Spatial representation of object location for image matching in CBIR

Tatiana Jaworska

Systems Research Institute, Polish Academy of Sciences,
01-447, 6 Newelska Street, Warsaw, Poland
Tatiana.Jaworska@ibspan.waw.pl

Keywords: CBIR, spatial location, image matching, principal component analysis (PCA), search engine.

Abstract. At present a great deal of research is being done in different aspects of Content-Based Image Retrieval (CBIR)**.** Representation of graphical object location in an image is one of the important tasks that must be dealt with in image DB as an intermediate stage prior to further image retrieval. The issue we address is the principal component analysis (PCA) applied to spatial representation of object location. We propose how to describe the object's spatial location to use it later in the search engine for image comparison. In this paper, we present the promising results of image retrieval based on the number of objects in images, object spatial location and object similarity.

1 Introduction

In recent years, the availability of image resources and large image datasets has increased tremendously. This has created a demand for effective and flexible techniques for automatic image classification and retrieval. Although attempts to construct Content-Based Image Retrieval (CBIR) in an efficient way have been made before [6], the extraction of semantically rich metadata from computationally accessible low-level features, is still considered a major scientific challenge because images and graphical data are complex in terms of visual and semantic contents. Depending on the application, images are modelled using their:

- visual properties (or a set of relevant visual features) [3],
- semantic properties [2], [15],
- spatial or temporal relationships of graphical objects [5].

The spatial relationships of graphical objects together with the classification problem are crucial for multimedia information retrieval in general, and for image retrieval in particular.

Object classification is mentioned here as an important issue in the context of CBIR because it is used for several purposes in the system, for example [13]:

- to compare whole images. Specifically, an algorithm which describes a spatial object location needs classified objects;
- to help the user form a query in the GUI. The user forms a query choosing graphical objects semantically collected in groups;
- to compare image objects coming from the same class as a stage in the image retrieval process.

1.1 CBIR concept overview

In general, our system consists of five main blocks (Fig. 1):

- the image preprocessing block, responsible for image segmentation and extraction of image object features, implemented in Matlab, (cf. [12]);
- the database, which is implemented in the Oracle Database (DB), stores information about whole images, their segments (here referred to as graphical objects), segment attributes, object location, pattern types and object identification, (cf. [14]);
- the classification module, used by the search engine and the GUI, is implemented in Matlab. Classification helps in the transition from rough graphical objects to human semantic elements. [12]
- the search engine, responsible for the searching procedure and retrieval process, based on feature vectors of objects and spatial relationship of these objects in an image, implemented in Matlab. The algorithms applied in this module will be described in general in section 3 and in detail in [12].
- the graphical user's interface (GUI) also implemented in Matlab, which allows users to compose their own image, consisting of separate graphical objects as a query. We have had to create a user-friendly semantic system.

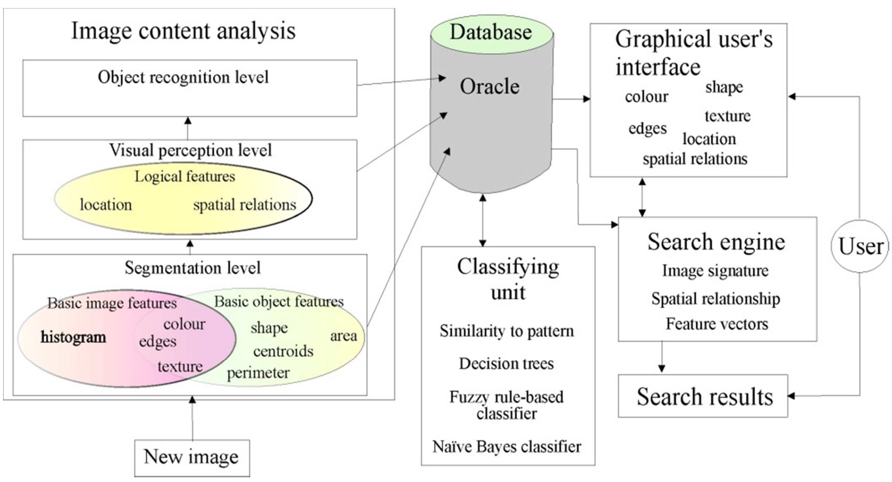

Fig. 1. Block diagram of our content-based image retrieval system.

