Daten- und Dimensionsreduktionstechniken für Big Data Analytics

Masterarbeit

Universität Rostock
Fakultä für Informatik und Elektrotechnik
Institut für Informatik

vorgelegt von: Asem Saleh
Matrikelnummer: 210208887
geboren am: 08.02.1982 in Lattakia (Syrien)
Erstgutachter: Prof. Dr. rer.nat.habil. Andreas Heuer
Zweitgutachter: Prof. Dr.-Ing. habil. Ralf Salomon
Betreuer: Dr.-Ing. Holger Meyer

Abgabedatum: September 27, 2016

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Vorlage der angegebenen Literatur und Hilfsmittel angefertigt habe.

Rostock, den September 27, 2016

Contents

Introduction

William Shakespeare once wrote: **"Brevity is the soul of wit"**[1]. It is a powerful quote that applies well to the field of data science and analytics too.

The recent years have witnessed an explosion of data-set size in terms of records and attributes. In an article published on **SINTEF**[2] official website in May 2013 Åse Dragland wrote that about 90% of the data in the world at that time has been created in the past two years alone [34]. The computer giant IBM revealed on its Big Data introductory page that "everyday we create 2.5 quintillion bytes of data" [10]. Citing an Infographic designed by Ben Walker, the Marketing Executive at VoucherCloud [42], the estimated amount of data generated in the entire world has risen from 100GB per day in 1992 to 100GB per hour in 1997 then 100GB per second in 2002. The amount of data generated in 2013 was 28875GB per second and it is expected to reach 50000GB per second in 2018.

The extensive use of social media, e-commerce and entertainment services is one of the major sources behind this data surge. An infographic from Excelacom titled " What Happens in an Internet Minute in 2016?"[16]shows the mind-blowing figures related to the content generated every minute nowadays with about 3.2 billion people around the globe having access to the internet. Here are the figures concerning some major social media sites and online services:

- 701,389 logins on Facebook
- 69,444 hours watched on Netflix
- 150 million emails sent
- 527,760 photos shared on Snapchat
- 51,000 app downloads on Apple's App Store
- $203,596 in sales on Amazon.com
- 120+ new Linkedin accounts
- 347,222 tweets on Twitter
- 28,194 new posts to Instagram
- 38,052 hours of music listened to on Spotify
- 2.4 million search queries on Google
- 2.78 million video views on Youtube
- 20.8 million messages on WhatsApp

[1] **Hamlet** Act 2, scene 2, Pages 86–92
[2] **SINTEF**: Applied Research, Technology and Innovation www.sintef.no

In addition to the impact of social media, there is another major source of data to consider: It is the sensor data, the rising star of the Internet-Of-Things IOT and the tremendous increase of sensor-appliances in the industry as well as everyday life. Following are a few examples:

- 10 Terabytes Sensor data produced by a jet every 30 minutes of flight time [8].
- One petabyte of data generated every second from nuclear physics experiments at the Large Hadron Collider at CERN [44].
- 15 (out of 17) is the number of industry sectors in the U.S. that have more data stored, per company, than the U.S. Library of Congress [20].
- Human activity monitors and assisting environments, E.G. 561 variables in the experiment conducted by Smartlab - Non-Linear Complex Systems Laboratory in Italy and CETpD - Technical Research Centre for Dependency Care and Autonomous Living in Spain [1].

This data explosion demanded that the people working with data have to be qualified to sufficiently analyze and make use of this data, one significant skill in particular is the ability to perform **Dimension Reduction**. This data big-ban urged the rise of new data management techniques. Although several big data platforms as well as parallel data analytics algorithms have been developed, the need for data dimensionality reduction procedures is still a must. No matter how powerful your equipments and sufficient your algorithms, there will be always performance issues when working with huge data sets, and the dimension reduction techniques will come in handy.

More data doesn't Always mean better Insights

Big data can be problematic when it comes to management. Gathering more data will not necessarily serve the ultimate goal of better analytics. Not only it can cause serious performance and capacity issues in many occasions but it can also give unrealistic or false predictions.

One obvious challenge of big data is its size. When it comes to the race between the data density and the storage scalability, the data is winning. The current widespread disk technologies such as RAID (Redundant Array of Independent Disks) can't match the escalating volumes of data that most enterprises are dealing with. New technologies such as SSD (Solid State Drive) can dramatically enhance the performance compared to traditional HDDs [21] , sometimes up to 70% when tested with big data platforms such as MapReduce [13]. However their costs are still much higher not to mention that traditional HDDs offer greater capacities. Bottom line, in order for SSDs to pay off they have to be used with the right job, such as IO-intensive tasks [7].

Big Data does not necessarily mean Good Data. A large percentage of the data might not be of interest. There is a need to intelligently filter this data without dropping essential portions that can serve our task. incomplete, out of context or contaminated data can lead the analysis in the wrong direction. "Big data is not about the data, it's about the analytics", says Harvard University professor Gary King. One of Professor King's big data projects aimed to use twitter feeds and other social media to predict the U.S. unemployment rate. A few keywords were considered, keywords that have relevance to unemployment, such as jobs, unemployment and classifieds. Tweets and social media feeds containing these keywords were extracted. The researchers then looked for correlations between the total number of words per month in this category and the monthly unemployment rate (sentiment analysis by word count). Suddenly there was a tremendous increase in the number of social media posts that fell into the monitored category. When everybody though that the project was paying off and the analysis looked promising, no body noticed that Steve **Jobs** died. There was of course a flood in the social media with posts containing the word "Jobs" as a reaction to the death of Steve Jobs of course,

not due a correlation to the unemployment status [39]. In a big data discussion "How to Think About the Future", Nate Silver, statistical expert and author of The Signal and the Noise, said that our subjective point of view often clashes with the massive volumes of data we are dealing with. Having so much data would possibly make the numbers say almost anything. Humans often tend to look for what they want and expect to find [23]. A study conducted by StreamSets found that companies working with big data are rarely capable of controlling their data flows. The study showed that almost 9 of 10 companies struggle with bad data polluting their data stores with no reliable tools to control the data flows, operations, quality and security [36].

In a nutshell, when faced with large amounts of diverse data, a few critical questions have to be answered:

- There are too many variables or attributes in the data set. Is it necessary to explore each and every variable? Are they all important?
- In case of numeric multi-collinear variables, how can they be identified?
- What is the right technique to use:
  - Decision Tree can sometimes select the right variables. Does it fit all scenarios?
  - Random Forest may do the work, however it takes a high execution time because of the too many attributes.
  - Is it possible to use a Machine-Learning approach that can automatically identify the most significant attributes?
  - It is a Classification problem. Can **SVM** be used with all variables?
- Which tool is better in order to deal with this huge amount of attributes. R or Python?

The work in this thesis is organized in six chapters. Chapter one will go over some preliminaries that are related to data and multidimensionality. It will then outline the concept of reduction with many associated metrics. Chapter two will discuss the native techniques provided by the relation database model, it will stress the superiority of multidimensional databases compared to the traditional relational ones when it comes to answering typical business questions. Multimedia information retrieval systems will the subject of chapter three with the similarity search introduced to handle queries the traditional database systems can't due to the fuzzy nature of data. Chapter four will cover the mathematical transformations that can be employed to reduce the number of attributes in a dataset, PCA will be covered in details. Bio-inspired models and the role they can play to reduce the amount of data to process will be shown in chapter five, while chapter 6 will host a case study about KNIME and how some of the reduction techniques will be used against real-life small and large databases. The thesis will then conclude with a summary and brief discussion.

Chapter 1

Basic Concepts and Metrics

This chapter will go over some preliminaries related to data and multidimensional databases as well as the methods usually employed to assess the reduction performed on the data set. In order for the reduction to produce the desirable result, it should fulfill certain conditions. For instance, a reduction technique should never exceed some defined (or generated) thresholds such as those concerned with the information-loss.

1.1 Preliminaries

1.1.1 Types of Data

Data usually falls into two distinctive broad types [22][38]:

- **Quantitative (numeric):** This type of data deals with numbers i.e. things that can be measured objectively such as dimensions, temperature, humidity and price. Data is represented as numbers that express an amount usually associated with a well-defined measurement unit.

- **Qualitative (attributes):** This type of data deals with characteristics and descriptors that define an object but can't be easily measured, however they can be approached subjectively such as smell, taste, feeling and attractiveness.

In nutshell, qualitative data is the result of measurement while qualitative data is the result of classification (categorization) or judgment.

Each one of the above-mentioned broad data types has two sub-types. Quantitative data will break down into continuous and discrete data, while qualitative data will break down into nominal and ordinal.

Quantitative Data Sub-Types: The distinction here is based on the ability to count.

- **Discrete data** is countable and can't be more precise, it deals with whole indivisible entities such as integers. For example, given a family. the number of children will be always a whole integer such as 0, 1, 2... There is no 2.5 kids in a family.

- **Continuous data** on the other hand can be divided and reduced to finer and finer levels. In stead of whole numbers, continuous data can take any value (usually within a range), such as heights and weights.

Qualitative Data Sub-Types: The distinction here is based on the order of the categories.

- **Qualitative data** will be nominal if there is no natural order between the categories, such as colors.
- With **ordinal data**, there exists some sort of order such as " Short Medium or Tall".

1.1.2 Dimension and Multidimensionality

A multidimensional database is a computer software system designed to allow for the efficient and convenient storage and retrieval of large volumes of data that is (1) intimately related and (2) stored, viewed and analyzed from different perspectives. These perspectives are called dimensions [4].

Multidimensional databases are optimized for data warehouse and online analytical processing (OLAP) applications. They are usually created using input from existing traditional relational databases. Queries will also differ when working with this type of database, they will be more related to typical business questions that can't be usually answered from raw data without viewing that data from various perspectives i.e. to get business insights, trends and summaries [30].

Example: In order to improve the business activity, a marketer in an automobile company might want to examine data collected throughout the organization. The evaluation would require viewing historical sales volume figures from multiple perspectives such as [4]:

- Sales volumes by model
- Sales volumes by color
- Sales volumes by dealership
- Sales volumes over time

Analyzing the sales volumes from multiple perspectives can give answers to important business questions such as:

What is the trend in sales volumes over a period of time for a specific model and color across a specific group of dealerships?

Both relational and multidimensional database systems can handle such questions, however the way the results are produced and presented in addition to the response times and flexibility will significantly differ giving the multidimensional databases superiority over the traditional relational ones, as will be shown in the next chapter.

1.1.3 Multidimensional Indexing

Data is organized in records, each record has a key field that makes it possible to uniquely recognize the record. In a traditional database, **Indexing** is a data structure technique (most commonly a B- tree) used to efficiently retrieve records from the database files based on a set of attributes, namely indexing attributes[11].

An index can be looked at as a sequential listing of the column data, it can be used on any column. Indexes are usually placed on the primary key of a table. An index increases the performance of a query. Similarly, when there is more than one column in a primary key, we have a multiple-column index or what is usually referred to as a **concatenated multiple index**. Let us consider the following scenario:

When two companies merge into one, the **employeeId** of either company isn't unique any more. We add a discriminator column, let us say **sunsidiaryId** to differentiate. Now, the querying still has to be fast, so we place both columns into the index[43]. One would assume that a multiple-column index is always a multidimensional one. However this is not the case in the light of concatenation.

In general, for a multi-dimensional structure, a multiple-column search-key can be defined by using special markers. If fields F1 and F2 are a string and an integer, respectively, and # is a character that cannot appeal in strings (special marker symbol), then the combination of values F1 = 'abcd' and F2 = 123 can be represented by the string 'abcd#123' [6, chapter 5, Page 187].

Most data structures for supporting queries on multidimensional data fall into one of the following categories: Hash-table-like Structures, Tree-like Structures and Bit-map indexes. These structures have their applications in a variety of systems such as Geographical Information Systems GIS and Data Cubes.[6, Chapter 5].

1.2 Dimension Reduction

Definition. Dimension reduction is the process that aims to reduce the data down into its basic components, stripping away any unnecessary parts [31].

Data dimensionality reduction will lead to a more efficient search using some indexing structure. If reduction to only one dimension is feasible then data can be ordered with the ability to build an index on it using conventional methods such as Binary Trees, which would make the search much easier. However with high-dimensional data it is usually desirable to reduce to a dimension larger than one in order to avoid too much information-loss. Following the reduction, some multidimensional indexing method can be employed. However this will not come cheap, the more the reduction the lower the quality of the the query results when performing the search, not to mention the problems associated with using multidimensional indexing [32, Chapter 4, Pages 662-663].

