The proposed techniques are based on scalar quantization
  - Avoid the use of vector quantization; complexity!
- Unsupervised approaches
  1. Equal-Interval Binning (EIB) - uniform quantization with a given number of bits for each feature
  2. Equal-Frequency Binning (EFB) - non-uniform quantization
    - yields intervals such that for each feature the number of occurrences in each interval is the same
    - also known as maximum entropy quantizer

# 4. Analysis of FS methods

- We use a *filter* approach:
  - these methods are not tied to the type of classifier to be used (they are *agnostic*)
  - we compute solely *relevance*
  
- Proposed methods (multi-class for standard and binary features):
  - $L_0$  norm
  - AD - Absolute Difference
  - AMGM – Arithmetic Mean Geometric Mean
  - $L_0$  norm - supervised variant

## 4. Analysis of FS methods

### $L_0$ norm (unsupervised)

- Input:  $p \times n$  TD (or TDI) matrix  $\mathbf{X}$   
 $m (< p)$  the desired maximum number of features
  - Output: Reduced training set and test set
- 

1. Compute the  $l_0$  norm of each feature
2. Remove non-informative features with  $l_0 = 0$  or  $l_0 = n$
3. Let  $s$  be the number of remaining features
4. If  $s < m$ , then stop, otherwise proceed to step 5
5. Keep only the  $m$  features with **largest  $l_0$  norm**

# 4. Analysis of FS methods

## Dispersion Measures

### AD – Absolute Difference (unsupervised)

Keeps up to  $m$  features such that maximize the ranking

$$AD_i = \sum_{j=1}^n |X_{ij} - \mu_i|$$

### AMGM – Arithmetic Mean Geometric Mean (unsupervised)

$$AM_i = \frac{1}{n} \sum_{j=1}^n \exp(X_{ij}) \qquad GM_i = \left( \prod_{j=1}^n \exp(X_{ij}) \right)^{\frac{1}{n}}$$

AM - Arithmetic Mean

GM – Geometric Mean

$$AMGM_i = AM_i / GM_i$$

The exponential function avoids the zero division problem

## 4. Analysis of FS methods

### $L_0$ norm (supervised)

- A binary feature is as much informative as the difference between its  $l_0$  norms for each of the classes
- Let  $l_0^{(i,-1)}$  and  $l_0^{(i,+1)}$  be the  $l_0$  norm of feature  $i$ , for patterns of class  $-1$  and  $+1$ , respectively
- We rank feature  $i$  with  $r_i = |l_0^{(i,-1)} - l_0^{(i,+1)}|$
- *Always Zero* and *Always Present* features have zero rank

# 4. Analysis of FS methods

## $L_0$ norm (supervised)

- Input:  $p \times n$  TD (or TDI) matrix  $\mathbf{X}$   
 $m (< p)$  the desired maximum number of features  
class label for each pattern
  - Output: Reduced training set and test set
- 

1. Compute the  $l_0$  norm of each feature
2. Remove non-informative features with  $l_0 = 0$  or  $l_0 = n$
3. Let  $s$  be the number of remaining features
4. If  $s < m$ , then stop, otherwise proceed to step 5
5. Compute the rank of each feature  $r_i = |l_0^{(i,-1)} - l_0^{(i,+1)}|$
6. Keep only the  $m$  features with largest ranks  $r_i$

# 4. Analysis of FS methods

## $L_0$ norm (supervised) extension to K classes

- Straightforward generalization
  - Perform the same actions (steps) as in the previous slide
- Instead of binary rank measure

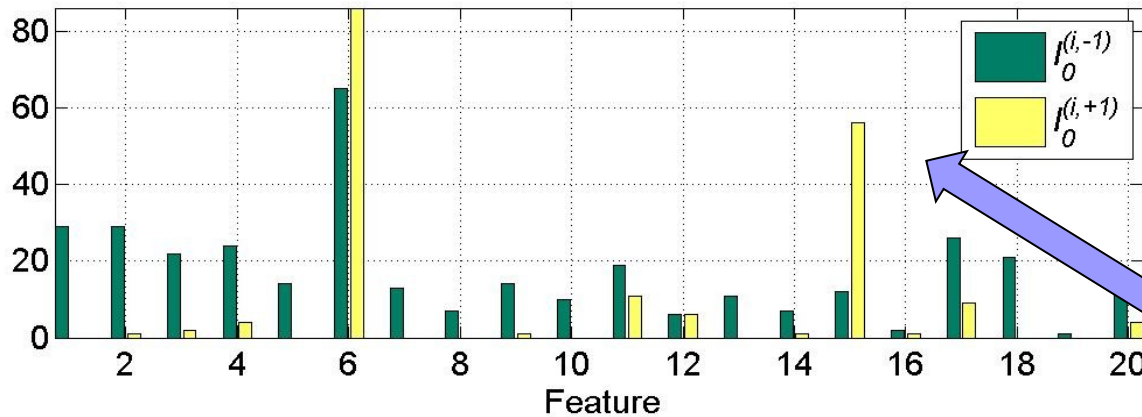
$$r_i = | l_0^{(i,-1)} - l_0^{(i,+1)} |$$