1.2 *Representation of graphical data*

In our system, a new image is segmented, yielding as a result a collection of objects. Both the image and the extracted objects are stored in the database. Each object, selected according to the algorithm presented in detail in [15], is described by some certain low-level features. The features describing each object include: average colour k_{av}, texture parameters T_p, area A, convex area A_c, filled area A_f, centroid $C=\{x_c, y_c\}$, eccentricity e, orientation α, moments of inertia m_{11}, bounding box $\{bb_1(x,y), ..., bb_s(x,y)\}$ (s – number of vertices), major axis length m_{long}, minor axis length m_{short}, solidity s and Euler number E and Zernike moments $Z_{00},\ldots,Z_{33}$. All features, as well as extracted images of graphical objects, are stored in the DB.

Let *Fo* be a set of features where:

$$F_O = \{k_{av}, T_p, A, A_c ,\ldots, E\} \tag{1}$$

Hence, for an object, we construct a feature vector: $\mathbf{x} = [x_1, x_2, \ldots, x_r]$, where n is the number of the above-mentioned features, in our system $r = 45$. All the information is fed into the database.

1.3 *Classification methods used in the CBIR system*

Thus, the feature vector *Fo* (cf. (1)) is used for object classification. We have to classify objects in order to assign them to a particular class and to compare objects belonging to the same class.

So far, four kinds of classifiers have been implemented in our system. Firstly, there is the most intuitive one based on a comparison of features of the classified object with a class pattern. The problem of finding adequate weights, especially in the case of comparing complex values of some features, is also solved. [1]

Secondly, decision trees as another option in a great number of classifying methods are used [7]. In order to avoid high error rates resulting from as many as 28 classes we use the hierarchical method. The more general division is created by dividing the whole data set into four clusters applying *k*-means clustering. The most numerous classes of each cluster constituting a meta-class are assigned to four decision trees, which results in 7 classes for each one.

Thirdly, to identify the most ambiguous objects we have built fuzzy rule-based classifiers (FRBC). There the ranges of membership functions for linguistic values for fuzzy rule-based classifiers according to crisp attributes are calculated [9, 10, 11].

Additionally, the Naïve Bayes classifier [17] has been implemented and now it seems to be as good as FRBC.

2 Spatial Relationship of Graphical Objects

It is easy for the user to recognize visually spatial location but the system supports full automatic identification based on rules for location of graphical elements which is a challenging task.

Let us assume that we analyse a house image. Then, for instance, an object which is categorized as a window cannot be located over an object which is categorized as a chimney. For this example, rules of location mean that all architectural objects must be inside the bounding box of a house. For the image of a Caribbean beach, an object which is categorized as a palm cannot grow in the middle of the sea, and so on.

In our system, spatial object location in an image is used as the global feature. For this purpose, the mutual position of all objects is checked. The location rules are also stored in the pattern library [13]. Moreover, object location reduces the differences between high-level semantic concepts perceived by humans and low-level features interpreted by computers.

For the comparison of the spatial features of two images an image I_i is interpreted as a set of n objects composing it:

$$I_i = \{o_{i1}, o_{i2}, ..., o_{in}\} \tag{2}$$

Each object o_{ij} is characterized by a unique identifier and a set of features discussed earlier. This set of features includes a centroid $C_{ij} = (x_{ij}, y_{ij})$ and a label L_{ij} indicating the class of an object o_{ij} (such as window, door, etc.), identified in the process described in [13]. Let us assume that there are, in total, M classes of the objects recognized in the database. For convenience, we number the classes of the objects and thus L_k's are just IDs of classes.

Formally, let I be an image consisting of n objects and k be the number of different classes of these objects, $k \le M$, because usually there are some objects of the same type in the image, for example, there can be four windows in a house.

Then, by the signature of an image I_i (2) we mean the following vector:

$$\text{Signature}(I_i) = [\text{nobc}_{i1}, \text{nobc}_{i2}, \ldots, \text{nobc}_{iM}] \tag{3}$$

where: nobc_{ik} denotes the number of objects of class L_k present in the representation of an image I_i, i.e. such objects o_{ij}.

