Modern Information Retrieval

Dimensionality reduction and feature selection

Hamid Beigy

Sharif university of technology

May 3, 2025

Table of contents

Introduction

The complexity of any classifier depends on the number of input variables or features. These complexities include

1. Time complexity: In most learning algorithms, the time complexity depends on the number of input dimensions(D) as well as on the size of training set (N). Decreasing D decreases the time complexity of algorithm for both training and testing phases.
2. Space complexity: Decreasing D also decreases the memory amount needed for training and testing phases.
3. Samples complexity: Usually the number of training examples (N) is a function of length of feature vectors (D). Hence, decreasing the number of features also decreases the number of training examples.

   Usually the number of training pattern must be 10 to 20 times of the number of features.

Introduction

1. In text classification, we usually represent documents in a high-dimensional space, with each dimension corresponding to a term.
2. In this lecture: axis = dimension = word = term = feature
3. Many dimensions correspond to rare words.
4. Rare words can mislead the classifier.
5. Rare misleading features are called noise features.
6. Eliminating noise features from the representation increases efficiency and effectiveness of text classification.
7. Eliminating features is called feature selection.

Introduction(example)

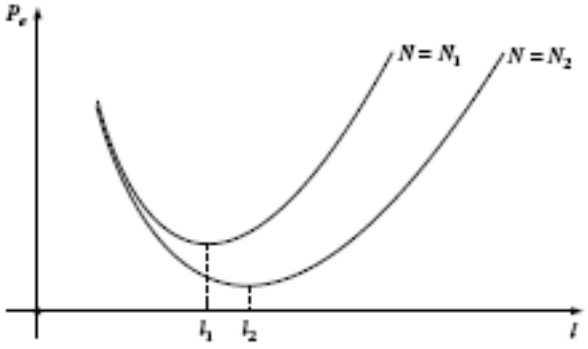
1. Let's say we're doing text classification for the class *China*.
2. Suppose a rare term, say ARACHNOCENTRIC, has no information about *China*.
3. But all instances of ARACHNOCENTRIC happen to occur in *China* documents in our training set.
4. Then we may learn a classifier that incorrectly interprets ARACHNOCENTRIC as evidence for the class *China*.
5. Such an incorrect generalization from an accidental property of the training set is called over-fitting.
6. Feature selection reduces over-fitting and improves the accuracy of the classifier.

There are several reasons why we are interested in reducing dimensionality as a separate preprocessing step.

1. Decreasing the time complexity of classifiers or regressors.
2. Decreasing the cost of extracting/producing unnecessary features.
3. Simpler models are more robust on small data sets. Simpler models have less variance and thus are less depending on noise and outliers.
4. Description of classifier is simpler / shorter.
5. Visualization of data is simpler.

Peaking phenomenon

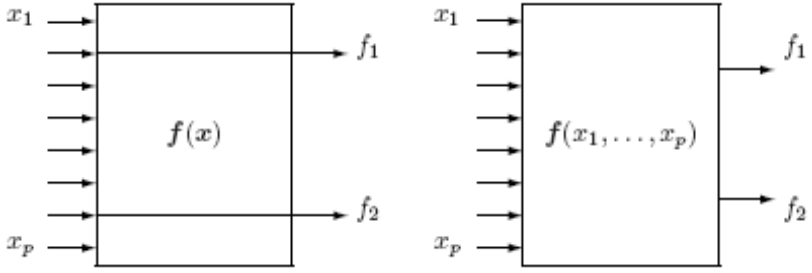
1. In practice, for a finite N, by increasing the number of features we obtain an initial improvement in performance, but after a critical value further increase of the number of features results in an increase of the probability of error.
2. This phenomenon is also known as the peaking phenomenon.



3. If the number of samples increases ($N_2 \gg N_1$), the peaking phenomenon occurs for larger number of features ($l_2 > l_1$).

Dimensionality reduction methods

1. There are two main methods for reducing the dimensionality of inputs
   - Feature selection: These methods select d ($d < D$) dimensions out of D dimensions and $D - d$ other dimensions are discarded.
   - Feature extraction: Find a new set of d ($d < D$) dimensions that are combinations of the original dimensions.



