Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Max Berrendorf, Julian Busch

Knowledge Discovery and Data Mining I

WS 2018/19

Exercise 2: Feature Extraction, Similarity Search, Data Reduction

Exercise 2-1 Feature Extraction and Similarity Search

Consider the 5 images in Figure 1 with 36 pixels each.

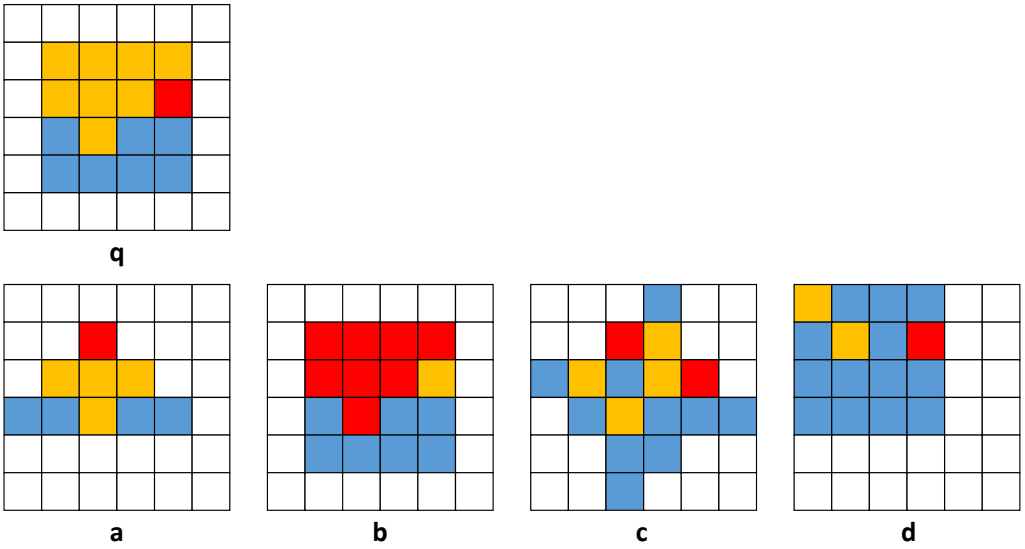

Abbildung 1: 6×6 Pixel Images.

(a) Extract the following features from all images (the white pixels can be ignored):

- Color features: Use a color histogram with bins *red, orange* and *blue*.

(b) Which images are most similar to q w.r.t color when the Euclidean distance is used? Provide a ranking.

Color histograms (red, orange, blue)

$q = (1, 8, 7)$
$a = (1, 4, 4); dist(q, a) = 5$
$b = (8, 1, 7); dist(q, b) = 9.9$
$c = (2, 4, 10); dist(q, c) = 5.1$
$d = (1, 2, 13); dist(q, d) = 8.5$

Ranking: a, c, d, b

(c) If you want to find only the top-k most similar images to q in a database, is there a more efficient way than computing the distance d between q and all images in the database? Name and describe two different approaches to this problem.

Two general approaches to fast query processing are:

- Filter-refine: First filter the whole database using a more efficient filter distance function to obtain a candidate set. Then refine the candidate set by a sequential scan on the candidate set using the original distance function. This makes sense if the original distance function is costly to compute. Useful criteria for filter quality are
  - Indexable: The filter distance should be indexable to allow for fast filtering
  - Complete: The complete query result should be included in the candidate set
  - Efficient: Individual filter distance calculations should be fast
  - Selective: The candidate set should be small so that the refine step is fast
- Indexing: Organize the data in a way that allows for fast access to relevant objects. The main idea is to prune the search space, such that the search can be restricted to a subset of the database. For instance, an R-tree is a spatial index structure which decomposes the full data space into a hierarchy of minimum bounding rectangles. A query then only needs to consider a certain rectangular region of the data space.

(d) The query result might not be fully satisfying. Can you think of different ways of modifying the feature extraction or distance function to obtain possibly better results? Provide at least one alternative for both components.