- Use the multi-class rank measure

$$r_i = \sum_{l=1}^K \sum_{k=1}^K | l_0^{(i,l)} - l_0^{(i,k)} |$$

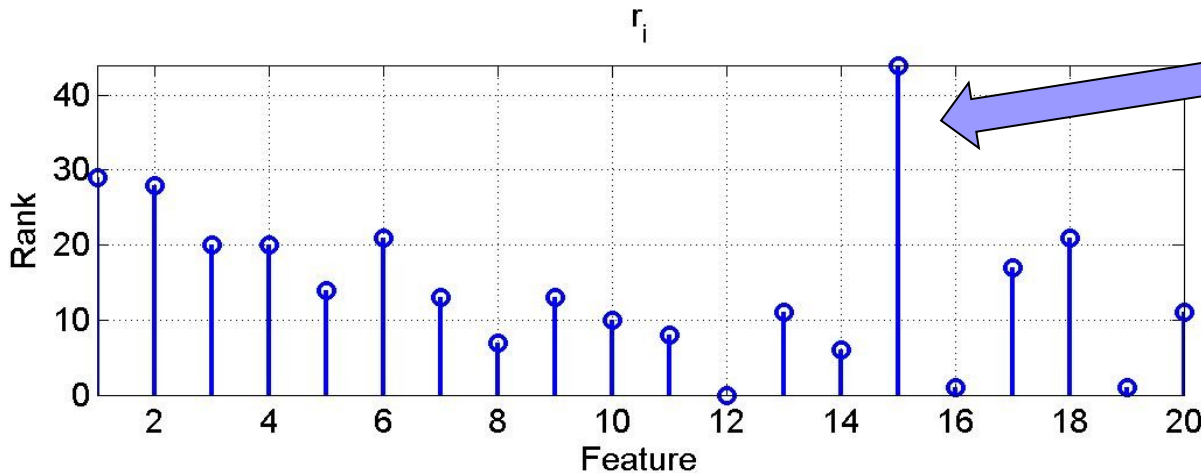
# 4. Analysis of FS methods

$$f_0^{(i,-1)}, f_0^{(i,+1)}$$



**L<sub>0</sub> norm  
(supervised)  
features  
and  
ranks**

**Feature 15  
has larger  
rank than  
feature 6**

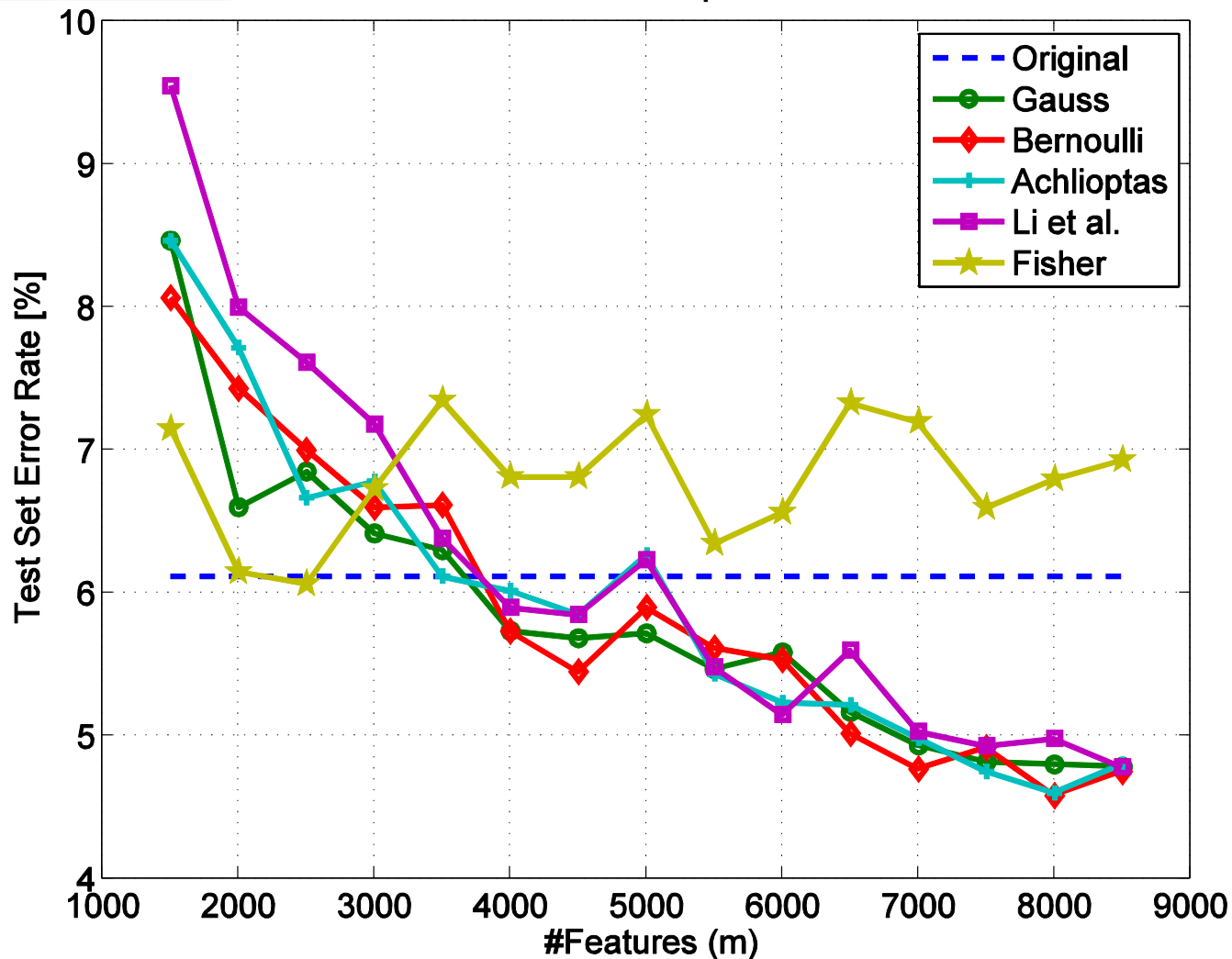


Feature	Rank	SVM Test Error	AdaBoost Test Error
6	21	34.6 %	41 %
15	44	33.5 %	34 %