Definition. Recall: defines the query quality. The lower the query radius (higher precision), the higher the recall [32, Chapter 4, Page 664].

Definition. **Dimension Reduction Method:** is a mapping f that transforms a vector v in the original space to a vector $v' = f(v)$ in the transformed lower-dimension space[32, Chapter 4, page 664].

Letd, d' be the distance metrics in the original and transformed space respectively.n, k number of dimensions in the original and transformed space respectively; $n > k$. The mapping f has the following properties:

1. Distances in the transformed space approximate distances in the original space i.e. $d(u, v) \approx d'(f(u), f(v))$ [32, Chapter 4, Page 664]

2. **Pruning Property:** $d'(f(a), f(b)) \leq d(a, b)$; // ensure 100% recall [32, Chapter 4, page 664]

1.3 Reduction Assessment

Many of the methods discussed in this thesis are associated with a threshold i.e. a critical value that will guide the reduction process so that the columns having redundant or unrelated data with percentage higher than the threshold will be removed. After the removal (reduction) another metric will be applied in order to measure the information loss. The threshold can be a user-defined value in some cases, however it is better to generate one based on the dataset. Three classification techniques come in handy for this purpose: MLP, Decision tree and Naive Bayes. After the reduction is complete, the information-loss can be computed using Kullback-Leibler Divergence.

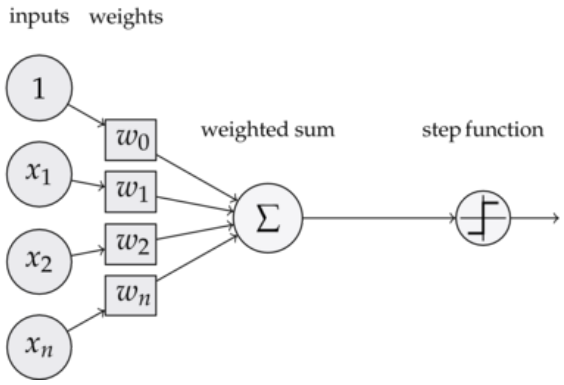


Figure 1.1: The Perceptron Model

1.3.1 Threshold Generation

As mentioned earlier three techniques can be used for this purpose, the MLP neural network is the most common approach and its basic functionality will be explained.

1.3.1.1 MLP (Multi-Layer Perceptron)

Definition. Perceptron: is a mathematical model of a biological neuron, it was proposed by Rosenblatt in the 1950s[28]. In stead of the electrical signals coming from the axons of other connected neurons, the perceptron will receive an input represented as numerical values. At the synapses between the dendrite and axons, electrical signals are modulated in various amounts. This is modeled in the perceptron as weights multiplied with the input values (the weights will usually differ along the connections to represent those various amounts). The response of the biological neuron is called **firing**, it will occur if the total strength of the input signals exceeds a certain threshold. The perceptron mimics this firing process by computing the weighted sum of the inputs then applying a step function on the sum to produce the its final output. As in the biological model, this output will become the input of other connected perceptrons [27]. Figure 1.3.1.1shows the perceptron model [2].

The power of neural networks comes from their ability to learn. A neural network can learn the training samples i.e. the pairs (input, weight) and how to best relate them to the output value we want it to predict. This is called mapping and neural networks can learn almost any mapping function. The hierarchical or multi-layered structure of the neural networks is the reason behind their predictive capability. Neural networks can learn features at different scales or resolutions then combine them into higher-order features. For examples from lines to collections of lines to shapes , somewhat similar to the Darwinism on the path of evolution. The first step in the workflow of a neural network is to initialize the weights, often to small random values, such as values in the range 0 to 0.3, although more complex initialization schemes can be used. Larger weights would indicate increased complexity and fragility, so it is always preferred to keep the weights small[3].

A neuron has a **net function** and an **activation function**. The net function determines how the inputs are combined inside the neuron [9, Chapter 1, Page 13], in the original perceptron model that would be the weighted sum of these inputs.

$$u(\underline{x}) = \sum_{i=1}^{n} w_i x_i + w_0 \tag{1.1}$$

The input with the value1 in figure 1.3.1.1is called the bias, it is used as the threshold. The result of the net function will be evaluated via a linear or non-linear transformation called the activation function to produce the neuron output [9, Chapter 1, Page 14].

$$a = f(u) \tag{1.2}$$

In the original perceptron model depicted in figure1.3.1.1, a single neuron with a linear weighted net function and a threshold (step) activation function is employed.

$$y(\underline{\mathrm{x}}) = \begin{cases} 1 & u(\underline{\mathrm{x}}) \geq 0 \\ 0 & u(\underline{\mathrm{x}}) < 0 \end{cases} \tag{1.3}$$

Additional types of activation functions are available to use with the artificial neurons such as the non-linear activation functions (sigmoid and hyperbolic). This non-linearity allows the network to combine the inputs in more complex ways and in turn widen its capability to model more functions.

The perceptron model can be used for applications such as detection and classification. Given a set of training samples $(x(i), d(i)); i \in I_r$ and testing samples $(x(i), d(i)); i \in I_t$. Here, $d(i) \in 0, 1$ is the desired output value of $y(x(i))$ if the weight vector w is chosen correctly, and I_r and I_t are disjoined index sets. A sequential online perceptron learning algorithm can be applied to iteratively estimate the correct value of w by presenting the training samples to the perceptron neuron in a random, sequential order. The learning algorithm has the following formulation[9, Chapter 1, Page 16]:

$$\underline{\mathrm{w}}(k+1) = \underline{\mathrm{w}}(k) + \eta(d(k) - y(k))\underline{\mathrm{x}}(k) \tag{1.4}$$

where $y(k)$ is computed using the equation 1.1and 1.3. The learning rate $\eta(0 < \eta < |\frac{1}{\underline{\mathrm{X}}(k)}|_{max})$ is a parameter chosen b the user, where $|\frac{1}{\underline{\mathrm{X}}(k)}|_{max}$is the maximum magnitude of the training samples $\{\underline{\mathrm{x}}(k)\}$. The index k is used to indicate that the training samples are applied sequentially to the perceptron in a random order. Each time a training sample is applied, the corresponding output of the perceptron $y(k)$ is to be compared with the desired output$d(k)$. If they are the same, it would mean that the weight vector $\underline{\mathrm{w}}$is correct for this training sample and the weights will remain unchanged. On the other hand, if $y(k) \neq d(k)$, then $\underline{\mathrm{w}}$will be updated with a small step along the direction of the input vector $\underline{\mathrm{x}}(k)$.

if the training samples are linearly separable, the perceptron learning algorithm will converge to a feasible solution of the weight vector within a finite number of iterations. On the other hand, if the training samples are not linearly separable, the algorithm will not converge with a fixed, nonzero value of η.

1.3.2 Kullback–Leibler Divergence

Definition: For two probability distributions f (x) and g(x) for a random variable X, the Kullback-Leibler divergence or relative entropy is given as [19]:

$$D(f||g) = \sum_{x \in X} f(x) log \frac{f(x)}{g(x)}$$

where 0 log $\frac{0}{g} = 0$ and p log $\frac{f}{0} = \infty$

The K-L Divergence compares the entropy of two distributions over the same random variable. In many respects it acts as a measure of dissimilarity or "distance" between distributions. In general, the Kullback-Leibler Divergence has the following properties [19]:

- $D(f||g) \geq 0$ //It is always positive

- $D(f||g) = 0$ iff $f(x) = g(x)$ for all $x \in X$
- $D(f||g) \neq D(g||f)$ // It is asymmetric
- $I(X;Y) = D(f(x,y)||f(x)f(y))$ // Expressed as **Mutual Information**

So the mutual information is the KL divergence between $f(x,y)$ and $f(x)f(y)$. It measures how far a distribution is from independence.

1.3.2.1 K-L as a Distance

K-L Divergence can be used as a tool for distinguishing between statistical populations. The Quantity $log\frac{f(x)}{g(x)}$ is referred to as "the information in x for discrimination between" the distributions f and g. Their divergence is then the mean information for discrimination per observation from P.

1.3.2.2 Information-Loss Metrics

Let $D = (A_1, A_2, ..., A_n)$ and $D\prime = (A_1\prime, A_2\prime, ..., A_n\prime)$ be the original and the the transformed (reduced-version) data sets.

The information-loss for one attribute can be measured using Kullback-Leibler as follows [18]:

$$D(A_i||A_i\prime) = \sum_{j=1}^{n_i}(freq(a_{ij}) * log_2\frac{freq(a_{ij})}{freq(a\prime_{ij})}$$

where a_{ij} is an attribute-value from $A_i$and $freq$ is the frequency of the attribute-value

1.3.2.3 Sum-of-Kullback-Leibler-Divergence

Normalizing the single information losses [18]:

$$D_N(A_i||A\prime_i) = \frac{D(A_i||A\prime_i)}{H(A_i)}$$

with $H(A_i) = \sum_j^{n_i} freq(a_{ij})log_2 freq(a_{ij})$

The information-loss over all attributes is the sum of the normalized information-loss over all A_i: $\sum_{i=1}^{n} D_N(A_i||A\prime_i)$

This chapter covered the background and basic concepts that will be used or needed to understand the materials introduced later. Next chapter will discuss the first set of reduction techniques stemming from the relational database mdoel.

Chapter 2

Native Database Techniques

The first group of dimension reduction techniques are already used to carry out queries on datasets, any one with a basic knowledge in relational databases should be familiar with their functionality. Three query-techniques (relational operators) that would fit the concepts of data and dimension Reduction will be covered: Projection, Selection and Aggregation. While Projection will work on the column-level i.e. dimension reduction, Selection on the other hand will work on the row-level i.e. data reduction. Aggregation will work on both levels, it will use projection to choose columns that fit the aggregation (not all columns necessary would, some valid columns will be circulated) and it will - depending on the aggregation function - introduce new data with a fewer number of rows usually. It is worth mentioning that projection may result in data reduction since the duplicated records will be eventually discarded.

2.1 Preliminaries

Before delving into the techniques a few basic definitions and background information will be mentioned first.

Definition. Algebra: is a mathematical system that uses operands and operators. Operands are variables or values from which new values can be constructed. Operators on the other hand are the symbols that denote the procedures (operations) used to construct new values from the operands. The operands come from a set or a carrier, the operations are closed with respect to this set. Example: (R, $\{*,\div\}$) is an Algebra.

Definition. Relational Algebra: is an algebra whose operands are relations or variables that represent relations. The operators in this scope are the procedures designed to perform all the common tasks needed to work with relations in a database.

Definition. Relation R: Given a set of n not necessarily distinct domains $D_1, D_2, ..., D_n$, a Relation R is a set of $n-tuples$ $d^k =< d_1^k, d_2^k, ..., d_n^k >$ i.e. a subset of the Cartesian Product $(D_1 \times D_2 \times ... \times D_n)$ such that for every element $d^k \epsilon R$ a predefined proposition $P(d^k)$ is true; D_i is the domain of the attribute d_i and n **is the Degree** of R i.e. the number of attributes in the relation. k **is the Cardinality** of R i.e. the number of $n-tuples$ in the relation [35, Chapter 3 : The relational model - Page 29].

In a nutshell, a relation is a table with rows and columns. The set or carrier in this case is the set of all finite relations. The usual set operations such as $\cup$ (Union), $\cap$ (Intersection) and $\setminus$ (difference) are defined, but both operands must have the same relation schema. A **Relation Schema** describes the column heads: Relation name, Name of each field (or column, or attribute) and the Domain of each field. A **Relation Instance** can be described as a concrete table content or a set of records (tuples) that match the schema.

No	Title	Page-Number	Genre	Authors	Year	Language
1	The Da Vinci Code	597	Mystery	Dan Brown	2009	English
2	Inferno	482	Conspiracy	Dan Brown	2013	English
3	The God Delusion	464	Philosophy	Richard Dawkins	2006	English
4	Thus Spoke Zarathustra	252	Philosophy	Friedrich Nietzsche	1883	German
5	A Brief History of Time	256	Cosmology	Stephen Hawking	1988	English
6	The Flowers of Evil	400	Lyric poetry	Charles Baudelaire	1857	French
7	The Origin of Species	502	Biology	Charles Darwin	1859	English

Table 2.1: An instance of the Book relation

Relational algebra forms the basis of a **query language** for relations. There are many query languages designed for Datasets of different types (Relational, XML, information retrieval systems, etc..). SQL (Structured Query Language) is commonly used to retrieve and manipulate data in many relational databases. Relational Algebra will usually work in the background to represent the queries (optimization and execution) in the RDBMS (Relational Database Management System). Oracle, Microsoft Sql Server and Sybase are examples of RDBMS that use Sql.