Additionally, for an image I_i we consider a representation of spatial relationships of the image objects. The o_{ij} objects' mutual spatial relationship is calculated based on the algorithm below.

```
Algorithm: PCV for an image
Input: ID_image object, object centroids, object classes
Output: PCV
Method:
```

```
    No_class = Binomial class combinations
    Com = Binomial centroid combinations
    For 1:object numbers
        V_max = max Euclid distance
        β = angle between V_max and the horizontal axis
        θ = 90-atan(V_max,L→_ij)
        if L_p>L_q
            θ = θ+360
        S = Set all (L_p,L_q,θ)
    End{for}
    V = princomp(S)
End{Method}
```

Now, we consider one image; let C_p and C_q be two object centroids with $L_p<Lq$, located at the maximum distance from each other in the image, i.e.,

$$\text{dist}\,(C_p\,,C_q) = \max\,\{\text{dist}\,(C_i\,,C_j)\ \forall i,j \in \{1,2,\ldots,k\} \text{ and } L_i \neq L_j\} \tag{4}$$

where: dist(•) is the Euclidean distance between two centroids (see Fig. 2 middle subplot). The line joining the most distant centroids is the line of reference and its direction from centroid C_p to C_q is the direction of reference for computed angles θ_{ij} between other centroids. This way of computing angles makes the method invariant to image rotation.

Thus, we received triples (L_i , L_j, θ_{ij}) where the mutual location of two objects in the image is described in relation to the line of reference (see Fig. 2 middle subplot). Thus, there are $T=m(m\text{-}1)/2$ numbers of triples, generated to logically represent an image consisting of m objects. Let S be a set of all triples, then we apply the concept of principal component analysis (PCA) proposed by Chang and Wu [4] and later modified by Guru and Punitha [8] to determine the first principal component vectors (PCVs). For further analysis we use the first nine coefficients of the PCV (some example are shown in Table 1), which are the "spatial components" of the representation of an image I_i, and are denoted PCV_i.

Fig. 2 presents the most important stages in the determination of the spatial object location: from the presentation of the original image (top), through the object centroid locations (colours indicate particular classes) (middle subplot), to the 3D subplot of the principal components (bottom).

Table 1. Representative principal component vectors for the images shown in Fig. 2.

Image name	First component	Second component	Third component
House-front	-0,001786	-0,003713	0,999992
Domy-banino-1	0,000206	0,003988	0,999992
Houselawn I_1	0,000388	0,001869	0,999998
Houselandscape I_2	0,004109	0,001557	0,999990

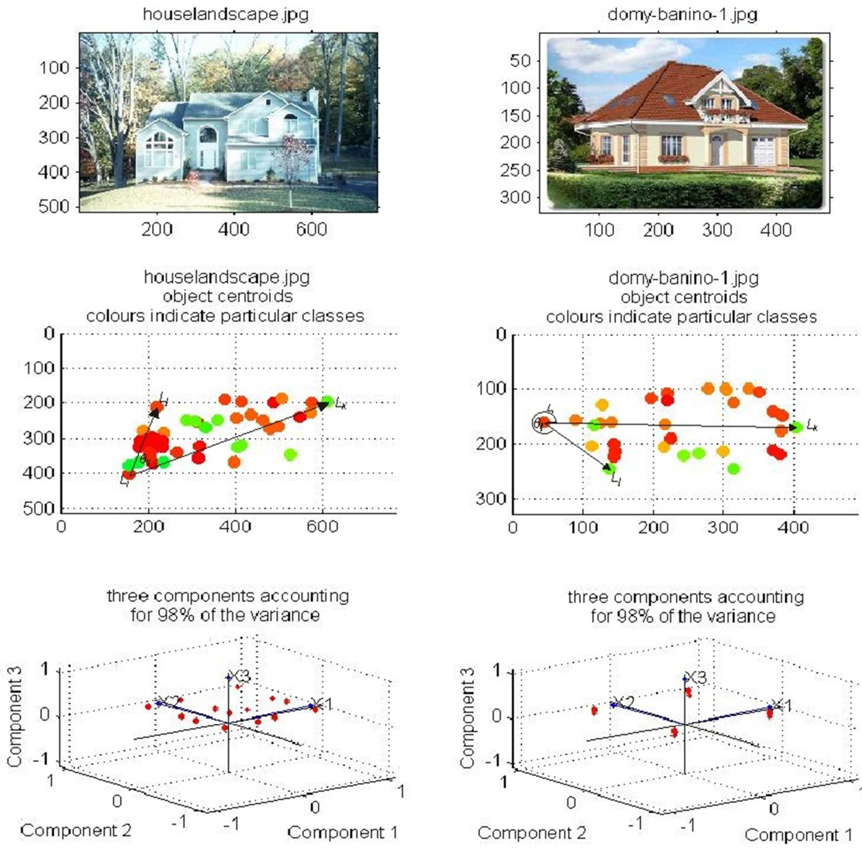

Fig. 2. The main stages of the PCV applied to determine the unique object spatial location in an image.