# 4.1 Experimental Results

Dataset	k	$m_R$	$m_G$
Example 1	47.1	253	436

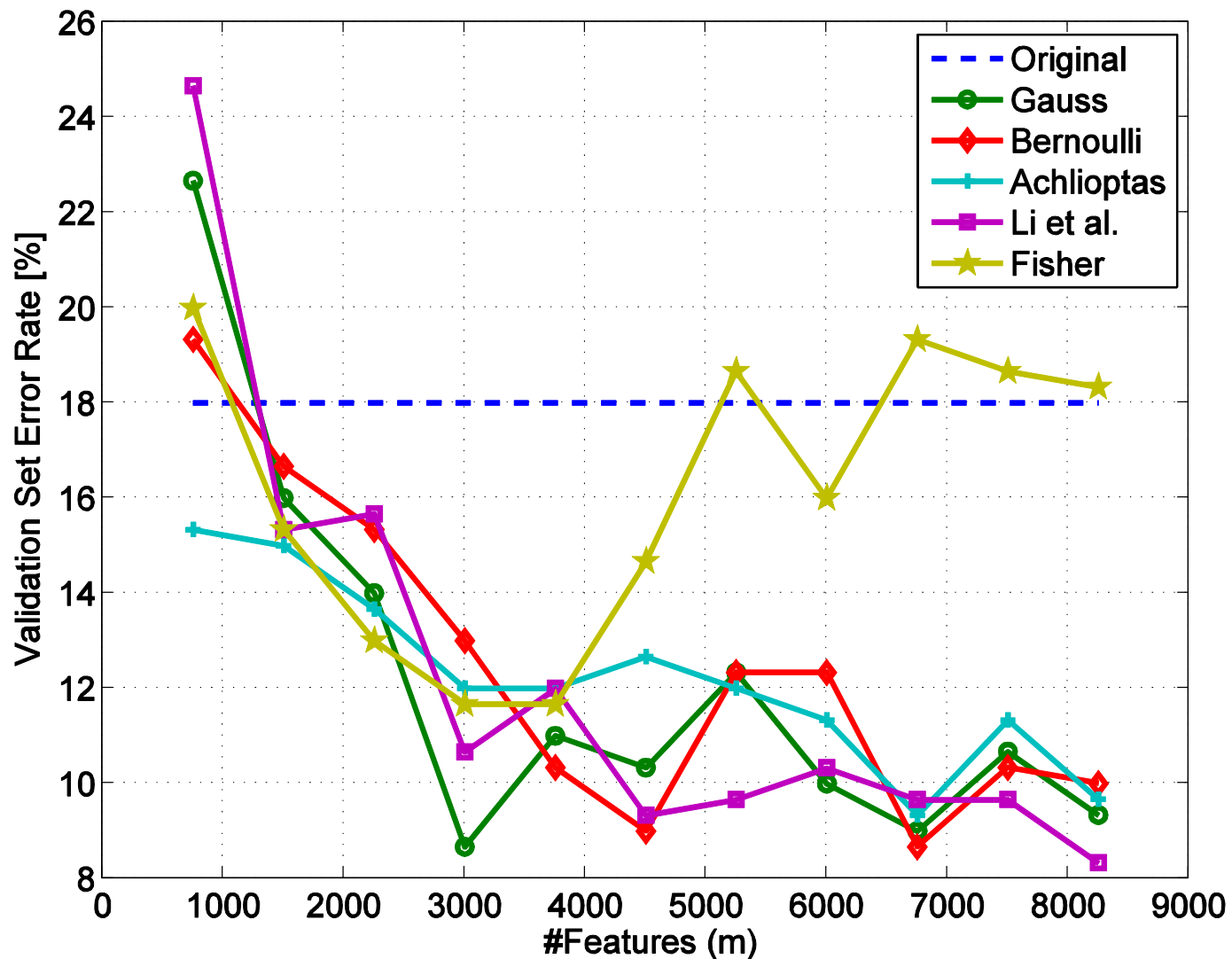
Test Set Error Rate on Example 1 with Linear SVM