- From the color perspective, image b might be more similar to q than a. It is basically the same image but with red and orange switched. The problem here is that the Euclidean distance considers all colors to be equally (dis-)similar. By using a Quadratic Form (or Mahalanobis-) Distance, we can explicitly specify similarities between colors. For instance we could specify red and orange to be very similar and blue to be maximally dissimilar to red and orange to get the following result:

$$A = \begin{pmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$dist(q, a) = \sqrt{(q-a) \cdot A \cdot (q-a)^T} = 5$
$dist(q, b) = 3.1$
$dist(q, c) = 4.3$
$dist(q, d) = 8.5$

- So far we have considered only color features. One might also want to find similar images based on shape. For instance, q is similar in shape to b and also to d (if want to be translation-invariant), but it is not similar to a or c. To this end, different shape descriptors have been proposed in the literature. A simple strategy might be to segment the image into sectors and count the number of colored pixels in each sector.

Exercise 2-2 Incremental Aggregation

Given a Data Warehouse with e.g. 10 million entries, additional 1000 entries arrive each day. Rather than recomputing the desired aggregates, an incremental adaptation to the new data should be supported. In order to accelerate the (re-)computation, precomputed intermediate results shall be stored and intermediate results for the new entries shall be computed. What (and how many) values suffice when considering the following aggregates? For each measure note whether it is an algebraic, holistic or distributive measure.

(a) Product.

The product is a distributive aggregation measure since it is an associative pairwise operation:

$$\begin{aligned} prod(D) &= \prod_{x \in D} x \\ &= \left(\prod_{x \in D_1} x \right) \cdot \left(\prod_{x \in D_2} x \right) \\ &= prod(prod(D_1), prod(D_2)) \end{aligned}$$

(b) Mean.

Let $D = D_1 \cup D_2$ with $|D_1| = n_1$ and $|D_2| = n_2$ where D_1 is the data currently in the data warehouse and D_2 is the increment. It suffices to store two values for D_1 and D_2, the sum and $count$, since

$$\begin{aligned} mean(D) = \frac{1}{n_1 + n_2} \sum_{x \in D} x &= \frac{\sum_{x \in D_1} x + \sum_{x \in D_2} x}{n_1 + n_2} \\ &= \frac{sum(D_1) + sum(D_2)}{count(D_1) + count(D_2)}. \end{aligned}$$

Thus, the mean is an algebraic measure. It is not a distributive measure. Towards contradiction assume it would, i.e. for all databases D and partitions $D_1 \uplus D_2$ it holds $mean(D) = mean(mean(D_1), mean(D_2))$, i.e. in particular for $D = \{0, 2, 4, 6\}$, and the partition $D = D_1 \uplus D_2$ with $D_1 = \{0\}$, $D_2 = \{2, 4, 6\}$. Then

$$\begin{aligned} mean(D) &= mean(mean(D_1), mean(D_2)) \\ \frac{0+2+4+6}{4} &= \frac{1}{2} \left(\frac{0}{1} + \frac{2+4+6}{3} \right) \\ \frac{12}{4} &= \frac{1}{2} \cdot \frac{12}{3} \\ 3 &= 2 \end{aligned}$$

which is a contradiction.

To further derive the conditions when the distribution works, consider

$$\begin{aligned}
mean(D) &= mean(mean(D_1), mean(D_2)) \\
\frac{1}{n_1+n_2}\sum_{x\in D} x &= \frac{1}{2}\left(\frac{1}{n_1}\sum_{x\in D_1} x + \frac{1}{n_2}\sum_{x\in D_2} x\right) \\
\frac{1}{n_1+n_2}\sum_{x\in D_1} x + \frac{1}{n_1+n_2}\sum_{x\in D_2} x &= \frac{1}{2n_1}\sum_{x\in D_1} x + \frac{1}{2n_2}\sum_{x\in D_2} x \\
\left(\frac{1}{n_1+n_2} - \frac{1}{2n_1}\right)\sum_{x\in D_1} x &= \left(\frac{1}{2n_2} - \frac{1}{n_1+n_2}\right)\sum_{x\in D_2} x \\
\left(\frac{2n_1-(n_1+n_2)}{2n_1(n_1+n_2)}\right)\sum_{x\in D_1} x &= \left(\frac{n_1+n_2-2n_2}{2n_2(n_1+n_2)}\right)\sum_{x\in D_2} x \\
\left(\frac{n_1-n_2}{2n_1(n_1+n_2)}\right)\sum_{x\in D_1} x &= \left(\frac{n_1-n_2}{2n_2(n_1+n_2)}\right)\sum_{x\in D_2} x \\
\left(\frac{n_1-n_2}{n_1}\right)\sum_{x\in D_1} x &= \left(\frac{n_1-n_2}{n_2}\right)\sum_{x\in D_2} x \\
\frac{1}{n_1}\sum_{x\in D_1} x &= \frac{1}{n_2}\sum_{x\in D_2} x
\end{aligned}$$