Example: Given a group of books. A relation $R = Book$ can have the following schema:

$book(No : integer, Title : string, Page - Number : integer, Genre : string, Authors : string; year : number; Language : string)$

A relation instance is shown in table 2.1.

In the relational model we have both unary and binary operations. Unary operations will affect only one relation (projection, selection, etc...) while the binary ones involve two relations. Some binary relational operations apply to a pair of union compatible relations such as the set operations mentioned above, others will apply to a pair of relations with compatible attributes such as join and division.

2.2 Projection

In simple words, to perform a projection means to pick certain columns. Projection is a unary operation that reduces the operand i.e. the relation to produce a new relation with only the set of attributes indicated in the projection list. The resulting relation is constructed by looking at each tuple of the original relation, extracting the attributes in the projection list, in the order specified, and creating from those components a tuple for the new relation. All duplicated tuples will be discarded.

Definition. Projection: $R_1 := \Pi_L(R)$. Where R is the original relation, R_1 is the projection result, L is a set of attributes: $L \subseteq X$; X is the set of all attributes in the schema of R ; $X = \{a_1, a_2, ..., a_n\}$[35, Chapter 4 : Relational algebra - Page 56]. A projection can be written as:

$\Pi_{a_1,a_2,...,a_m}(R)$ where $L = \{a_1, a_2, ..., a_m\}$; $m \leqq n$.

A projection can produce a relation instance with the same values as in the original one or modify them using operators compatible with the attribute type. For instance with the numeric fields, arithmetic expressions involving the attributes can be used. This is called **Extended Projection**. An extended projection may look like this:

$\Pi_{a_1,2*a_2,a_3}$, given that $a_2$has a numeric type. The result will be the same values as in the original instance for the attributes $a_1$and $a_3$and multiplied with 2 for the attribute a_2. It is possible to rename the new column and give it a more descriptive or shorter name: $\Pi_{a_1,2*a_2->c,a_3}$. It is also possible to use expressions that involve more than one attribute of the same type: $\Pi_{a_1+a_2->b,a_3}$given that both $a_1$and $a_2$are of numeric types.

Authors	Language
Dan Brown	English
Richard Dawkins	English
Friedrich Nietzsche	German
Stephen Hawking	English
Charles Baudelaire	French
Charles Darwin	English

Table 2.2: Projection the relation book over the attributes :Authors and Language

No	Title	Page-Number	Genre	Authors	Year	Language
1	The Da Vinci Code	597	Mystery	Dan Brown	2009	English
2	Inferno	482	Conspiracy	Dan Brown	2013	English
3	The God Delusion	464	Philosophy	Richard Dawkins	2006	English
5	A Brief History of Time	256	Cosmology	Stephen Hawking	1988	English
7	The Origin of Species	502	Biology	Charles Darwin	1859	English

Table 2.3: Select all books written in English

Example: Using the book relation in table 2.1, the projection $\Pi_{Authors,Language}$will give the relation depicted in table 2.2.

Both dimension and data reduction occurred in this example. Only 2 out of 7 columns remained and one duplicated row was discarded. Performing the projection over the "language" attribute will drop more duplicated rows leaving only 3.

2.3 Selection

With selection it goes about rows, it means picking certain rows. Selection is a unary operation that eliminates all the records that don't fulfill a specific condition. The result will be a relation with the same schema as the original one but with fewer (subset) tuples.

Definition. Selection: $R_1 := \sigma_C(R)$. Where R is the original relation, $R_1$is the selection result and$C$ is a condition. The condition is an expression that involves some or all the attributes in R, it composes pairs $(Attribute - Name\ \Theta\ Attribute - value)$ where Θ is an operand closed with respect to the attribute domain [35, Chapter 4 : Relational Algebra - Pages 60-61].

Example: Using the book relation in table 2.1, the selection $\sigma_{Language="English"}$ will give the relation depicted in table 2.3. It is a query to retrieve all books written in English. Data reduction occurred here with two discarded rows. In this example it is also possible to discard the *Language* field (dimension reduction) since it is already know and used in the condition of the selection and it has the same value across all the selected rows in the new relation. Using the condition $language =$ "$German$" or $Language =$ "$French$" will lead to more data reduction since more rows will be dropped in the resulting relation.

2.4 Aggregation

Definition. Data Aggregation: is a type of data mining process in which scattered data (from one source or more) is searched, gathered and presented in a report-like summarized form, for purposes such as statistical analysis [37][29]. Finding totals, counting records, calculating averages and many descriptive measures can be achieved through data aggregation. Data aggregation plays a critical rule in the world of business analysis, it can help to give insights about

Language	Count
Egnlish	5
German	1
French	1

Table 2.4: Number of books written in each language

the trends dominating certain groups. Data aggregation involves two basic steps:

- **Data Grouping** identifies one or more data groups based on values in the selected features.
- **Data Aggregation** puts together (aggregates) the values in one or more selected columns for each group.

A query language such as *Sql* supports data aggregation and provides the so-called "Aggregation Functions" that work on the values under the selected columns to produce one single value. These functions are: [41]

- $AVG()$ - Returns the average value
- $COUNT()$ - Returns the number of rows
- $FIRST()$ - Returns the first value
- $LAST()$ - Returns the last value
- $MAX()$ - Returns the largest value
- $MIN()$ - Returns the smallest value
- $SUM()$ - Returns the sum

*Sql*also provides "group by" to better view and organize the aggregation results. It is used in conjunction with the aggregation functions and helps to group the aggregated data by one or more columns [40].

Example: Back to the book relation in table 2.1, the aggregation function *count*() can be used to get the number of books written in each language as shown in table 2.4.

The same can be done to get the number of books in each genre, or the number of books written by each author. In case of a book inventory system, aggregation could help to get insight about the reading trends among people who borrow the books.

2.5 Contrasting Relational and Multi-Dimensional Models: An Example

Figure2.5 shows the sales volumes dataset for the Gleason automobile dealership using the relational model, with the SALES VOLUMES field holding the data we wish to analyze, while the COLOR and MODEL fields contain the perspectives we will analyze the data from [4].

By examining the table:

- The first record in the table tells us that six sales were made for blue mini vans.
- The field MODEL ranges across only three possible values: MINI VAN, SPORTS COUPE and SEDAN.

MODEL	COLOR	SALES VOLUME
MINI VAN	BLUE	6
MINI VAN	RED	5
MINI VAN	WHITE	4
SPORTS COUPE	BLUE	3
SPORTS COUPE	RED	5
SPORTS COUPE	WHITE	5
SEDAN	BLUE	4
SEDAN	RED	3
SEDAN	WHITE	2

Figure 2.1: SALES VOLUMES FOR GLEASON DEALERSHIP

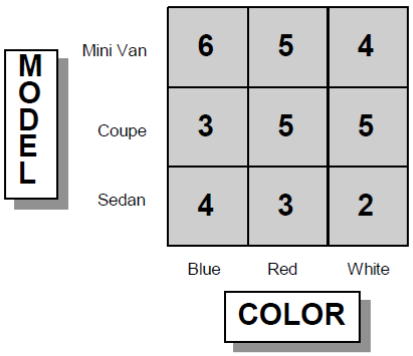


Figure 2.2: Sales Volumes as Matrix

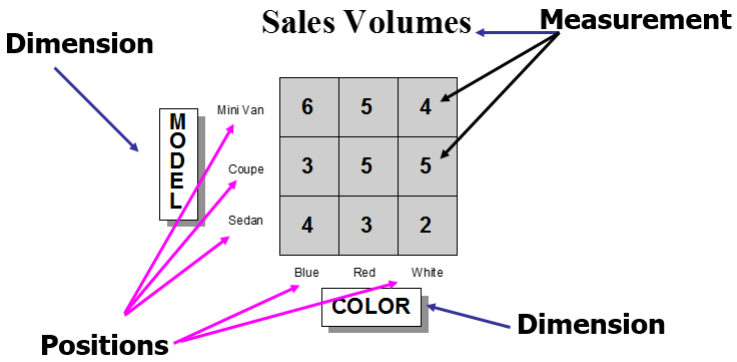


Figure 2.3: Multidimensional Structure

- The field COLOR also ranges across three possible values: BLUE, RED and WHITE.

Another way to represent data is using a “Cross Tab” View or a data matrix.

The matrix shown in figure 2.5 is an example of a two-dimensional “array.” An array is the basic component of a multidimensional database. In the array:

- Each axis is called a dimension (a data perspectives)
- Each element within a dimension is called a position.

In this example, the data has two dimensions, the first one is MODEL; it has three positions: MINI VAN, SEDAN and COUPE. The second is COLOR; it also has three positions: BLUE, WHITE and RED. The Sales Volume figures are located at the intersections of the dimension positions. These intersections are called cells and are occupied with our the data itself, as figure 2.5 shows:

The array is more efficient and effective for presenting the Sales Volume dataset than the relational table:

- A great deal of information is gleaned immediately upon direct inspection of an array. In this example, it is much easier to recognize immediately that there are exactly two dimensions of three positions each, while in the relational table it would need a deeper inspection.
- The array conveniently groups "like" information in columns and rows. In this example, all Sales Volume figures for SEDANs are lined up. It is so easy to get their total or compare them in order to get an idea which SEDAN COLOR is more popular. The same way we can determine which model is more demanded.

The multidimensional array structure offers a higher level of organization than the relational table. It depicts much better the relationships between the data elements, since our data "perspectives" are embedded directly in the structure as dimensions not just placed into fields. For example, the structure of the relational table can only tell us that there are three fields: COLOR, MODEL and DEALERSHIP. It tells us nothing about the possible contents of those fields. The array, on the other hand, tells us not only that there are two dimensions, COLOR and MODEL, but it also presents all possible values of each dimension as positions along the dimension. Bottom line, the array structure makes data browsing and manipulation highly intuitive to the end user. It is an intelligent structure that comes in handy for better data analysis.

This chapter introduced the first set of reduction techniques natively exist in the relational model to perform queries on the dataset. The projection was used as a mean of dimensionality reduction while selection targeted the data itself i.e. the rows. Aggregation however targeted the both, dimensions and data with new dimensions introduced as circulations of other dimensions. As mentioned earlier projection will sometimes affect entire data rows since it will delete any duplicated records. Next chapter will delve into the techniques used in multimedia information retrieval systems and how they may be suited to fit the goal of dimensionality reduction.

Chapter 3

Multimedia Information Retrieval

When working with a traditional DBMS, queries have to exact i.e. based on the notion of "equality". However multimedia data such as video, music and image files can't be approached the same way used with numeric or textual data. The traditional index structures are not suited to work with this type of data that typically has no natural order. There is still a gap between what computer systems can index and the high-level human concepts. Since everything in the digital world is stored as bits and not all of these bits are equally-important, we can use the less-significant bits in a multimedia document to store what is necessary to make this document more "searchable" rather than just relying on its title alone. Multimedia search engines can make use of the meta-data embedded in multimedia files i.e. creating what can be refereed to as **text surrogates** for multimedia. However this is not enough to perform a content-based information retrieval in many scenarios, scenarios in which the query itself is a multimedia excerpt. What about questions like: Can doctors submit scans of a patient to identify medically similar images of diagnosed cases in a database? Can we retrieve the right information about some musical work having only a piece of it? Can I take a photo of some landmark then use that photo to search for information about that landmark? These queries are relatively vague or fuzzy and instead of the notion of "equality", the query will be based on the notion of " similarity". Such queries are typically approached though a special type of search called **Similarity Search**.