# 4.1 Experimental Results

Dataset	k	$m_R$	$m_G$
Dexter	94.1	505	878

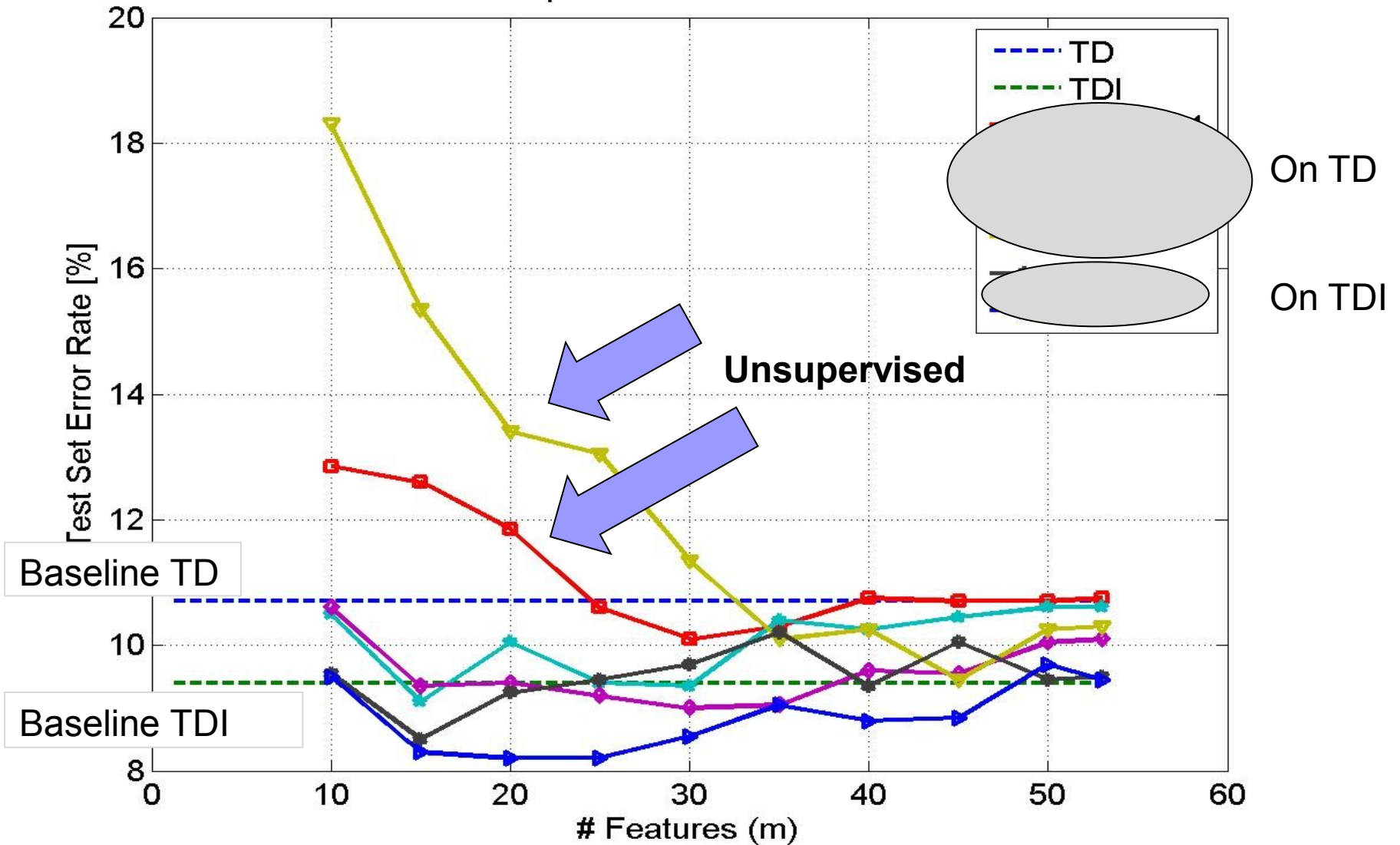
Validation Set Error Rate on Dexter with Linear SVM



# 4.1 Experimental Results

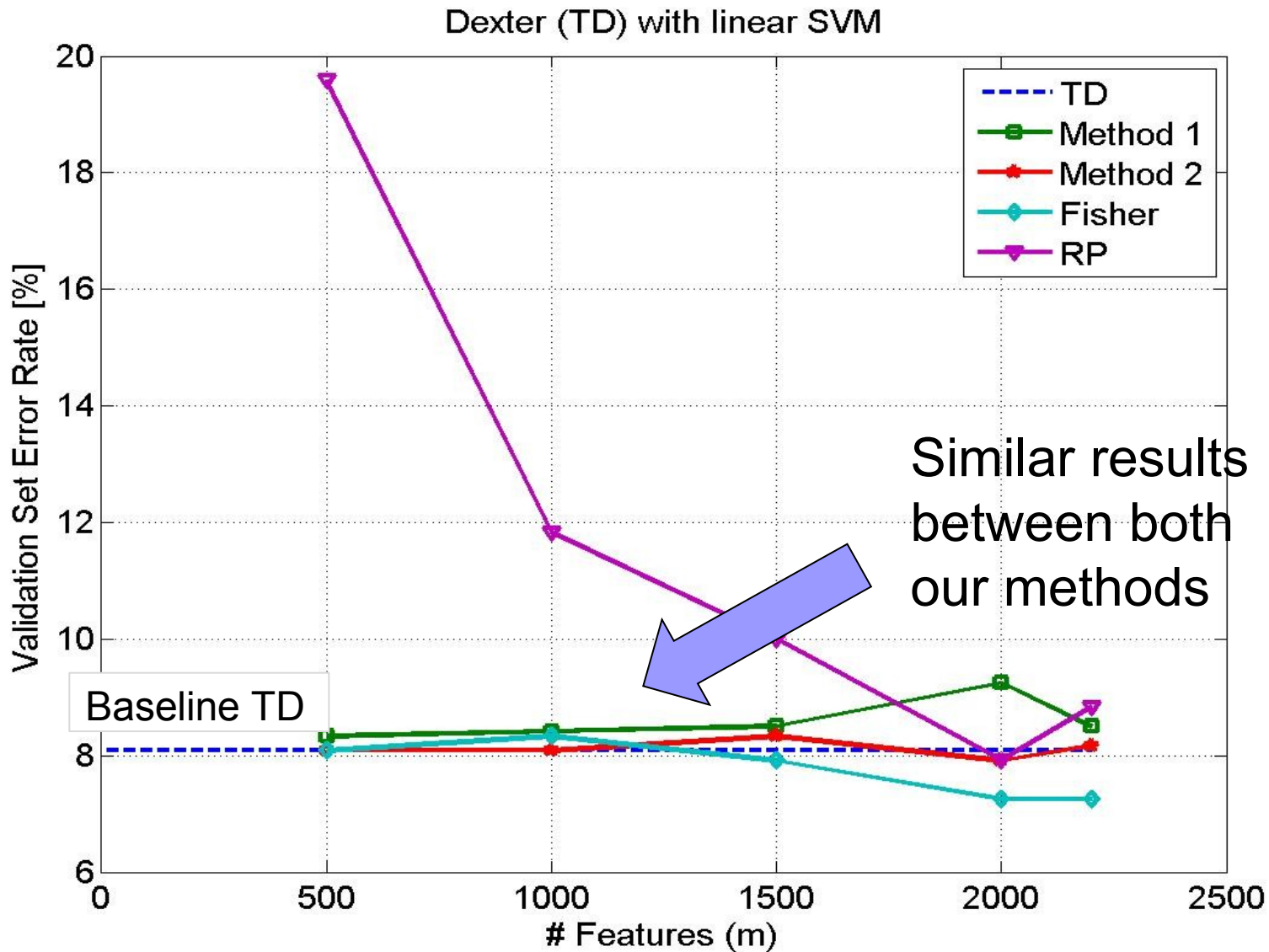
Method 1 =  $L_0$  unsupervised  
Method 2 =  $L_0$  supervised