3 Construction of the search engine

Now, we will describe how the similarity between two images is determined and used to answer a query. Let a query be an image I_q, such as $I_q = \{o_{q1}, o_{q2}, \ldots, o_{qn}\}$ (cf. (2)). An image in the database is denoted as I_b, $I_b = \{o_{b1}, o_{b2}, \ldots, o_{bm}\}$. In order to answer the query, represented by I_q, we compare it with each image I_b from the database in the following way.

A query image will be obtained from the GUI, where the user will construct their own image from selected DB objects. Now the GUI is under construction.

First of all, we determine a similarity measure $\mathrm{sim}_{\mathrm{sgn}}$ between query I_q and image I_b:

$$\mathrm{sim}_{\mathrm{sgn}}(I_q, I_b) = d_H(\mathrm{sgn}(I_q), \mathrm{sgn}(I_b)) \tag{6}$$

computing the Hamming distance between two vectors of their signatures (cf. (3)).

If the similarity (6) is smaller than a threshold (a parameter of the query), then image I_b is rejected, i.e., not considered further in the process of answering query I_q. Otherwise, we proceed to the next step and we find the spatial similarity $\mathrm{sim}_{\mathrm{PCV}}$ of images I_q and I_b, computing the Euclidean distance between their PCVs as:

$$\mathrm{sim}_{\mathrm{PCV}}(I_q, I_b) = 1 - \sqrt{\sum_{i=1}^{3} (PCV_{bi} - PCV_{qi})^2} \tag{7}$$

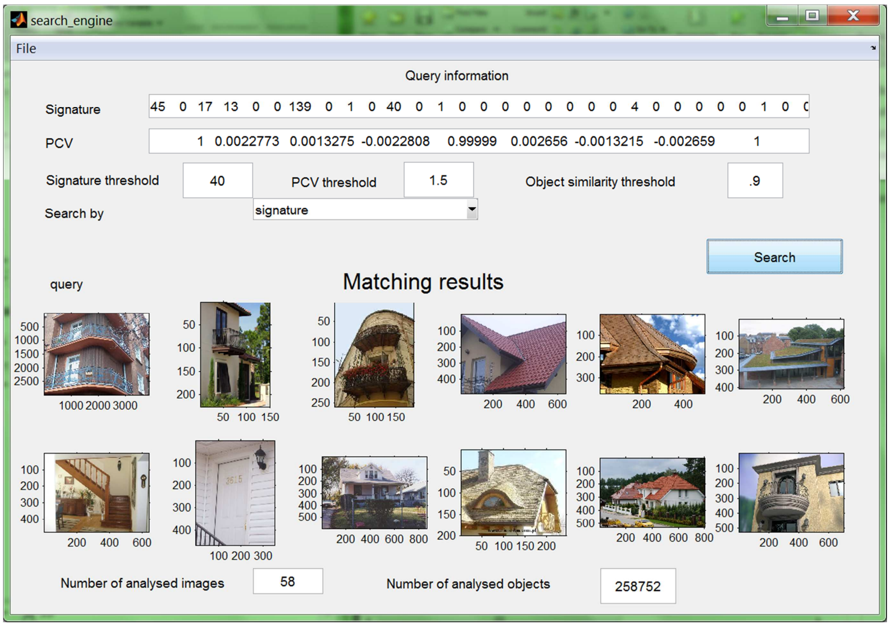

Fig. 3. An example of the search engine's operation. Matching images are found as an answer to the query (left image). The first step in the process of comparison is signature.

If the similarity (7) is smaller than the threshold (a parameter of the query), then image I_b is rejected, i.e., not considered further in the process of answering query I_q. The order of steps 6 and 7 can be reversed because they are the global parameters and hence can be selected by the user.