3.1 Similarity Search

Definition. Similarity Search: is the process of finding the record(s) in a large dataset that are the closest to a given query. The closeness here is based on the similarity among objects, and the dataset would be typically a large space of objects [17][32]. Similarity search methods can be employed in a variety of applications including:

- Content-based retrieval
- Data-mining and pattern recognition
- Geographic Information Systems (GIS)

Similarity Measures: In order to retrieve an image based on its content, it is necessary to extract the features i.e. characteristics of the image and index this image based on these features. Such features can be histograms, shape descriptions, texture properties and so on. The various aspects of each feature can be described using a few qualitative measures. So a feature will be represented as a multidimensional vector with each component denoting a different feature

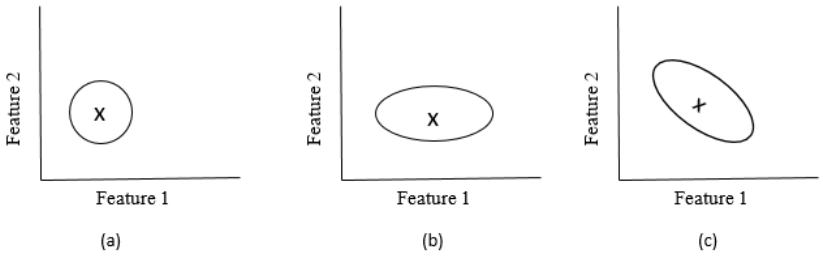


Figure 3.1: W (a) when A=I (b) W when A=diagonal (c) W when A=non-diagonal and symmetric

measure. A feature F comprising n feature measures can be represented as [24, Chapter 2, Pages 299-301]:

$$F = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ f_n \end{bmatrix}$$

Example: The texture attribute of an image can be modeled as a three-dimensional vector with measures of directionality, contrast and coarseness. There will be a need for multidimensional indexing methods in order to index these image features.

A suitable measure of similarity between an image feature vector F and a query vector Q is the weighted metric$W$:

$$W = (F - Q)^T \cdot A \cdot (F - Q)$$

where A is an $n \times n$ matrix which can be used to specify suitable weighting measures such as those which take into account between various feature components or which emphasize one feature component over the other while calculating the similarity. In general A is one of the following:

1. $A = I$: A is the identity matrix such that W corresponds to the normal Euclidean distance. The points that lie at the same distance from the query point are all equally similar. in a 2-D feature space this similarity measure can be represented as a circle as shown in figure 3.1(a).

2. $A = diagonal$: This corresponds to the weighted Euclidean distance measure where different components of the feature Fhave different weights associated with it. In this case, equal similarity corresponds to points along an ellipsoidal contour which has its major and minor axes oriented parallel to the feature axes as shown in figure 3.1(b). Such as similarity measure is useful for attributing differing importance to the various components of Fas well as for normalizing the distance calculated for each component of F by taking into account the distribution of values along that axis.

3. $A = non - diagonal$ and mainly symmetric: This case takes into account the cross-correlation between the various components of feature $F$and can be represented as an ellipsoid which is oriented at an angle to the feature axes as shown in figure 3.1(c).

In general, there are three query types in similarity search:

- ε-search: The distance tolerance is defined. It will be used at the earlier stage and can be very "loose".
- K-nearest-neighbor-queries: The user specifies the number of close matches to the given query object.
- Range Queries: An interval is given for each dimension of the feature space and all the records which fall inside this hypercube are retrieved.

3.2 Nearest Neighbor Queries

These queries are particularly useful in situations like:

- Finding the nearest object to a given point e.g. Given a star, find the 5 closest stars.
- Finding the closest object given a range e.g. Find all stars between 5 and 20 light years of a given star.
- Spatial joins e.g. Finding the three closest restaurants for each of two different movie theaters.

The pruning property alone does not guarantee 100% recall, thus other properties have to hold.

Definition: Proximity-preserving-property: preserve ordering of the observed objects in the transformed space. Given: d,d' the distance metrics in the original and transformed space respectively and the Transformation f.

$d(a,b) \leq d(a,c) \rightarrow d'(f(a),f(b)) \leq d'(f(a),f(c))$ for any objectsa, b, and$c$ [32, Chapter 4, Page 665].

3.3 Using Only One Dimension

In this scenario, the original data dimension is known e.g. data is represented as feature vectors in a high-dimensional space. The simplest technique will focus on selecting a subset of features i.e. ignoring some of the features and retaining the most discriminating ones.

3.3.1 The K-Nearest Neighbor Algorithm

The most drastic and easiest to implement method such as the K-Nearest Neighbor Algorithm (Friedman, Baskett and Shustek) will work as follows [32, Chapter 4, Page 669]:

- Use just one of the given features without applying any transformation.
- Feature f has been chosen $\rightarrow$ All objects are sorted with respect to this featuref-distance.
- Given q the query object, the k-nearest neighbors are found by processing the objects in an increasing order of their f-distance from q.
- Stop processing when encountering an object o with f-distance from q greater than actual distance from q to the nearest kth-neighbor so far.

One major issue (drawback) with this algorithm is that many objects may be represented by the same point. The efficiency of the k-nearest neighbor algorithm of Friedman et al.[649] depends, in part, on which feature f is used. This feature f can be obtained globally or locally.

Global Perspective: f is the feature with the largest range (spread) of values. In such case we have to examine all the objects before starting the search.

Local Perspective: f is the feature with the largest expected range of values about the value of the query object q (q_f). In this case objects have to be sorted with respect to all features. The local density around q depends of the expected number N' of objects that will be examined during the search. Friedman obtains N' from the radius of the expected search region by using a uniform distribution. The local density of feature i is determined by calculating the size of the range containing the $\frac{N'}{2}$ values less than or equal to q_i and the $\frac{N'}{2}$ values greater than or equal to q_i and choosing f as the one with the largest range.

Friedman et al. vs. Brute force algorithm: Friedman et al. is considerably more efficient when the dimension of the underlying data is relatively small, however the brute force will do much better when the dimension of the underlying data exceeds 9.

3.3.2 Representative Feature

This method combines different features into one by using some information from each of the features [32, Chapter 4, Page 669].

Example: Given that each object is represented by n different features, each has 64-bit value. Represent a single number by concatenating the values of the Most Significant Bit MSB from each of the n different features.

Drawback: As with some other methods that reduce to only one-dimension, many objects will be represented by the same point.

3.3.3 Space-Ordering-Approach

Uses one of the space-ordering methods such as **Morton and Peano-Hilbert orders**. A major drawback with this approach is that the **Pruning proberty** does not hold [32, Chapter 4, Page 669].

3.4 Representative Point Methods

Transform a spatial object to a "Representative Point" in the space of the same or higher dimension (with respect to the space from which they are drawn). Small-Sized Feature vectors are used with representative features of the object that will serve as the basis of the feature vector [32, Chapter 4, Pages 670-671].

Example: Represent a t-dimensional object by

- Its centroid $\rightarrow$ t-features
- Axis-aligned minimum bounding rectangle $\rightarrow 2 \cdot t$ features corresponding to the coordinate values of two diagonally opposite corners.
- Minimum bounding sphere $\rightarrow t + 1$ features corresponding to the coordinate values of the centroid plus the magnitude of the radius.

Dimension-reduction method: The number of features used to represent the object has been reduced in comparison with the feature-per-pixel method used to indicate the space occupied by the object.

Drawback: Pruning Property will not hold.

As mentioned earlier, we live in the age of an unprecedented data big ban and the datasets continue to expand. In this big ban, the multimedia data may have the lion's share, which makes the content-based retrieval methods essential to keep up with the increasing demands in the enterprises. The techniques mentioned in this chapter have their pros and cons. While some will reduce too much and sacrifice a lot of information, others may result a bad precision since the pruning property will not hold. The next chapter will cover two mathematical transformations that work solely on numerical data. These transformations are statistical methods that help reducing the number of attributes in a data set.

Chapter 4

Mathematical Transformations

If there exist a subset of features that discriminate best between the data, it would be preferable to base the spatial index solely on them. In such a case the methods will depend on the data domain.

As an alternative, we could transform the features into another set of more relevant i.e. discriminating features. The data will be transformed so that most of the information is concentrated in a small number of features. Two techniques will be covered in this chapter, Principal Component Analysis and Singular Value Decomposition.

4.1 Principal Component Analysis

PCA is a simple yet popular and useful statistical linear transformation technique. The main goal of a PCA analysis is to identify patterns in data; PCA aims to detect the correlation between variables. If a strong correlation between variables exists, the attempt to reduce the dimensionality only makes sense. In a nutshell, finding the directions of maximum variance in high-dimensional data and project it onto a smaller dimensional subspace while retaining most of the information. PCA can be used in a variety of applications including Face Recognition, Image Compression and Data Dimensionality Reduction [25].

Definition. Principal Components: are the underlying structure in data. They are the directions where there is the most variance, the directions where the data is most spread out. It is often useful to measure data in terms of its principal components rather than on a normal x-y axis. Following is a simple example that shows how the principle components can be detected in a dataset.

4.1.1 Identifying the Principal Components in a Dataset

Given a set of data represented as triangles forming the shape of an ellipse as shown in figure 4.1.1 [5][12].

To find the direction where there is most variance, find the straight line where the data is most spread out when projected onto it. A vertical straight line (the small diameter of the ellipse) with the points projected on to it will look like the one in figure 4.1.1:

The data is not very spread out here, therefore it doesn't have a large variance. It is probably not the principal component.

A horizontal line (the large diameter of the ellipse) with lines projected on it will look like the one in figure 4.1.1:

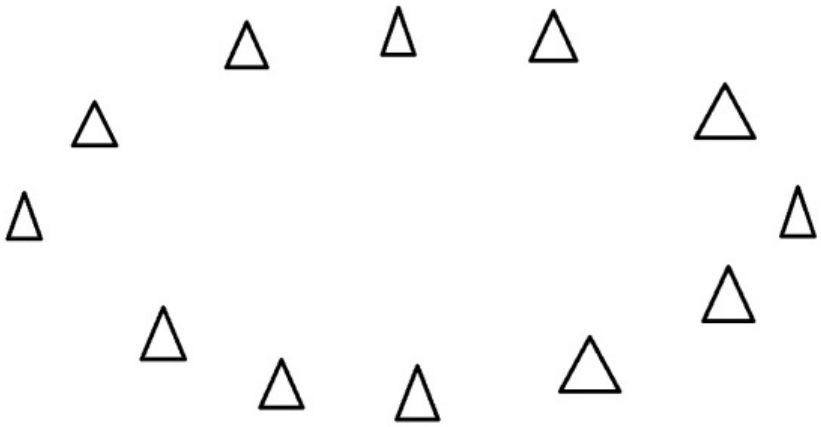


Figure 4.1: Dataset represented as triangles on oval

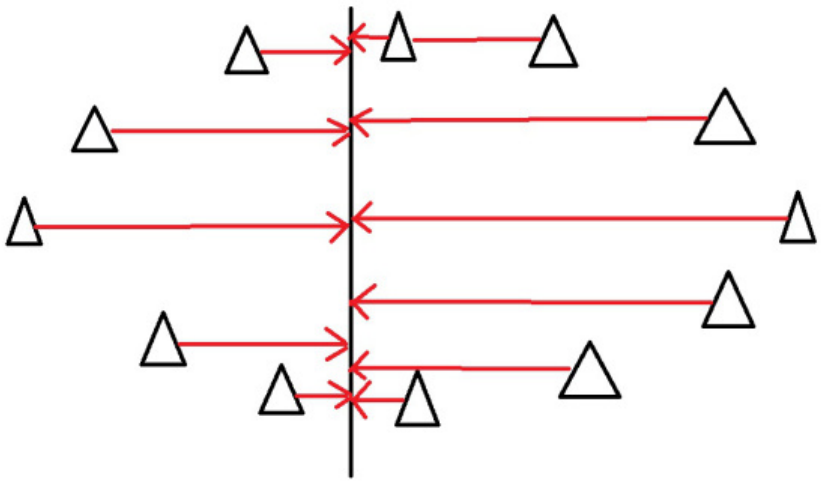


Figure 4.2: Projection on the small diameter of the ellipse

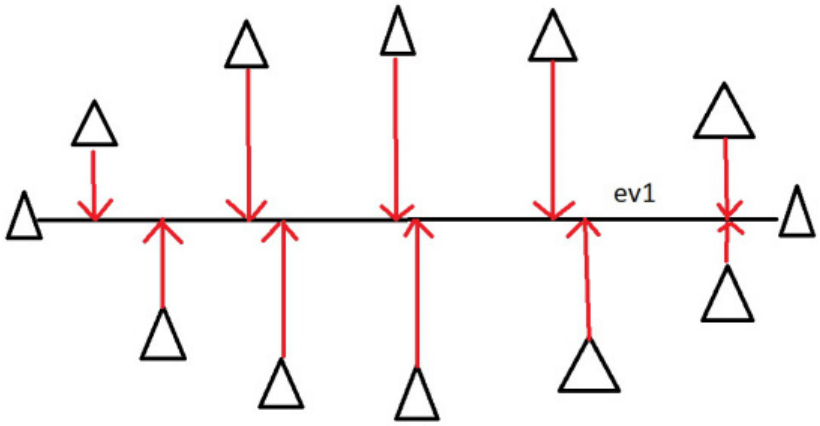


Figure 4.3: Projection on the large diameter of the ellipse

On this line the data is way more spread out, it has a large variance. In fact there isn't a straight line you can draw that has a larger variance than a horizontal one. A horizontal line (the large diameter) is therefore the principal component in this example.