Spam with linear SVM



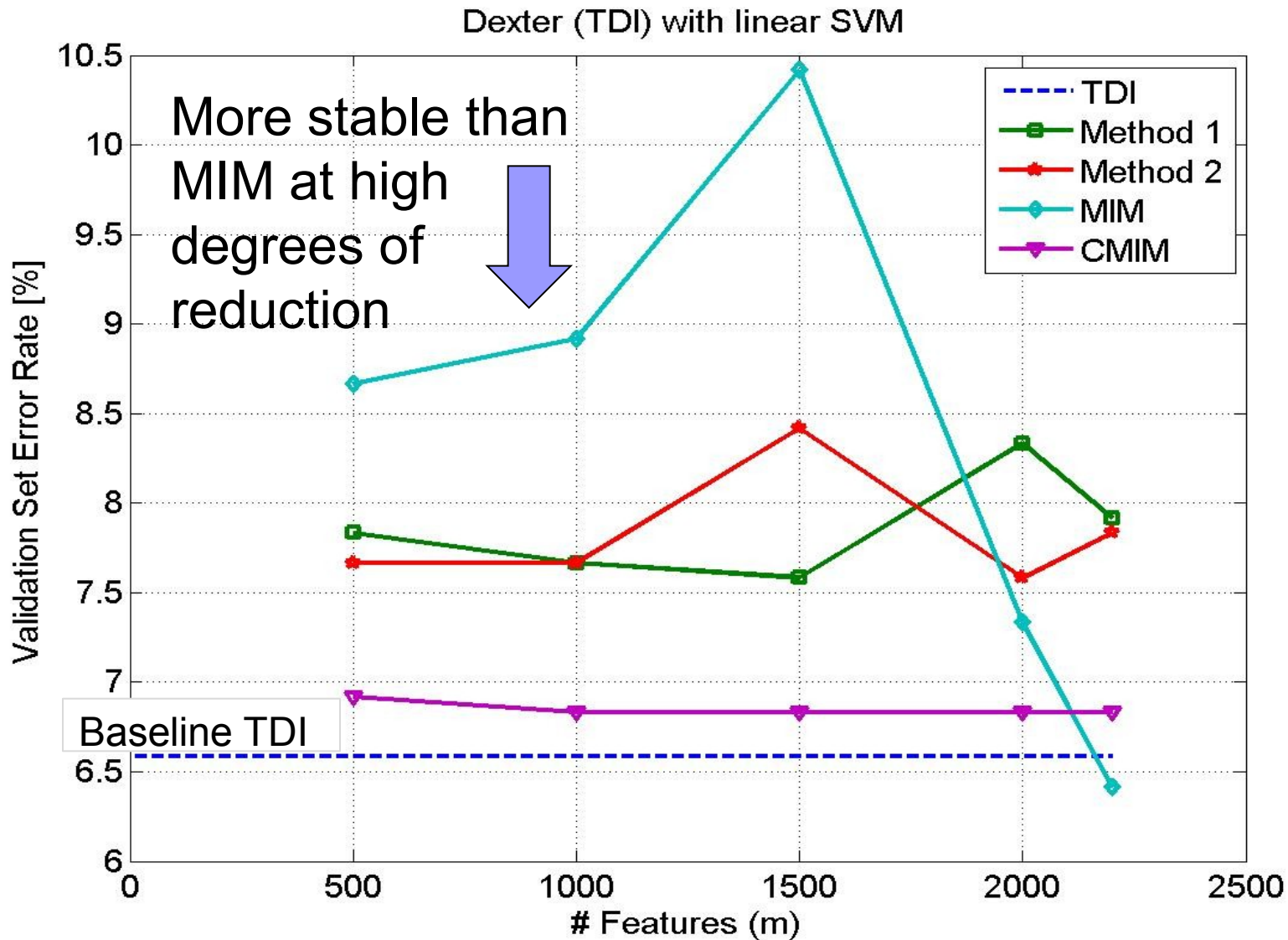
# 4.1 Experimental Results

With TD and its FS versions



# 4.1 Experimental Results

With TDI and its FS versions



## 4.1 Experimental Results: Discussion

- ❑ The four types of RP matrices significantly reduce the number of features, improving the classification accuracy
- ❑ The number of reduced dimensions is computed by a sparsity analysis of the training data, regardless of the class label
- ❑ The reduced features obtained by (sparse) RP are adequate for sparse BoW-like representations
  - ❑ lower test set error rate
  - ❑ the method is fully unsupervised

