Unsupervised feature discretization and selection for sparse data

Priberam Machine Learning Lunch Seminar
Lisbon, 1st March 2011

Artur Ferreira
(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)

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>p</th>
<th>Subset</th>
<th>n</th>
<th>+1</th>
<th>-1</th>
<th>Avg. L₀</th>
<th>Avg. L₀+1</th>
<th>Avg. L₀-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpamBase</td>
<td>54</td>
<td>---------</td>
<td>4601</td>
<td>1813</td>
<td>2788</td>
<td>841.2</td>
<td>411.8</td>
<td>429.4</td>
</tr>
<tr>
<td>Example1</td>
<td>9947</td>
<td>Train</td>
<td>2000</td>
<td>1000</td>
<td>1000</td>
<td>9.5</td>
<td>4.5</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>600</td>
<td>300</td>
<td>300</td>
<td>2.4</td>
<td>1.1</td>
<td>1.3</td>
</tr>
<tr>
<td>Dexter</td>
<td>20000</td>
<td>Train</td>
<td>300</td>
<td>150</td>
<td>150</td>
<td>1.4</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>2000</td>
<td>1000</td>
<td>1000</td>
<td>9.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Validation</td>
<td>300</td>
<td>150</td>
<td>150</td>
<td>1.4</td>
<td>0.7</td>
<td>0.7</td>
</tr>
</tbody>
</table>
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

![Graph showing Bag-of-Words (BoW)]

**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)**
  \[
  FiR_i = \frac{\left| \mu_i^{(-1)} - \mu_i^{(+1)} \right|}{\sqrt{\text{var}_i^{(-1)} + \text{var}_i^{(+1)}}}
  \]

  \(\mu_i^{(\pm 1)}\) and \(\text{var}_i^{(\pm 1)}\)
  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)**
   1. Compressed Learning theory - 2009
2.4 Some unsupervised FS methods

- **Random Projections (RP)**
  - Let $A$ be an $m \times p$ random matrix, with $m < p$
  - Let $x$ be the feature (BoW) vector
  - Then $y = Ax$ is a reduced (BoW) vector

  The entries of $A$ are randomly generated

The following distributions yield good RP matrices $A$:

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 $A$ must satisfy the $(k, \varepsilon)$ RIP – Restricted Isometry Property if for any $k$-sparse vector $x$ (up to $k$ non-zeros) obeys

$$\left(1 - \varepsilon\right) \|x\|^2 \leq \|Ax\|^2 \leq \left(1 + \varepsilon\right) \|x\|^2$$

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

\[
(1 + \varepsilon) x^T x' - 2R^2 \varepsilon \leq y^T y' \leq (1 - \varepsilon) x^T x' + 2R^2 \varepsilon
\]

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

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 $X$
  
  $m (< p)$ 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

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)

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

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

L₀ 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
- Use the multi-class rank measure

\[ r_i = \left| l_0^{(i,-1)} - l_0^{(i,+1)} \right| \]

\[ r_i = \sum_{l=1}^{K} \sum_{k=1}^{K} \left| l_0^{(i,l)} - l_0^{(i,k)} \right| \]
4. Analysis of FS methods

$L_0$ norm (supervised) features and ranks

Feature 15 has larger rank than feature 6

<table>
<thead>
<tr>
<th>Feature</th>
<th>Rank</th>
<th>SVM Test Error</th>
<th>AdaBoost Test Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>21</td>
<td>34.6 %</td>
<td>41 %</td>
</tr>
<tr>
<td>15</td>
<td>44</td>
<td>33.5 %</td>
<td>34 %</td>
</tr>
</tbody>
</table>
4.1 Experimental Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$k$</th>
<th>$m_R$</th>
<th>$m_G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>47.1</td>
<td>253</td>
<td>436</td>
</tr>
</tbody>
</table>

Test Set Error Rate on Example 1 with Linear SVM
### 4.1 Experimental Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k</th>
<th>$m_R$</th>
<th>$m_G$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dexter</td>
<td>94.1</td>
<td>505</td>
<td>878</td>
</tr>
</tbody>
</table>

Validation Set Error Rate on Dexter with Linear SVM

![Graph showing validation set error rate vs. #Features (m)]
4.1 Experimental Results

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

Spam with linear SVM

Test Set Error Rate [%] vs. # Features (m)

Baseline TD
Baseline TDI

On TD
On TDI

Unsupervised
4.1 Experimental Results

With TD and its FS versions

Dexter (TD) with linear SVM

Similar results between both our methods
4.1 Experimental Results

With TDI and its FS versions

More stable than MIM at high degrees of reduction
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, “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

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

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

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 \( D = 0.01\var(X_i) \)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>EFB</th>
<th>Our FD step</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>Example1</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Dexter</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>SpamBase</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>WDBC</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Wine</td>
<td>2</td>
<td>16</td>
</tr>
</tbody>
</table>
5.1 Experimental Results (on FD)

Amount of memory (bytes) to represent the datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Original</th>
<th>EFB</th>
<th>Our FD step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>98,600,000</td>
<td>6,033,300</td>
<td>1,654,900</td>
</tr>
<tr>
<td>Dexter</td>
<td>198,300,000</td>
<td>26,742,300</td>
<td>3,542,825</td>
</tr>
<tr>
<td>SpamBase</td>
<td>947,800</td>
<td>479,655</td>
<td>98,347</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>45,500</td>
<td>10,881</td>
<td>5,265</td>
</tr>
<tr>
<td>WDBC</td>
<td>65,100</td>
<td>4,268</td>
<td>8,891</td>
</tr>
<tr>
<td>Wine</td>
<td>8,800</td>
<td>890</td>
<td>1,157</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Representation</th>
<th>Ionosphere</th>
<th>WDBC</th>
<th>Wine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SVM</td>
<td>NB</td>
<td>KNN</td>
</tr>
<tr>
<td>Original</td>
<td>9.95</td>
<td>12.94</td>
<td>11.94</td>
</tr>
<tr>
<td>EFB</td>
<td>16.92</td>
<td>30.35</td>
<td>17.41</td>
</tr>
<tr>
<td>FD step</td>
<td>13.43</td>
<td>18.91</td>
<td>16.42</td>
</tr>
</tbody>
</table>
5.1 Experimental Results (on FD + FS)

![Graph showing test set error rate vs. number of features for WDBC with KNN. The graph compares different feature selection methods: Float + FI, EFB + FI, FD Step + FI, and FD Step + UFS Step. The error rate decreases as the number of features increases, with the UFS Step method showing the lowest error rate.]
5.1 Experimental Results (on FD + FS)

SpamBase with linear SVM

Test Set Error Rate [%]

# Features (m)

Float + mrMR
EFB + mrMR
FD Step + mrMR
FD Step + UFS Step
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

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

http://www.prtools.org/

http://zti.if.uj.edu.pl/~merkwirth/entool.htm
References


References


References


References