4.1.2 Eigenvectors and Eigenvalues

A data set can be deconstructed into eigenvectors and eigenvalues. Eigenvectors and values exist in pairs: every eigenvector has a corresponding eigenvalue. An eigenvector is a direction; In the example above the eigenvector was the direction of the line (vertical, horizontal, 45 degrees etc.). On the other hand, an eigenvalue is a number, telling us how much variance there is in the data in that direction. In the example above the eigenvalue is a number telling us how spread out the data is on the line. The eigenvector with the highest eigenvalue is therefore the principal component [5].

The amount of eigenvectors/values that exist equals the number of dimensions the data set has. Eigenvectors put the data into a new set of dimensions, and these new dimensions have to be equal to the original amount of dimensions.

Figure 4.1.2 shows the ellipse on an x-y axis. X and Y could be any two variables in a dataset. These are the two dimensions that the dataset is currently being measured in.

4.1.3 Dimension Transformation

The principal component of the triangle ellipse is a line splitting it longways (the large diameter) as shown in figure 4.1.3[5].

In a 2-D dataset there will be two eigenvectors. The other eigenvector in this example is perpendicular to the principal component (the small diameter). Eigenvectors have to be able to span the whole x-y area. In order to do this (most effectively) the two directions need to be orthogonal (i.e. 90 degrees) to one another. Figure 4.1.3shows both eigenvectors.

The eigenvectors can give us a much more useful axis to frame the data in. We can now re-frame the data in these new dimensions as figure 4.1.3shows.

Notice that nothing has been done to the data itself. We're just looking at it from a different angle. Getting the eigenvectors gets you from one set of axes to another. These axes are much

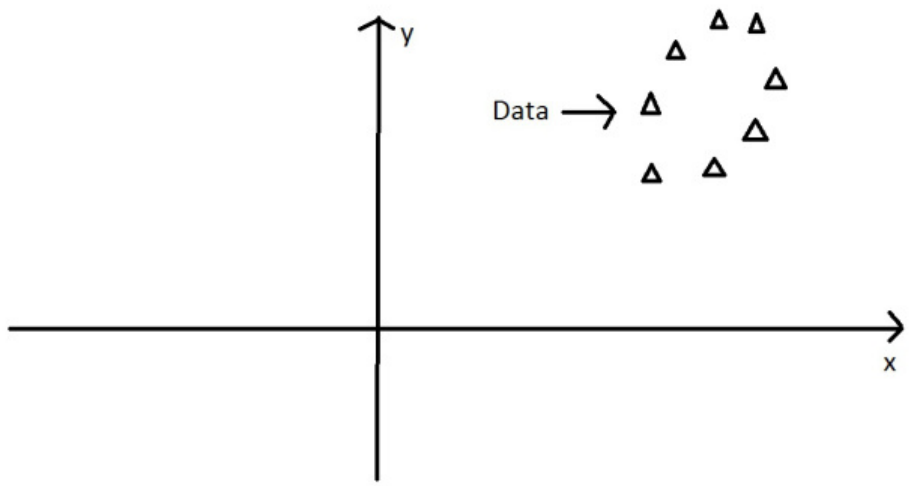


Figure 4.4: The dataset represented on x-y axis

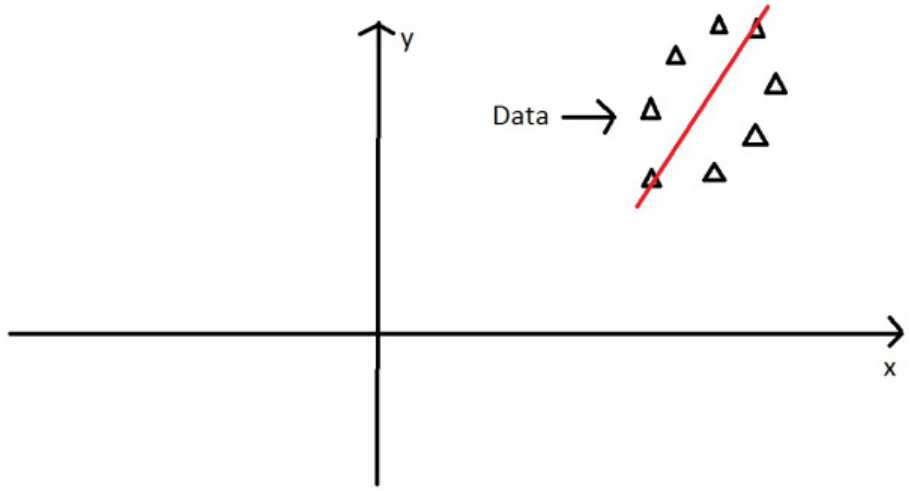


Figure 4.5: The principal component of the dataset on x-y axis

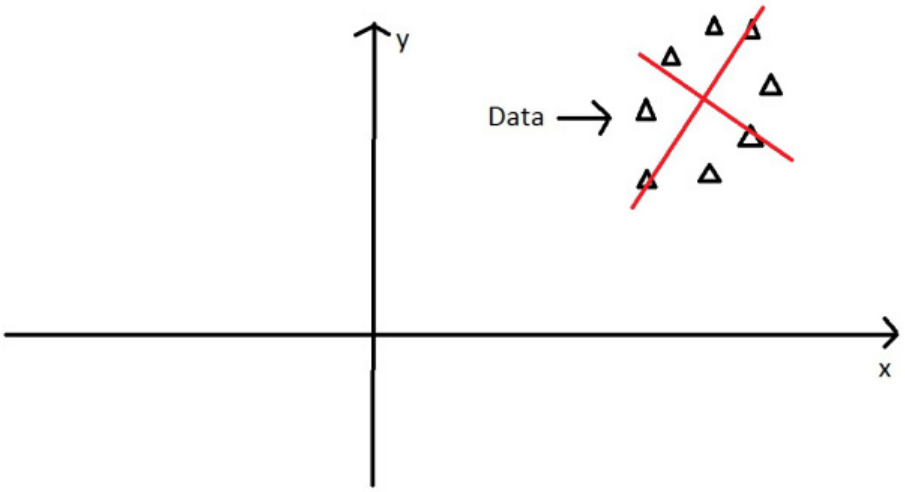


Figure 4.6: Two orthogonal eigenvectors

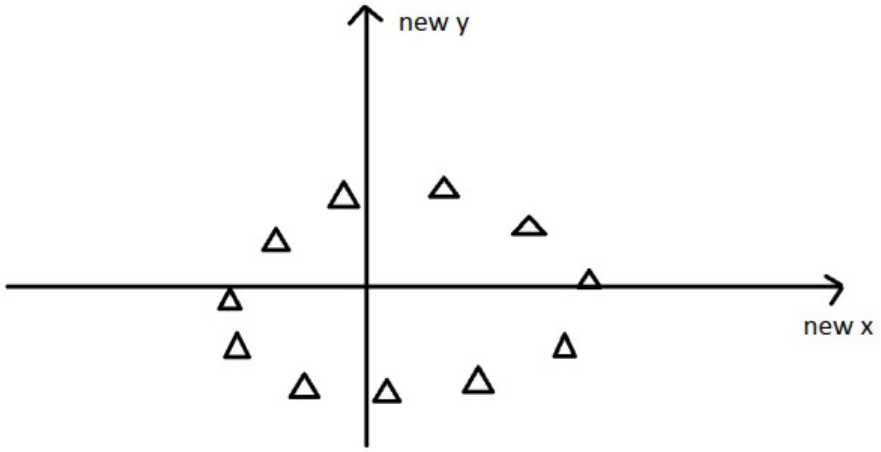
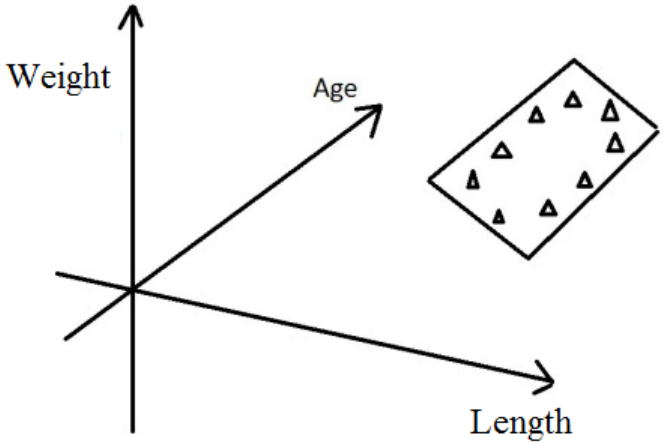


Figure 4.7: Dataset represented using the new dimensions

Figure 4.8: Dataset on a plane within a 3-D graph



more intuitive to the shape of the data. These directions are where there is most variation, and that is where there is more information.

Thinking about this the reverse way round: If there was no variation in the data [e.g. everything was equal to 1] there would be no information. In this scenario the eigenvalue for that dimension would equal zero, because there is no variation.

4.1.4 Dimension Reduction

In case we are measuring three things, let's say: age, weight and length, then there are 3 variables so it is a 3D dataset. 3 dimensions is an x,y and z graph. Imagine that the data forms into an ellipse like the one above, but this ellipse is on a plane. i.e. all the data points lie on a piece of paper within this 3D graph (having width and depth, but no height), as shown in figure 4.1.4[5]:

When we find the 3 eigenvectors/values of the data set (3D problem = 3 eigenvectors), 2 of the eigenvectors will have large eigenvalues, and one of the eigenvectors will have an eigenvalue of zero. The first two eigenvectors will show the width and depth of the data, but because there is no height on the data (it is on a piece of paper) the third eigenvalue will be zero. Figure 4.1.4 shows the three eigenvectors of the dataset: ev1 is the first eigenvector (the one with the biggest eigenvalue, the principal component), ev2 is the second eigenvector (which has a non-zero eigenvalue) and ev3 is the third eigenvector, which has an eigenvalue of zero.

We can now rearrange our axes to be along the eigenvectors, rather than age, weight and Length. The ev3, the third eigenvector, is pretty useless. Instead of representing the data in 3 dimensions, we can get rid of the useless direction and only represent the dataset in 2 dimensions, like before as figure shows:

It is worth mentioning that we can reduce dimensions even if there isn't a zero eigenvalue. Back to the same example above: Let us suppose that instead of the ellipse being on a 2D plane, it had a tiny amount of height to it. There would still be 3 eigenvectors, however this time all the eigenvalues would not be zero. Let us suppose that the values would be something like 10, 8 and 0.1. The eigenvectors corresponding to 10 and 8 are the dimensions where there is a lot of information. The eigenvector corresponding to 0.1 will not have much information at all, so we can therefore discard the third eigenvector again in order to make the dataset much simpler. The most insignificant eigenvectors (paired with too small eigenvalues compared to the others) can be discarded without getting rid of much of the information.