# 4.1 Experimental Results: Discussion

- The proposed methods have results
  - similar or better than the supervised Fisher Ratio
  - better than RP with TD matrix
- For TDI matrices (binary BoW), under some conditions, we get better and more stable results than supervised Mutual Information Maximization (MIM)
- Our results are close to those obtained by the Conditional MIM method (which is more complex)
- For high-dimensional datasets, our methods attain dimension reduction of the order of 40

## 4.1 Experimental Results: Discussion

- *Artur Ferreira and Mário Figueiredo, “**Unsupervised Feature Selection for Sparse Data**”, ESANN’2011, April 2011*
- *Artur Ferreira and Mário Figueiredo, “**Efficient Unsupervised Feature Selection for Sparse Data**”, EuroCon-2011/ConfTele-2011, April 2011*
- *Artur Ferreira and Mário Figueiredo, “**Feature Transformation and Reduction for Text Classification**”, PRIS 2010, pp 72-81, Funchal, Portugal, June 2010*

## 5. Analysis of FD and FD+FS methods

1. Equal-Interval Binning (EIB) performs poorly
2. Equal-Frequency Binning (EFB) performs much better than EIB
3. We proposed a FD procedure based on the Lloyd-Max scalar quantization algorithm

*Artur Ferreira and Mário Figueiredo, “**Unsupervised Joint Feature Discretization and Selection**”, IbPRIA’2011, June 2011.*

# 5. Analysis of FD and FD+FS methods

## Our FD procedure (scalar feature quantization)

1. The algorithm runs for a given target distortion  $D$  in a Mean Square Error (MSE) sense and a maximum number of bits  $q$
2. The Lloyd-Max procedure is applied individually to each feature using the pair  $(D, q)$  as the stopping condition
3. The procedure stops when
  - distortion  $D$  is achieved, or
  - the maximum number of bits  $q$  per feature is reached

# 5. Analysis of FD and FD+FS methods

## Our combined FD + FS procedure

Unsupervised FS (UFS) step with a filter approach, on the discretized features. Ranking of feature  $i$  given by

$$r_i = \text{var}(X_i) / b_i$$

where  $b_i \leq q$  is the number of bits allocated to feature  $i$  in the FD step;  $\text{var}(X_i)$  is the variance of the original (non-discretized) feature

Key ideas:

- features with higher variance are more informative
- for a given variance, features quantized with a smaller number of bits are preferable

# 5.1 Experimental Results (on FD)

Minimum, maximum, average bits allocated using EFB and our FD step, up to  $q=16$  bits. EFB stops when the discretized feature entropy exceeds 99% of its maximum value. FD step uses  $D=0.01\text{var}(X_i)$

Dataset	EFB				Our FD step			
	Min	Max	Avg	Mem	Min	Max	Avg	Mem
Example1	2	8	7.99	6.033.300	2	5	2.19	<u>1.654.900</u>
Dexter	2	16	15.98	26.742.300	2	4	2.12	<u>3.542.825</u>
SpamBase	2	16	15.74	479.655	2	4	3.23	<u>98.347</u>
Ionosphere	2	16	7.52	10.881	2	4	3.64	<u>5.265</u>
WDBC	2	2	2.00	<u>4.268</u>	4	5	4.17	8.891
Wine	2	16	3.08	<u>890</u>	4	4	4.00	1.157

# 5.1 Experimental Results (on FD)

Amount of memory (bytes) to represent the datasets

Dataset	Original	EFB	Our FD step
Example1	98.600.000	6.033.300	<u>1.654.900</u>
Dexter	198.300.000	26.742.300	<u>3.542.825</u>
SpamBase	947.800	479.655	<u>98.347</u>
Ionosphere	45.500	10.881	<u>5.265</u>
WDBC	65.100	<u>4.268</u>	8.891
Wine	8.800	<u>890</u>	1.157