Next, we proceed to the final step, namely, we compare the similarity of the objects representing both images I_q and I_b. For each object o_{qi} present in the representation of the query I_q, we find the most similar object o_{bj} of the same class, i.e., $L_{qi} = L_{bj}$. If there is no object o_{bj} of the class L_{qi}, then $\mathrm{sim}_{\mathrm{ob}}$ (o_{qi}, o_b) is equal to 0. Otherwise, similarity sim_{ob} (o_{qi}, o_b) between objects of the same class is computed as follows:

$$\mathrm{sim}_{\mathrm{ob}}(o_{qi}, o_{bj}) = 1 - \sqrt{\sum_l (Fo_{qil} - Fo_{bjl})^2} \tag{8}$$

where l indexes the set of features F_O used to represent an object, as described in (1).

When we find highly similar objects (for instance, $\mathrm{sim}_{ob} > 0.9$), we eliminate these two objects from the process of comparison described by Mucha and Sankowski [16]. This process is realized according to the Hungarian algorithm for the assignment problem implemented by Munkres. Thus, we obtain the vector of similarities between query I_q and image I_b.

$$\mathrm{sim}(I_q, I_b) = \begin{bmatrix} \mathrm{sim}_{\mathrm{ob}}(o_{q1}, o_{b1}) \\ \vdots \\ \mathrm{sim}_{\mathrm{ob}}(o_{qn}, o_{bn}) \end{bmatrix} \tag{9}$$

where n is the number of objects present in the representation of I_q. In order to compare images I_b with the query I_q, we compute the sum of sim_{ob} (o_{qi}, o_b) and then use the natural order of the numbers. Therefore, the image I_b is listed as the first in the answer to the query I_q, for which the sum of similarities is the highest.

4 Results

Fig. 3 and 4 present tentative results obtained by having applied the above-mentioned procedure. We set all the thresholds for the search engine and the similarities are calculated. The first 11 most similar images to the query are displayed.

Now, we are using the whole images as a set of objects because the GUI is being constructed. Each image consists of a different number of elements, hence is divided into a different number of segments but on average there are 41 ones per image. In the next step, a measure or a ranking of image matching quality should be determined.

So far , the program code has not been optimised in terms of the time efficiency when confronted with thousands of images. Many experiments have to be carried out and different options have to be examined before taking a decision about the code optimization.

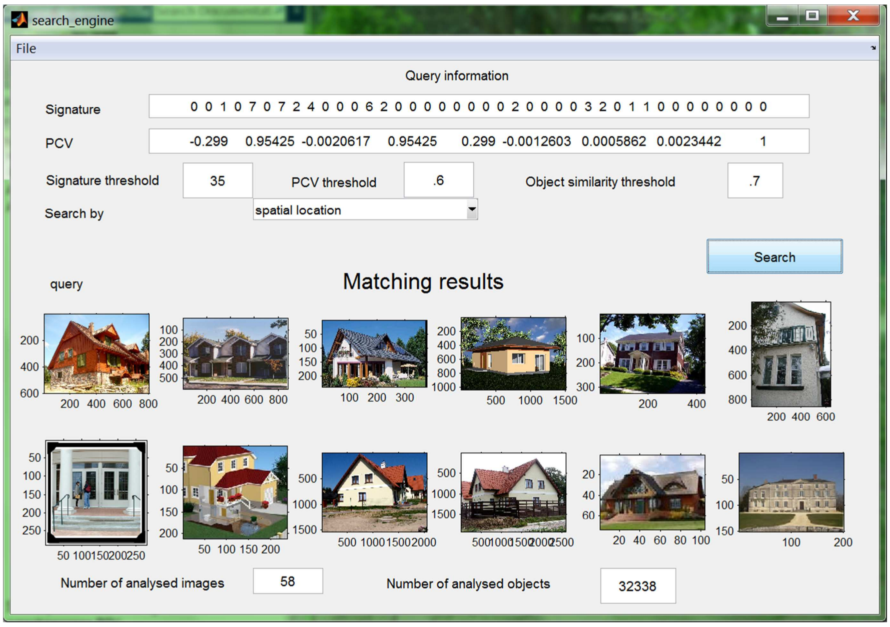

Fig. 4. An example of the search engine's operation. Matching images are found as an answer to the query (top left image). The first step in the process of comparison is the spatial location.

5 Conclusions

The methods already implemented will be also evaluated in terms of the addition of new classes to the system. GUI development will also enforce the introduction of subclasses to some of the most numerous classes.