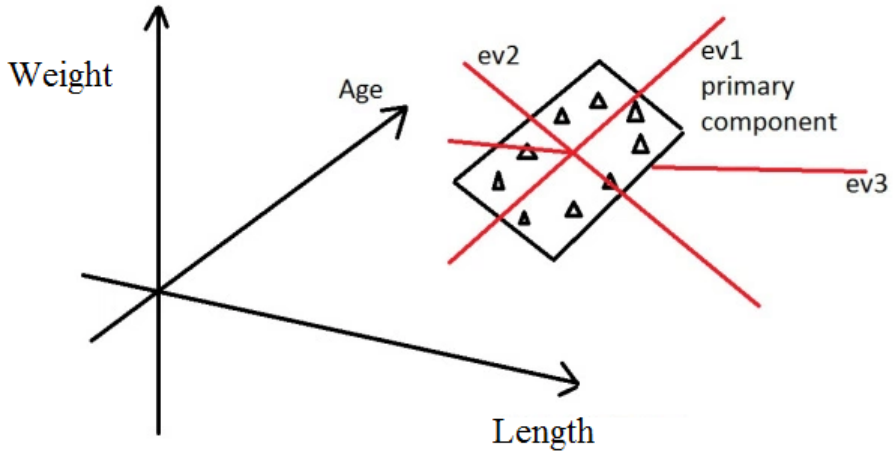


Figure 4.9: The three eigenvectors of the dataset in the new dimensions

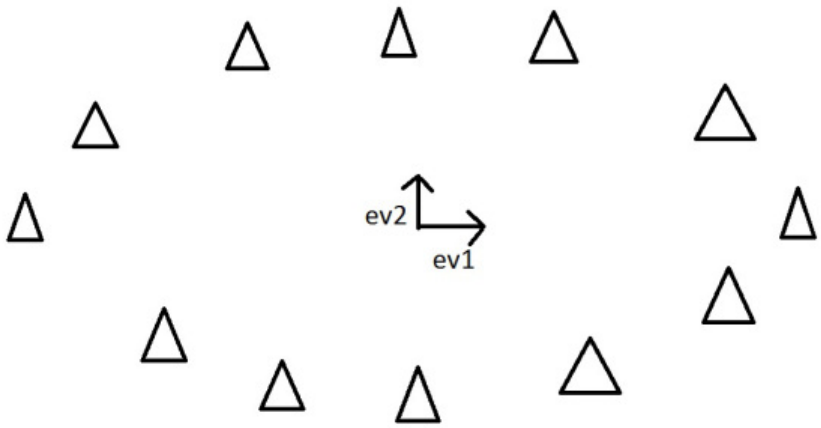


Figure 4.10: The 3-D dataset is now represented using two dimensions only

4.1.5 PCA vs. LDA

It is important to distinguish PCA from another technique called LDA or Linear Discriminant Analysis [33][26].

- Both are linear transformation methods.
- PCA yields the directions (principal components) that maximize the variance of the data.
- LDA also aims to find the directions that maximize the separation (or discrimination) between different classes, which can be useful in pattern classification problem.
- PCA "ignores" class labels.
- PCA projects the entire dataset onto a different feature (sub)space.
- LDA tries to determine a suitable feature (sub)space in order to distinguish between patterns that belong to different classes.

4.1.6 Background Mathematics

Basic Statistics [1]

Given a sample X:

- The Mean
$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$
- The Standard Deviation (SD) of a data set is a measure of how spread out the data is.
$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$$
- Variance is another measure of the spread of data in a data set. In fact it is almost identical to the standard deviation.
$$S^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

4.1.6.1 Covariance

Standard Deviation and variance are 1-dimensional. In case of more than 1-dimension, it would critical to consider any relationship between the dimensions. While standard Deviation and variance are computed for each dimension of the data set independently from each other, **Covariance** can be used to address a relationship between the dimensions. Covariance is a measure to find out how much the dimensions vary from the mean with respect to each other. If you calculate the covariance between one dimension and itself, you get the variance. Covariance is given in the following equation:

$$Conv(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

The exact value of the covariance is not as important as it's sign (i.e. Positive or negative).

[1] Crash Course on Basic Statistics - Marina Wahl - University of New York at Stony Brook - November 6, 2013

- If the value is positive: Both dimensions increase together.
- If the value is negative: As one dimension increases, the other decreases.
- If the covariance is zero: The two dimensions are independent of each other.

This technique is often used to find relationships between dimensions in high-dimensional data sets. Covariance is symmetric: Cov(X,Y) = Cov(Y,X)

For an n-dimensional data set you can compute $\frac{n!}{2.(n-2)!} = n \cdot \frac{n-1}{2}$

Definition: Covariance matrix for a set of data with n dimensions:

$$C^{nxn} = (c_{i,j}, c_{i,j} = conv(Dim_i, Dim_j))$$

For three data sets X,Y,Z

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

It is obvious that the diagonal values represent a variance between a dimension and itself, and since the covariance is symmetric so it will be the resulted matrix.

Matrix Algebra All the necessary mathematical operations concerning the eigenvalues and eigenvectors will be presented in the following sections [14].

4.1.6.2 Eigenvectors and Eigenvalues

Definition: Let A be an $n \times n$ matrix, a scalar λ is called an eigenvalue of A if there is an non-zero vector X such that $AX = \lambda X$. The vector X is called an eigenvector of A corresponding to λ.

Example:

For matrix $A = \begin{bmatrix} 3 & 2 \\ 3 & -2 \end{bmatrix}$ there is a vector $X = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ that fits as an eigenvector of A corresponding to the eigenvalue $\lambda = 4$

If λ is an eigenvalue of A and X is an eigenvector belonging to λ, then any non-zero multiple of X will be an eigenvector. In the previous example: Since $X = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is an eigenvector then: $X = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$ or $X = \begin{bmatrix} 20 \\ 10 \end{bmatrix}$ ... are also eigenvectors. This can be easily proved since: $A \cdot (c \cdot X) = c \cdot A \cdot X = c \cdot \lambda \cdot X = \lambda \cdot (c \cdot X)$

4.1.6.3 Finding Eigenvalues and Eigenvectors

Given A, an $n \times n$ matrix

1. Multiply an $n \times n$ identity matrix by the scalar λ
2. Subtract the result of the multiplication in step 1 from the matrix A
3. Find the determinant of the matrix resulted in step 2 and the difference
4. Solve for the values of λ that satisfy the equation: $det(A - \lambda \cdot I) = \vec{0}$; I is the identity matrix
5. Solve for the corresponding vector for each λ

Example: $A = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix}$

- Step1: $\lambda.I = \lambda \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$
- Step2: $A–\lambda \cdot I = \begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix}$
- Step3: $det(\begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix}) = (7-\lambda) \cdot (-1-\lambda)–3 \cdot 3 = \lambda \cdot 2 - 6 \cdot \lambda - 16$
- Step4: Solve the equation: $\lambda^2 - 6\lambda - 16 = 0$ for λ
- We have two eigenvalues: $\lambda = 8$ and $\lambda = -2$
- From step 2: $A–\lambda \cdot I = \begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix}$. For $\lambda = 8 \rightarrow$We get the matrix $B = \begin{bmatrix} -1 & 3 \\ 3 & -9 \end{bmatrix}$
- Now we have to solve: $B \cdot X = \vec{0} \rightarrow \begin{bmatrix} -1 & 3 \\ 3 & -9 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightarrow X = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$

4.1.7 Method

PCA is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Once you have found these patterns in the data and you can compress the data, i.e. by reducing the number of dimensions, without much loss of information. Often, the desired goal is to reduce the dimensions of a d-dimensional dataset by projecting it onto a k-dimensional subspace (where k<d) in order to increase the computational efficiency while retaining most of the information.

Important Question: what is the size of k that represents the data "well"? This can be either defined by the user or as recommended generated using the MLP neural network discussed in chapter 1.

To perform PCA Follow these steps [12]:

- Standardize the data (in case the data was measured on different scales for instance).
- Obtain the Eigenvectors and Eigenvalues from the covariance matrix or correlation matrix, or perform Singular Vector Decomposition.
- Sort eigenvalues in descending order and choose the k eigenvectors that correspond to the k largest eigenvalues where k is the number of dimensions of the new feature subspace.
- Construct the projection matrix$W$ from the selected K eigenvectors.
- Transform the original dataset X via W to obtain a k-dimensional feature subspace$Y$.

Example:

Given a two-dimensional data set. At first a data adjustment has to be performed: Subtract the mean of each dimensions from the corresponding values. This produces a data with a mean equal to 0.

Data =

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

Data-Adjusted =

X	Y
0.69	0.49
-1.31	-1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
0.71	-1.01

PCA will be performed as follows:

- Calculate the covariance matrix $Cov = \begin{bmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{bmatrix}$
- $Eigenvalues = \begin{bmatrix} 0.0490833989 \\ 1.28402771 \end{bmatrix}$
- $Eigenvectors = \begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$

Choosing components and forming a feature vector

- The eigenvector with the *highest* eigenvalue is the principle component of the data set.
- We can choose to leave out the smaller, less significant component and only have a single column.
- Projection matrix $W = \begin{bmatrix} -0.677873399 & -0.735178656 \end{bmatrix}$

Deriving the new data set

$FinalData = W.Data - Adjusted^T$

The transformed data set will be:

X
-0.827970186
1.77758033
-0.992197494
-0.274210416
-1.67580142
-0.912949103
1.14457216
0.438046137
1.22382056

4.2 SVD: Singular Value Decomposition

Principal. Find a linear transformation of n-dimensional feature vectors to k-dimensional feature vectors (k≦n) that minimizes the sum of the squares of the Euclidean distances between the set of n-dimensional feature vectors and their corresponding k-dimensional feature vectors. An alternative characterization would be to minimize the mean square error [32, Chapter 4, Page 671].

There are two equivalent techniques: KLT (Karhunen-Loeve Transform) and PCA (Principal Component Analysis). The goal of all these techniques is to Find the most important features i.e. a linear combination of features for a given set of feature vectors. In an nutshell it is a transformation of the original set of feature vectors$S$ to obtain a new set of feature vectors $S\prime$.

The individual features that make up $S\prime$ are ranked by their importance. The less important features will be ignored by projecting onto the most important features $\rightarrow$ This would preserve much of the variation between the elements of the original set of feature vectors S.

4.2.1 Method

Given: a set of m $n-dimensional$ feature vectors $f_1, f_2, f_3, ..., f_m$ that are to be transformed into a set of m $k-dimensional$ vectors $t_1, t_2, t_3, ..., t_k$. The two sets can be written as an $m \times n$ matrix F and an $m \times k$ matrix T

$$F = \begin{bmatrix} f_{11} & f_{12} & .. & .. & f_{1n} \\ f_{21} & f_{22} & .. & .. & f_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ f_{m1} & f_{m2} & .. & .. & f_{mn} \end{bmatrix} \text{ and } T = \begin{bmatrix} t_{11} & t_{12} & .. & .. & t_{1k} \\ t_{21} & t_{22} & .. & .. & t_{2k} \\ . & . & . & . & . \\ . & . & . & . & . \\ t_{m1} & t_{m2} & .. & .. & t_{mk} \end{bmatrix}$$

The $m \times n$ $matrix$ F; $m > n$ can be written as a product of three matrices U, Z, V as follow [32, Chapter 4, Page 672]:

$F = UZV^T(SingularValueDecomposition)$

- U is an $m \times n$ matrix, a set of n orthonormal column vectors u_i i.e. $u_i.u_j = \delta_{ij}$
- $U^TU = I_m$
- Z is an $n \times n$ singular value matrix with nonnegative elements along its diagonal known as singular values

$$Z = \begin{bmatrix} \delta_{11} & 0 & .. & .. & 0 \\ 0 & \delta_{22} & .. & .. & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & 0 & .. & .. & \delta_{nn} \end{bmatrix}$$

- V is an $n * n$ orthonormal matrix ; $VV^T = I_n$ and $V^TV = I_n$

The result of taking the singular value decomposition of a matrix F is the determination of a set of orthonormal column basis vectors $v_1, v_2, \ldots, v_n$. These basis vectors have the property that the original m feature vectors are spread out most widely along the v_1 direction, then along the v_2 direction and so on. The Nonzero entries δ_{ii} in Z correspond to the minimum variance of the data in F along the basis vector in the v_i directions ($1 <= i <= n$), supposing that the mean is at the origin [32, Chapter 4, Page 673].

Apply the Transformation V to F, which rotates the vectors that make up F in the direction of the best least squares fit. A **line of best fit** is a straight line that is the best approximation of the given set of data (least square method)

$t_i = f_i.V$or $FV = T = UZ$

V is the SVD transformation matrix (same matrix used in PCA). Note that the lengths are preserved $|f_i| = |t_i|$; $v_i.v_j = 0; i \neq j$ as V is an orthonormal matrix and orthonormal transformation preserves the Euclidean norm. This transformation can rotate a point to a new location without changing its Euclidean distance form the origin. We reduce the dimension of

the feature vectors to k by retaining only the most significant k values in Z (largest ones). This is done by zeroing the remaining $n-k$ least significant values (smallest ones) and discarding the corresponding entries in T, U and V.