Feature selection methods

Feature selection methods

1. Feature selection methods can be categorized into three categories.
   - Filter methods: These methods use the statistical properties of features to filter out poorly informative features.
   - Wrapper methods: These methods evaluate the feature subset within classifier/regressor algorithms. These methods are classifier/regressors dependent and have better performance than filter methods.
   - Embedded methods:These methods use the search for the optimal subset into classifier/regression design. These methods are classifier/regressors dependent.
2. Two key steps in feature selection process.
   - Evaluation: An evaluation measure is a means of assessing a candidate feature subset.
   - Subset generation: A subset generation method is a means of generating a subset for evaluation.

1. The filter methods has the following structure

$\textsc{SelectFeatures}(\mathbb{D}, c, k)$
1 $V \leftarrow \textsc{ExtractVocabulary}(\mathbb{D})$
2 $L \leftarrow []$
3 **for each** $t \in V$
4 **do** $A(t, c) \leftarrow \textsc{ComputeFeatureUtility}(\mathbb{D}, t, c)$
5 $\quad \textsc{Append}(L, \langle A(t, c), t \rangle)$
6 **return** $\textsc{FeaturesWithLargestValues}(L, k)$

2. How do we compute A, the feature utility?

Different filter methods

1. A feature selection method is mainly defined by the feature utility measure it employs
2. Feature utility measures:
   - Frequency – select the most frequent terms
   - Mutual information – select the terms with the highest mutual information
   - Mutual information is also called information gain in this context.
   - Chi-square (see book)

Mutual information

1. In probability theory and information theory, the mutual information (MI) of two random variables is a measure of the mutual dependence between the two variables.
2. MI determines how similar the joint distribution $p(x, y)$ is to the products of factored marginal distribution $p(x)$ and $p(y)$.
3. Formally, the mutual information of two discrete random variables x and y can be defined as
$$MI(x, y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right)$$
4. In the case of continuous random variables, the summation is replaced by a definite double integral
$$MI(x, y) = \int_{\mathcal{X}} \int_{\mathcal{Y}} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) dxdy$$

Mutual information

1. Compute the feature utility $A(t, c)$ as the mutual information (MI) of term t and class c.
2. MI tells us "how much information" the term contains about the class and vice versa.
3. For example, if a term's occurrence is independent of the class (same proportion of docs within/without class contain the term), then MI is 0.
4. Definition:

$$I(U; C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t) P(C = e_c)}$$

1. Based on maximum likelihood estimates, formula we actually use is

$$I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}}$$

2. N_{10}: number of documents that contain t ($e_t = 1$) and are not in c ($e_c = 0$);
3. N_{11}: number of documents that contain t ($e_t = 1$) and are in c ($e_c = 1$);
4. N_{01}: number of documents that do not contain t ($e_t = 1$) and are in c ($e_c = 1$);
5. N_{00}: number of documents that do not contain t ($e_t = 1$) and are not in c ($e_c = 1$);
6. $N_{1.} = N_{10} + N_{11}$.
7. $N = N_{00} + N_{01} + N_{10} + N_{11}$.

How to compute MI values

1. Alternative way of computing MI:

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{N(U = e_t, C = e_c)}{E(U = e_t)E(C = e_c)}$$

2. $N(U = e_t, C = e_c)$ is the count of documents with values e_t and e_c .
3. $E(U = e_t, C = e_c)$ is the expected count of documents with values e_t and e_c if we assume that the two random variables are independent.

MI example for poultry/EXPORT in Reuters

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$
$e_t = e_{\text{EXPORT}} = 1$	$N_{11} = 49$	$N_{10} = 27{,}652$
$e_t = e_{\text{EXPORT}} = 0$	$N_{01} = 141$	$N_{00} = 774{,}106$

Plug these values into formula:

$$
\begin{aligned}
I(U;C) &= \frac{49}{801{,}948} \log_2 \frac{801{,}948 \cdot 49}{(49+27{,}652)(49+141)} \\
&\quad + \frac{141}{801{,}948} \log_2 \frac{801{,}948 \cdot 141}{(141+774{,}106)(49+141)} \\
&\quad + \frac{27{,}652}{801{,}948} \log_2 \frac{801{,}948 \cdot 27{,}652}{(49+27{,}652)(27{,}652+774{,}106)} \\
&\quad + \frac{774{,}106}{801{,}948} \log_2 \frac{801{,}948 \cdot 774{,}106}{(141+774{,}106)(27{,}652+774{,}106)} \\
&\approx 0.000105
\end{aligned}
$$