Intensive computational experiments are under way in order to draw some conclusions regarding the choice of parameters for the search engine. However, the results we have obtained so far, using the simplest configuration, are quite promising.

As for the prospects for future work, image onthology should be prepared as a more powerful method for representation of objects in images. Image onthology will represent the semantic connections among images which are now impossible to add to image retrieval by the search engine.

References

1. Albert A., Zhang L.: A novel definition of the multivariate coefficient of variation, Biomedical J., Vol. 52 (5), Wiley-VCH, pp. 667-675 (2010)
2. Berzal F., Cubero J.C., Kacprzyk J., Marin N., Vila M.A., Zadrożny S.: A General Framework for Computing with Words in Object-Oriented Programming. In: Bou-

chon-Meunier B. (ed.), International Journal of Uncertainty. Fuzziness and Knowledge-Based Systems, Vol. 15, (Suppl.), Feb., pp. 111-131, World Scientific Publishing Company, Singapore (2007)

3. Candan K.S., Li W-S.: On Similarity Measures for Multimedia Database Applications. Knowledge and Information Systems, Vol. 3, pp. 30-51 (2001)
4. Chang, C.C., Wu, T.C., 1995. An exact match retrieval scheme based upon principal component analysis. Pattern Recognition Letters 16, pp. 465–470,
5. Cubero J.C., Marin N., Medina J.M., Pons O., Vila, M.A.: Fuzzy Object Management in an Object-Relational Framework, Proceedings of the 10th International Conference IPMU, Perugia, Italy, pp. 1775-1782 (2004)
6. Deb, S. (ed.).: Multimedia Systems and Content-Based Image Retrieval, Chap. VII and XI, IDEA Group Publishing, Melbourne (2004)
7. Fayyad U. M. and Irani K. P.: The attribute selection problem in decision tree generation, Proceedings of the 10th National Conference on artificial Intelligence, AAAI, 7, pp. 104-110 (1992)
8. Guru, D.S., Punitha, P.: An invariant scheme for exact match retrieval of symbolic images based upon principal component analysis. Pattern Recognition Letters, Vol. 25, pp. 73 -86 (2004)
9. Hamilton-Wright, A., Stashuk, D.W.: Constructing a Fuzzy Rule Based Classification System Using Pattern Discovery. In NAFIPS 2005 Annual Meeting of the Narth American Fuzzy Information Processing Society. IEEE. pp. 460-465 (2005)
10. Ishibuchi H. and Nojima Y.: Toward Quantitative Definition of Explanation Ability of fuzzy rule-based classifiers, IEEE International Conference on Fuzzy Systems, June 27-39, 2011,Taipei, Taiwan, pp. 549-556 (2011)
11. Jaworska T., 2013. Towards Fuzzy Classification in CBIR, In: Information Systems Architecture and Technology. Knowledge Based Approach to the Design, Control and Decision Support, (Eds.) L. Borzemski, A. Grzech, J. Świątek, Zofia Wilimowska, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, Poland, pp. 53-62.
12. Jaworska T.: A Search Engine Concept Based on Multi-feature Vectors and Spatial Relationship, LNAI Vol. 7022, Springer, pp. 137-148 (2011)
13. Jaworska T.: Database as a Crucial Element for CBIR Systems. Proceedings of the 2nd International Symposium on Test Automation and Instrumentation, Vol. 4, World Publishing Corporation, Beijing, China, pp. 1983-1986 (2008)
14. Jaworska T.: Object extraction as a basic process for content-based image retrieval (CBIR) system. Opto-Electronics Review, Association of Polish Electrical Engineers (SEP), Vol. 15(4), Warsaw, pp. 184-195 (2007)
15. Liu Y., Zhang D., Lu G., Ma W-Y.: A survey of content-based image retrieval with high-level semantics. Pattern Recognition, Vol. 40, Elsevier, pp. 262-282 (2007)
16. Mucha M., Sankowski P.: Maximum Matchings via Gaussian Elimination, In: Proceedings of the 45th Annual Symposium on Foundations of Computer Science (FOCS'04), pp. 248-255 (2004)
17. Rish I.: An empirical study of the naive Bayes classifier, Proceedings of IJCAI-2001 workshop on Empirical Methods in AI, pp. 41-46 (2001)