4.2.2 Mathematical point of view

- Compute the eigenvectors of the covariance matrix:

$$V_{ij} = cov(x_i, x_j) \equiv ((x_i - \mu_i)(x_j - \mu_j))$$

- Sort the eigenvectors in order of decreasing eigenvalue.
- Choose a value $k \leqq n$
- Approximate each of the original $n - dimensional$ feature vectors by its projection onto the first k eigenvectors.

Whether to use SVD or not depends on the data itself, it is most appropriate where the database is static (e.g. stored on a CD-ROM) or changes a little. However it is quite expensive, for instance having m feature vectors in an $n - dimensional$ space would require $O(m.n^2)$.The cost is linear in terms of m, but the values of constants are quite high. Another drawback is that regardless of the amount of data reduction, we compute the SVD transform matrix using the entire data.

With Dynamic Datasets, we have to reuse the existing SVD Transform matrix or recompute it. Reuse the matrix as long as the precision of the query responses doesn't degrade too much (5% to 10% is tolerable) [32, Chapter 4, 674].

Statistical methods such as PCA help to identify and visualize highly-correlated variables in a data set. Depending on the accompanying threshold, a number of these variables will be removed, while the attributes with the most spread data will be kept intact. PCA does that by transforming the data into new coordinates based on the data distribution. PCA is a simple, non-parametric exploratory method used in a variety of applications such as neuroscience, social sciences, market research and many industries that deal with high dimensional data. The next chapter will examine the Saliency-Maps model, a bio-inspired model used to extract a reduced version of an image that represents the most salient locations i.e. locations where data most varies and that variance is assessed from a visual-stimuli perspective.

Chapter 5

Computational Neuroscience Techniques

Attention and other cognitive tasks still pose a serious challenge to the computer systems. In terms of performance, although there have been significant developments in the past few years, the artificial vision systems are still struggling to perform a little bit of what the humans intuitively can [59]. While some artificial systems achieved relatively good results in a well-controlled environment, their performance is still very poor in real-world situations with varying lightning conditions and high-complex visual signals [66].

One way to overcome these limitations is to develop bio-inspired models i.e. using the same techniques the human brain employs to reach a human-like efficiency.

5.1 Visual Attention

Definition. Visual attention: or just attention is the cognitive process of selectively concentrating on one aspect of the environment while ignoring other things. Attention has also been referred to as the allocation of processing resources [53].

This definition indicates that in a complex visual field we can only attend to one thing at a time. In other words, the human awareness is limited to only a small portion of the information flow or the visual stimulus that affects our senses. Attention is what basically drives our senses in the visual scene.

Definition. Stimulus (plural stimuli)[1]**:** a Latin word refers to a thing or event that evokes a specific functional reaction in an organ or tissue.

Definition. Visual perception: is the ability to comprehend or make sense of what we attend i.e. paint a coherent scene in mind. The method we perceive and interpret the scene is a combination of attention, eye movements and memory [54][52][47].

In this regard, it is important to distinguish between **perception** and **sensation**. **Sensations** are un-interpreted sensory impressions detected by specialized receptor-cells in the sense-organs as physical energy (light, heat and sound), converted to electrochemical impulse or action potential, and passed through the nerve system to the brain where they get their meaning via the process of perception [60]. Visual attention and visual perception have been always the subject of heavy research, and until today the process is not completely understood. It is believed that attention may involve more than one neural system i.e. different cortical regions collaborate to accomplish a single sub-function in the process [46, "Overview", Introduction].

[1] Oxford Online English Dictionary: http://www.oxforddictionaries.com/definition/english/stimulus

Neurobiological research since 1980 has shown that visual perception does not only rely on the neural response triggered by individual objects in the visual field, but is also highly affected by the contextual sensory information and the behavior of the observer [46, Imaging Expectations and Attentional Modulations in the Human Brain].

In their book **The invisible Gorilla**, prof. Daniel J. Simons and Christopher Chabris conducted an experiment about the selective attention. In this experiment two groups of people, the first in white shirts and the second in black ones. The members of each group are passing a basketball between each other. At the beginning of the video you will be asked to count how many times the "players" wearing white shirts passed the ball. You will concentrate in order to count, and probably give the right answer at the end, however most likely you will never notice the one dressing as gorilla who passed the scene while you were busy counting. In this case, the task at hand drives the attention to ignore a portion in the visual scene that is supposed to be strange and usually attracts the attention. Watch the YouTube video of the experiment on prof. Simons' channel [62].

The brain employs a variety of visual attention mechanisms in order to serve two critical purposes [49]:

- Select behavior-based relevant information and drop the irrelevant i.e. the attention is directed to the objects of interest. This will result in significant reduction in the amount of information that will reach further processing stages in the brain.
- Alter or enhance the selected information according to the state and goals of the perceiver i.e. the perceiver can interact intelligently with the environment rather than being just a passive receiver of information.

5.2 Types of Attention

Attention is not an instant process, however it has been shown that it operates on two phases [71]. When perceiving a scene, a pre-attentive processing will first occur, a process in which the attention will be distributed equally over the entire visual field. The basic features of objects will be detected such as colors, closure, line ends, contrast, tilt, curvature and size. Pre-attentive processing is done quickly, easily and in parallel with no attention being focused on the display (Treisman, 1985, Treisman, 1986). In the next phase - serially done - the attention will be concentrated on a specific object in the visual scene. The sensory data representing the attended object will be transferred for further processing in higher levels, while the rest of the visual stimuli will be suppressed. Psychology recognizes three kinds of attention associated with learning [64]:

- **Sustained Attention:** is the ability to concentrate on one thing for an extended period of time. For instance, when a student concentrate on solving the questions in some exam.
- **Selective attention:** is to focus on one thing (object) at a time, and discard the rest of the visual stimuli (the other elements) in the scene. Selective attention is the typical behavior that humans usually accomplish efficiently [72]. This type of attention can also be noticed in the experiment done in Prof. Daniel Simons' video, when the observer fails to spot the invisible gorilla [62].
- **Divided Attention:** is to distribute focus or concentration among different objects in the visual scene. Due to limitations in our visual resources not all details will be captured and some objects will be missed when responding to a complex visual stimulus [72]. The attention will be divided among different objects i.e. only one object will be attended at a time. This is similar to a processor dividing its clock-time among different tasks in a multi-tasking single-processor system.

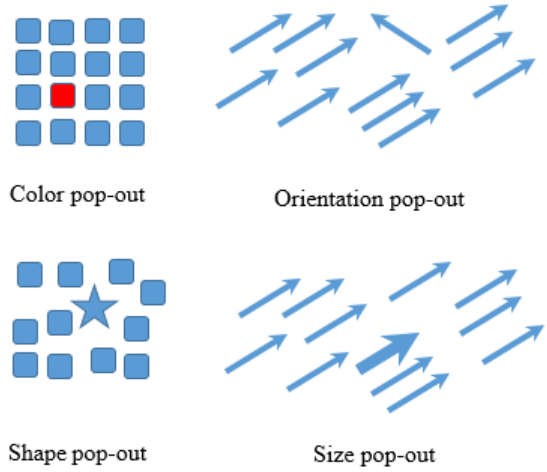


Figure 5.1: Pop-Out effect with different features

5.3 Visual Search

Definition. Visual search: is a perceptual process that requires attention, it involves an active scan of the visual scene seeking a target among distractors. The target can be an object or a feature of some object, while the distractors are non-target elements in the visual scene (other surrounding objects or features) [61][50].

Definition. Feature: is a distinctive property of the object, such as the color, size, orientation or shape.

As mentioned above, due to limitations in the process of visual attention, only a small portion of the visual scene can be processed at a time i.e. in order to achieve the search goals, several attentional deployments have be done. The Search goes on until the target is found or the search is aborted [67].

5.3.1 Types of Visual Search:

5.3.1.1 Feature Search

Also referred to as disjunctive search or efficient search. In this type of search, the target is defined with only one feature. Feature search can be performed fast and pre-attentive (in parallel), and doesn't depend on the set-size (number of items in the display) [69]. If the target is distinctive compared to its surrounding distractors then it gives the observer the impression that it pops-out of the display, this pop-out effect will speed up the search process [65]. The feature that causes the pop-out effect is called "Feature Singleton" [51]. Figure 5.3.1.1 shows some examples of the pop-out effect.

5.3.1.2 Conjunction Search

Also referred to as inefficient search. In this type of search, the target is defined with a combination of two or more features [68]. When compared with feature search, conjunction search is much slower, also harder, performed serially and depends linearly on the set-size [45]. Figure 5.3.1.2 shows an example of conjunction search. The search task in figure 5.3.1.2 is to find the orange square (two features: Shape = square & Color = orange) in a slightly-big group of

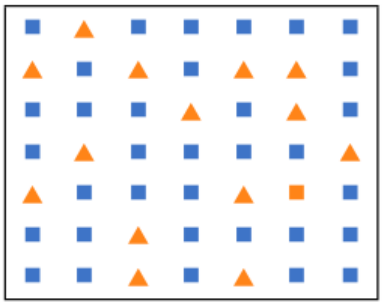


Figure 5.2: Conjunction Search [55]

triangles and squares. Colors also vary in the stimuli between orange and blue to ensure more distraction.

5.4 Bottom-up vs. Top-down Search Mechanisms (Guiding)

Definition. Visual Saliency: The salience or saliency of an object or element in a cluttered visual scene is the perpetual distinctive characteristic or physical property that makes this element stand out from its neighbors to grab our attention [55]. However, it is worth mentioning that having this salience-quality actually comes as a result from a series of interactions among the salient-object or salient-stimulus with the surrounding-objects or surrounding-stimuli, and also with the visual-search system. In other words, being able to pop-out from its surroundings because it is different in some case, but it will not pop-up in a scene with surrounding stimuli similar to it. Having a salience-property in one scenario does not guarantee that it will always stay like that in other scenarios with other surrounding stimuli and other visual-search systems (color-blinded person attending a colored scene).

There are two types of cues that direct our attention [72]: Exogenous and Endogenous with the former referring to the cues that are external to any goals we might have i.e. they automatically attract our attention (bright colors, loud noise), while the latter refers to the cues that are more internalized and involve internal knowledge to understand the cues in the first place, and the intention to follow it. Salient visual cues (also called the pop-out effect) fall under the first type; for instance: a yellow circle amid a group of red ones. Exogenous and endogenous cues mark up two main visual search techniques:

- **Bottom-Up Approach:** is also referred to as exogenous attention or stimulus-driven attention. In this type of visual search we attend to a stimulus because of it salient visual or audible properties, and usually there is no previous intent to search for it. The attention often occurs because of the pop-out effect. In this type of search we attend to the visual-salient stimulus whether we want or not [63].

- **Top-Down Approach:** is also referred to as endogenous attention or goal-driven attention. The user controls the process in this type of visual search. It is also worth mentioning that the Bottom-Up deployment of attention can be sometimes overridden by the user-driven or top-down factors [48] [57]. The invisible Gorilla is again a good example of this.