MI feature selection on Reuters

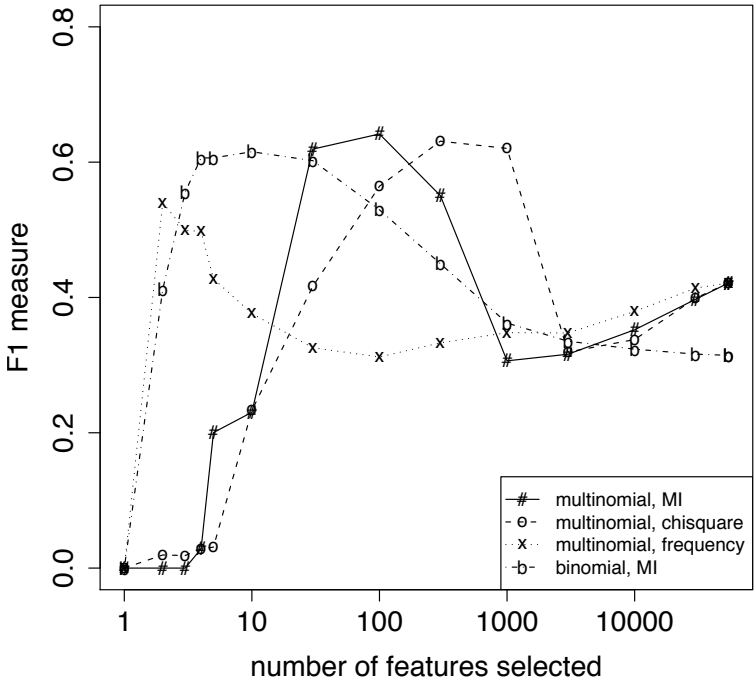
Class: *coffee*

term	MI
COFFEE	0.0111
BAGS	0.0042
GROWERS	0.0025
KG	0.0019
COLOMBIA	0.0018
BRAZIL	0.0016
EXPORT	0.0014
EXPORTERS	0.0013
EXPORTS	0.0013
CROP	0.0012

Class: *sports*

term	MI
SOCCER	0.0681
CUP	0.0515
MATCH	0.0441
MATCHES	0.0408
PLAYED	0.0388
LEAGUE	0.0386
BEAT	0.0301
GAME	0.0299
GAMES	0.0284
TEAM	0.0264

Effect of feature selection (Naive Bayes)



Feature extraction

1. Let S consist of N points over D feature, i.e. it is an $N \times D$ matrix

$$S = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{pmatrix}.$$

2. Point $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ is a D-dimensional vector spanned by the D basis vectors $e_1, e_2, \dots, e_D$, e_i corresponds to i^{th} feature.

3. The standard basis is an orthonormal basis: the basis vectors are pairwise orthogonal $e_i^T e_j = 0$, and have unit length $\|e_i\| = 1$.

4. Given any other set of D orthonormal vectors $u_1, u_2, \dots, u_D$,with $u_i^T u_j = 0$ and $\|u_i\| = 1$ (or $u_i^T u_i = 1$), we can re-express each point x as the linear combination

$$x = a_1 u_1 + a_2 u_2 + \dots + a_D u_D.$$

1. In PCA, we compute the eigenvalues of Σ.
2. Since Σ is positive semidefinite, its eigenvalues must all be non-negative, and we can thus sort them in decreasing order

   $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_{j-1} \geq \lambda_j \geq \ldots \geq \lambda_D \geq 0$
3. We then select the k largest eigenvalues, and their corresponding eigenvectors to form the best k-dimensional approximation.
4. Since Σ is symmetric, for two different eigenvalues, their corresponding eigenvectors are orthogonal. (Show it)
5. If Σ is positive definite ($x^T \Sigma x > 0$ for all non-null vector x), then all its eigenvalues are positive.
6. If Σ is singular, its rank is k ($k < D$) and $\lambda_i = 0$ for $i = k + 1, \ldots, D$.

References

Reading

1. Chapter 13 of Information Retrieval Book[1]
2. Section 9.1 of Search Engines - Information Retrieval in Practice Book[2].

[1] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

[2] W. Bruce Croft, Donald Metzler, and Trevor Strohman (2009). *Search Engines - Information Retrieval in Practice*. Pearson Education.

References

- Croft, W. Bruce, Donald Metzler, and Trevor Strohman (2009). *Search Engines - Information Retrieval in Practice*. Pearson Education.
- Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Questions?