The last operation is only an equivalence if $n_1 \neq n_2$. If $n_1 = n_2$, the statement holds trivially. Concluding, the mean can be computed in distributive manner if and only if the partitions have same size, or the same mean.

(c) Variance.

Similarly, the variance is also an algebraic measure:

$$\begin{aligned}
var(D) &= \frac{1}{n_1+n_2-1}\left(\sum_{x\in D} x^2 - \frac{1}{n_1+n_2}\left(\sum_{x\in D} x\right)^2\right) \\
&= \frac{1}{n_1+n_2-1}\left(\sum_{x\in D} x^2 - \frac{1}{n_1+n_2}\left(\sum_{x\in D} x^2 + \sum_{x\in D_1, y\in D_2} xy + \sum_{x\in D_1, y\in D_2} yx\right)\right) \\
&= \frac{1}{n_1+n_2-1}\left(\sum_{x\in D} x^2 - \frac{1}{n_1+n_2}\left(\sum_{x\in D_1} x^2 + \sum_{x\in D_2} x^2 + 2\left(\sum_{x\in D_1} x\right)\left(\sum_{x\in D_2} x\right)\right)\right) \\
&= \frac{ss(D_1) + ss(D_2) - \frac{1}{count(D_1)+count(D_2)}\left(ss(D_1) + ss(D_2) + 2\cdot sum(D_1)\cdot sum(D_2)\right)}{count(D_1) + count(D_2) - 1}
\end{aligned}$$

We need to store three values, the sum, $count$ and additionally the sum of squares (ss). Note that the variance is not distributive, since the information about central tendency is lost (the variance is shift-invariant). The variance $var(D)$ depends on where D_1 and D_2 are located in the data space and in general there is no way to infer that from $var(D_1)$ and $var(D_2)$ alone. However, if $mean(D_1) = mean(D_2) = 0$, one can show that

$$var(D) = \frac{n_1}{n_1+n_2} var(D_1) + \frac{n_2}{n_1+n_2} var(D_2).$$

(d) Median.

The median is a classical holistic measure which means intuitively that we need to look at the whole data at once in order to compute it. For the median to be an algebraic measure, we would need to be

able to represent the median of D as an algebraic function of constant size aggregates of D_1 and D_2. Assume that we have computed such aggregates. Now the idea is that for any two sets D_1 and D_2, we can construct an example where the k-th element of D_1 (or D_2) is the median. That is, we potentially need to access every single element in D_1 (or D_2) from a constant size aggregate. This is clearly not possible. Thus, we need to look at the whole sets D_1 and D_2 together in order to find the median, i.e. the median is a holistic measure.

Exercise 2-3 Histograms

(a) Given are the following data points:

$$1; 1; 4; 4; 5; 5; 5; 5; 6; 7; 7; 9$$

Using three bins, compute

(i) the equi-width histogram

Bin	Elements	Size
$1-3$	1; 1	2
$4-6$	4; 4; 5; 5; 5; 5; 6	7
$7-9$	7; 7; 9	3

(ii) the equi-height histogram

Bin	Elements	Size
$1-4$	1; 1; 4; 4	4
5	5; 5; 5; 5	4
$6-9$	6; 7; 7; 9	4

(b) Assume there is an additional data point 29. What changes in the histograms compared to (a)?

(i) The equi-width histogram degenerates into a first bin containing all data points except 29, an empty bin in the middle, and one dedicated bin for 29. Effectively, we loose all information except that there is an outlier.

(ii) The bins stay the same. Only the range of the last bin is changed. Compared to equi-width, the equi-height histogram is much more robust to outliers.