# 5.1 Experimental Results (on FD)

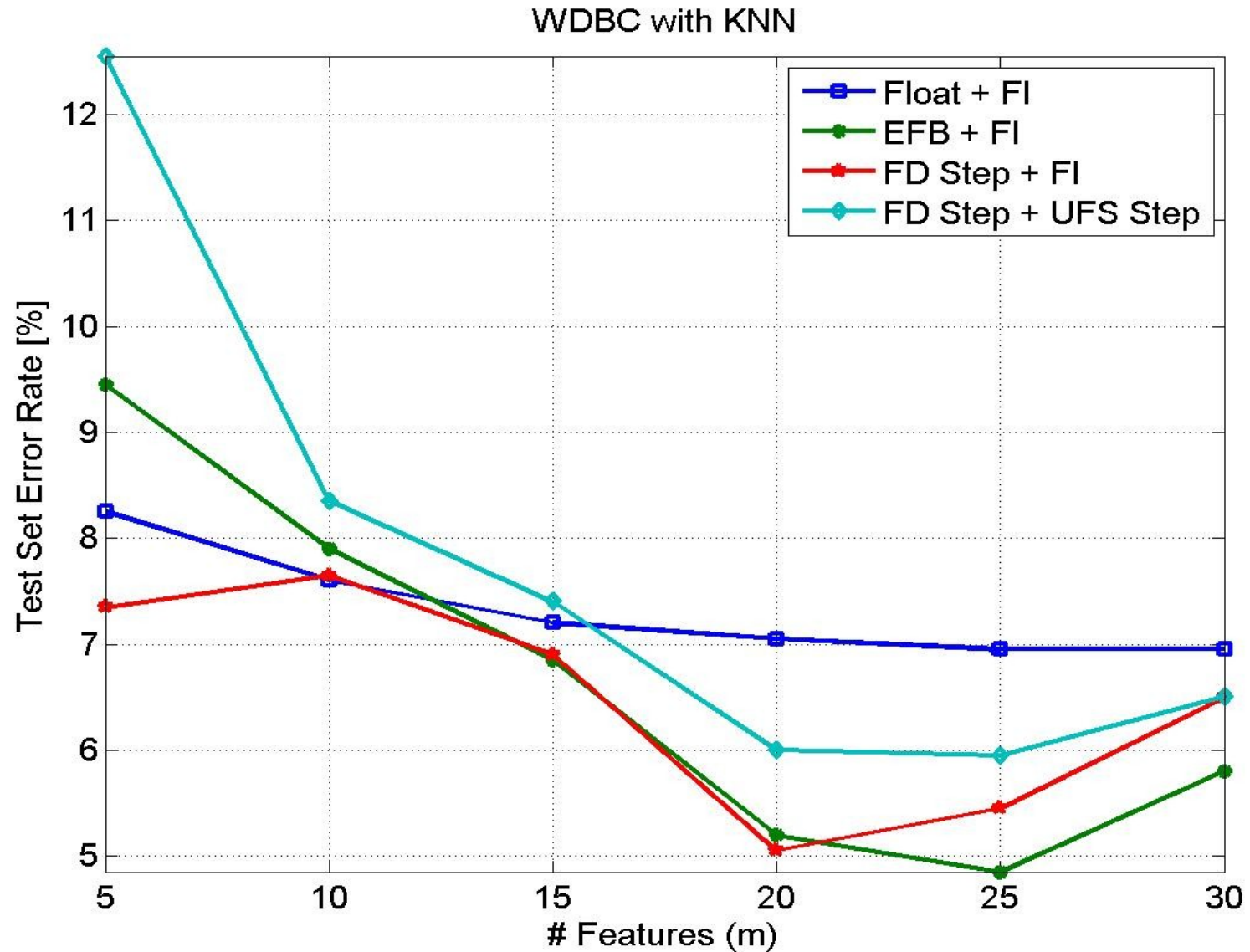
Test set error rate (%) (average of 10 runs) for the Ionosphere, WDBC, and Wine datasets, without FS using:

- linear SVM
- Naïve Bayes (NB)
- K-Nearest Neighbors (KNN) with K=3 classifiers

For each dataset and each classifier, the best result is underlined

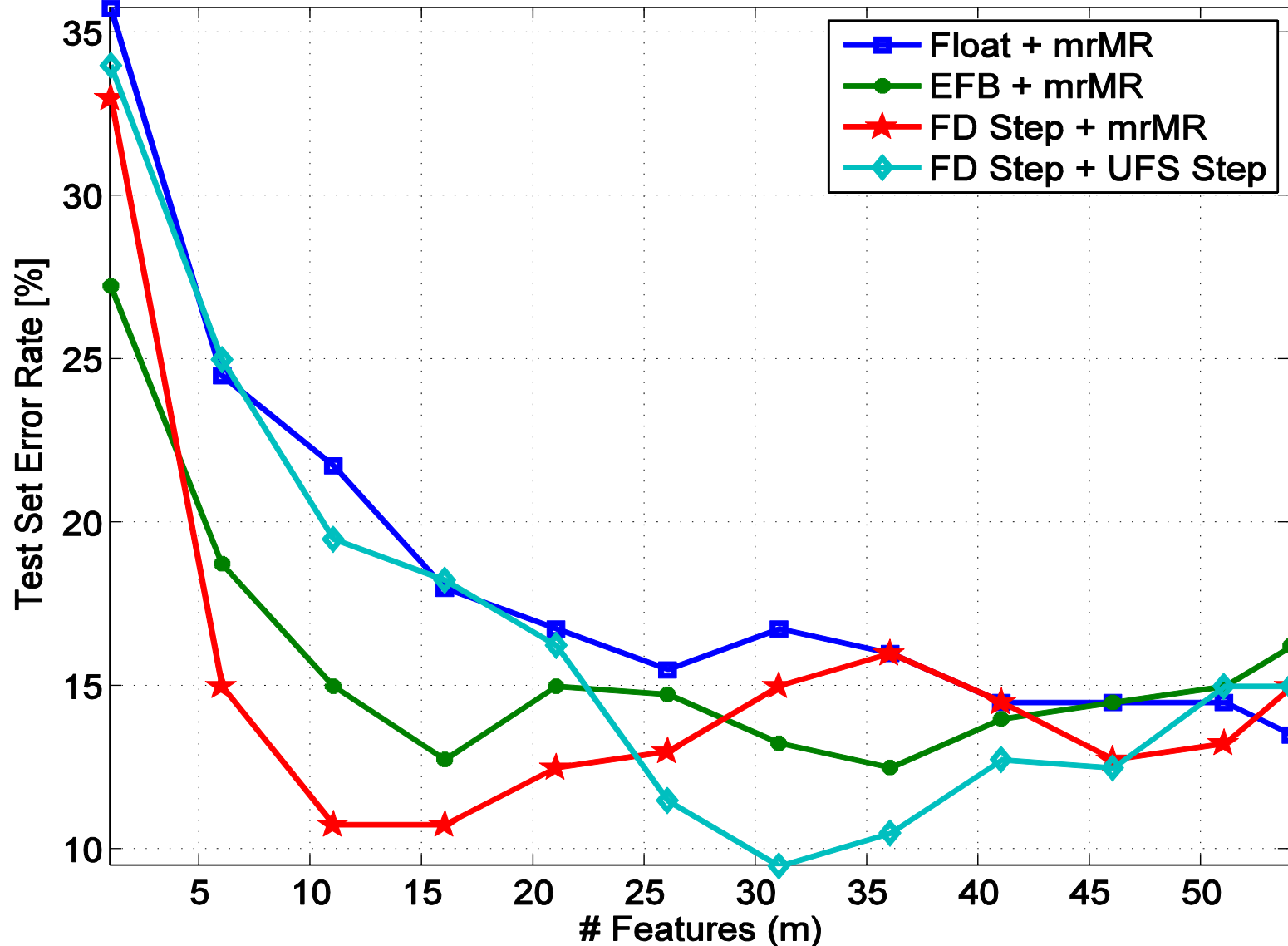
Representation	Ionosphere			WDBC			Wine		
	SVM	NB	KNN	SVM	NB	KNN	SVM	NB	KNN
Original	<u>9.95</u>	<u>12.94</u>	<u>11.94</u>	4.80	<u>6.60</u>	8.40	1.51	1.24	12.62
EFB	16.92	30.35	17.41	4.20	6.80	6.00	1.86	1.95	11.37
FD step	13.43	18.91	16.42	<u>3.60</u>	6.80	<u>4.80</u>	<u>0.98</u>	<u>1.06</u>	<u>2.4</u>

# 5.1 Experimental Results (on FD + FS)



# 5.1 Experimental Results (on FD + FS)

SpamBase with linear SVM



# 6. Concluding Remarks

- Feature selection, reduction, and discretization are open problems (for sparse data)
- There is no method that clearly outperforms all the others
  - Depends on the learning problem
  - Depends on the statistical properties of each feature
- On sparse data, unsupervised feature selection can be done efficiently with dispersion measures
- Information theoretic methods do not perform so well on sparse data as they do on dense data

# 6. Concluding Remarks

- For text classification TDI matrices (1 bit per feature) are adequate
  - fast training
  - good experimental results
- The Loyd-Max discretization method (FD step) usually allocates a small number of bits per feature
  - attains efficient dataset representation and large memory savings
  - leverages feature selection methods
- The joint use of feature discretization and selection method is adequate for sparse data and non-sparse data

# References

[1] UCI Machine Learning Repository

<http://archive.ics.uci.edu/ml/datasets>

[2] NIPS 2003 FS Challenge datasets

<http://www.nipsfsc.ecs.soton.ac.uk>

[3] PRTools: The Matlab Toolbox for Pattern Recognition

<http://www.prtools.org/>

[4] The ENTOOL Matlab Toolbox

<http://zti.if.uj.edu.pl/~merkwith/entool.htm>

# References

- [5] F. Fleuret and I. Guyon. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004 (**MIM and CMIM**)
- [6] D. Achlioptas. Database-friendly random projections. In *ACM Symposium on Principles of Database Systems*, pages 274–281, Santa Barbara, USA, 2001 (**RP**)
- [7] P. Li, T. Hastie, and K. Church. Very sparse random projections. In *KDD '06: Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 287–296, Philadelphia, USA, 2006. (**RP**)
- [8] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *KDD'01: Proc. of the 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 245–250, San Francisco, USA, 2001. (**RP**)

# References

- [9] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8) (August 2005) 1226–1238 (**mrMR**)
- [10] A. Zien, N. Kr"amer, S. Sonnenburg, G. R"atsch, The feature importance ranking measure. *Proc. of the European Conference on Machine Learning PKDD*, Springer, 5782 (2009) 694–709 (**FIRM**)
- [11] T. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998. (**RSM**)
- [12] C. Lai, M. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," *Pattern Recognition Letters*, vol. 27, no. 10, pp. 1067–1076, 2006. (**RSM**)

# References

- [13] L. Liu, J. Kang, J. Yu, and Z. Wang, “A comparative study on unsupervised feature selection methods for text clustering,” in *IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2005*, pp. 597–601. **(TV)**
- [14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, 2nd ed.* Springer, 2009. **(FS)**
- [15] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh (Editors), *Feature Extraction, Foundations and Applications.* Springer, 2006. **(FS)**
- [16] F. Escolano, P. Suau, and B. Bonev, *Information Theory in Computer Vision and Pattern Recognition.* Springer, 2009. **(FS)**
- [17] T. Joachims, *Learning to Classify Text Using Support Vector Machines.* Kluwer Academic Publishers, 2001. **(TC)**

# References

- [18] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. **(TC)**
- [19] G. Forman, “An extensive empirical study of feature selection metrics for text classification,” *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 2003. **(FS)**
- [20] K. Hyunsoo, P. Howland, and H. Park, “Dimension reduction in text classification with support vector machines,” *Journal of Machine Learning Research*, vol. 6, pp. 37–53, 2005. **(FS)**
- [21] K. Torkkola, “Discriminative features for text document classification,” *Pattern Analysis and Applications*, vol. 6, no. 4, pp. 301–308, 2003. **(FS)**