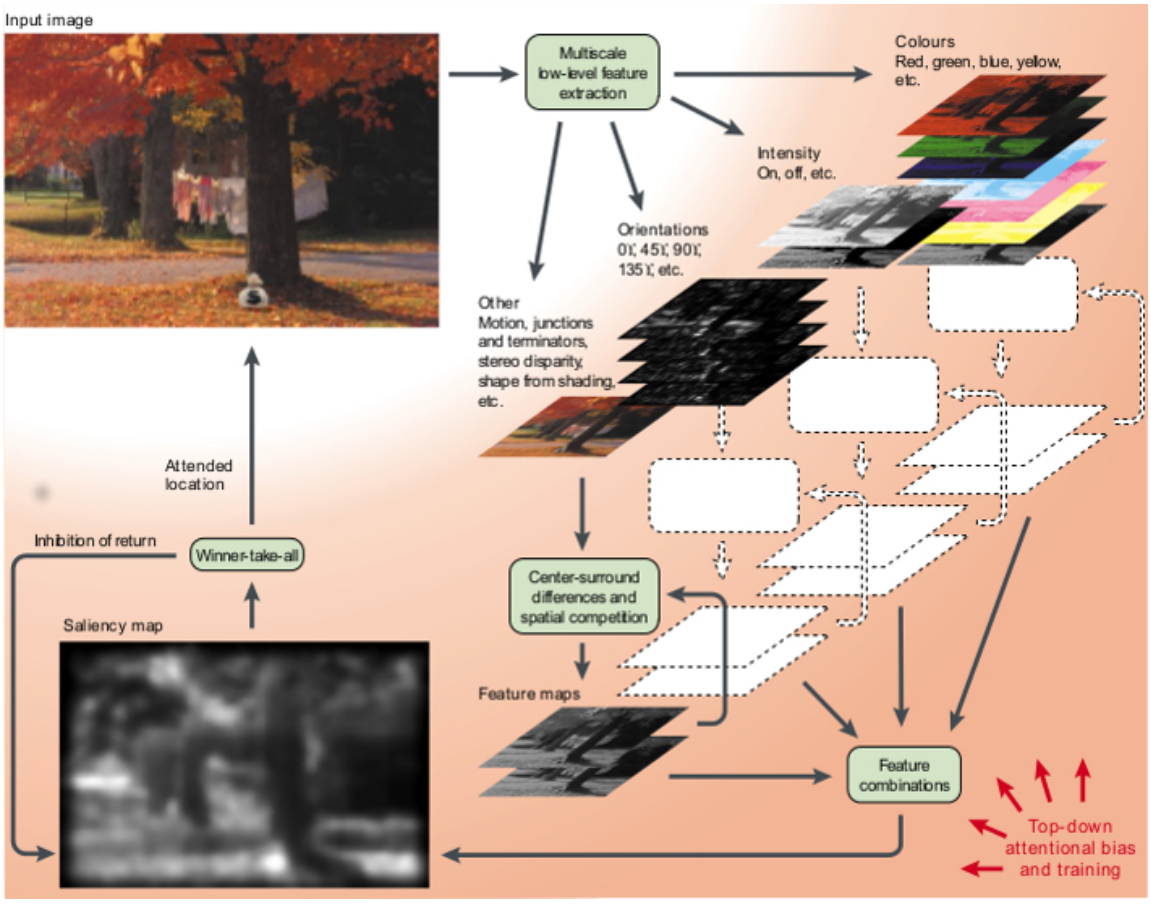


Figure 5.3: The Saliency-Maps Model

5.5 Visual Attention Status

Two types of attentional behavior can be recognized according to the involvement of the sense organs in the process, Overt and Covert attention. While in the first, the body, head, eyes, ears, etc... will be oriented towards the stimuli in the visual field trying to grasp the main features of the objects. This usually includes several eye movements or fixations on different parts of the visual field. In the second, no such body orientation will occur. Covert attention is some sort of mental or neural process that involves focusing on a single stimulus in the visual field [70].

5.6 The Saliency Maps Model

The Saliency Map is a topographically arranged map that represents visual saliency of a corresponding visual scene. Using the bottom-up approach i.e. scene properties it defines the most salient locations in the visual field. The Saliency map can effectively help to reduce the computation by performing a selection process based on the pre-attentive computed simple features. Different features will be computed on different scales then integrated into a space-related map. The Saliency-Maps model is briefly depicted in figure 5.6 [58][56].

Computing the Saliency-Maps involves the following:

- The Input Image or Retina Image: usually digitized at resolution 640*480.
- Low-Pass Filtering: Create Multi-Resolution Pyramid (Gaussian Dyadic Pyramid). Nine Spatial Scales in the pyramid: Subsampling (horizontal and vertical reduction) From 1:1 (scale 0) to 1:256 (scale 8).
- Computation of different channels feature filter for Colors, Intensity and Orientations across the nine scales.
- Combination of the feature maps into a Conspicuity-Map for each channel.
- Combination of the Conspicuity-Maps into a saliency map.
- Serial selection of salient locations using a Winner-Take-All neural network.

5.7 Visual Processing

As figure 5.7 shows, in order to finally produce the Saliency-Map of the Retina or Input image, 12 Color-Maps, 6 Intensity-Maps and 24 Orientation-Maps (computing for 4 degrees: 0,45,90,135).

Using the color components r, g, b of the pixels in the input image:

- Compute the Intensity map : $I = \frac{r+g+b}{3}$
- For each pixel in the pyramid, generate the color channels:
  - Red: $R = r - (g + b)/2$
  - Green: $G = g - (r + b)/2$
  - Blue: $B = b - (r + g)/2$
  - Yellow: $Y = (r + g)/2 - |r - g|/2 - b$
  - Negative values are set to zero.
- Determine color opponency $RG = R - G$ and $BY = B - Y$
- Orientation O will be computed at each point using Gabor-Filter.

5.7.1 Center-Surround Differences

Compute center-surround differences in order to determine the contrast or the Center-surround antagonism. This computation conforms with the understanding of how the brain processes visual information.

The Retinal Ganglion Cells (RGC) carry the visual information to the brain. RGCs have a receptive field that can be described as two concentric circles: A small circular **Center** and a **broader ring** around the center called the "**Surround**". These receptive fields fall into two categories: **Off-Center/On-Surround** and **On-Center/Off-Surround.** Different light stimuli cause different "**firing**" reactions in the cells. When the light stimulus coincides with the on-center, the firing rate is at maximum, while the surround plays an inhibitory role.

This architecture is sensitive to local spatial discontinuities. It is particularly well-suited to detecting locations which stand out from their surround i.e. salient locations.

Example: The first set of feature maps is concerned with intensity contrast, which, in mammals, is detected by neurons sensitive either to dark centers on bright surrounds or to bright centers on dark surrounds. Center-surround is implemented in the model as the difference between fine and coarse scales: The center is a pixel at scale $c\epsilon 2, 3, 4$, and the surround is the

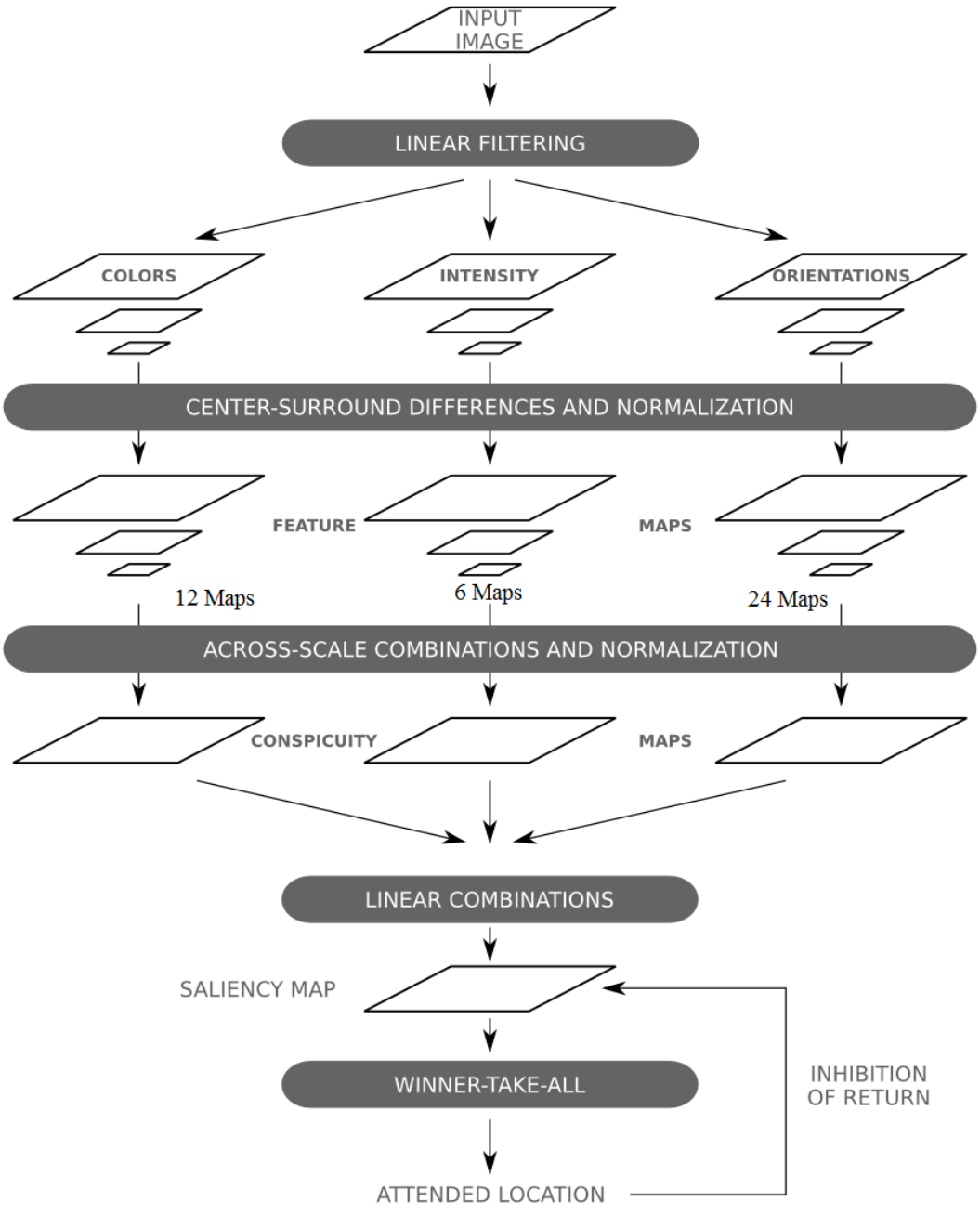


Figure 5.4: Computation of the Saliency-Map

corresponding pixel at scale $s = c + d$, with $d\epsilon 3, 4$. The across-scale difference between two maps is obtained by interpolation to the finer scale and point-by-point subtraction, as shown in the equations below:

$$RG(c, s) = |(R(c) - G(c)) \ominus (G(s) - R(s))|$$

$$BY(c, s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|$$

$$I(c, s) = |I(c) \ominus I(s)|$$

$$O(c, s, \theta) = |O(c, \theta) \ominus O(s, \theta)|$$

5.7.2 Normalization:

Saliency Map is the combination of multiple feature maps. One difficulty in combining different feature maps is that they represent a priori not comparable modalities, with different dynamic ranges and extraction mechanisms. Because all feature maps are combined, salient objects appearing strongly in only a few maps may be masked by noise or by less-salient objects present in a larger number of maps. Therefor the model promotes maps in which a small number of strong peaks of activity (conspicuous locations) is present while suppresses globally maps which contain numerous comparable peak responses.

- Normalize the valus in the map to a fied rang $[0..M]$, in order to eliminate modality-dependent amplitude differences.
- Find the locations of the map's global maximum M and compute the average $\bar{m}$of all its other local maxima.
- Globally multiply the map by $(M - \bar{m})^2$

5.7.3 Conspicuity Maps

The feature maps are combined into three Conspicuity-Maps. One for Intensity, one for color and one for orientation. They are obtained through across-scale addition, which consists of reduction of each map to scale four and point-by-point addition, as shown in the equations below.

$$\bar{I} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{c=4} N(I(c, s))$$

$$\bar{C} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{c+4} [N(RG(c, s)) + N(BY(c, s))]$$

$$\bar{O} = \sum_{\theta \epsilon \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}} N\left(\oplus_{c=2}^{4} \oplus_{s=c+3}^{c+4} N(O(c, s, \theta))\right)$$

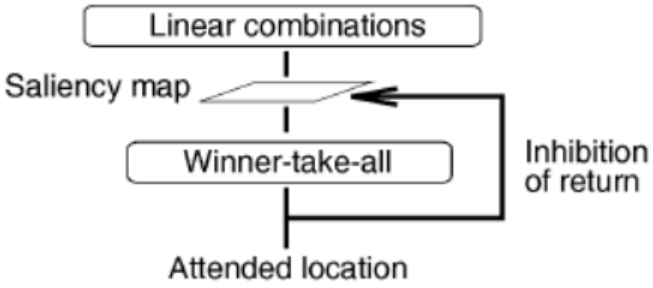


Figure 5.5: Serial Selection of the Salient Location

5.7.4 Saliency Map

The three conspicuity maps are normalized (individually) and summed into the final input S to the saliency map.

$$S = \frac{1}{3}\left(N(\bar{I}) + N(\bar{C}) + N(\bar{O})\right)$$

The motivation for the creation of three separate conspicuity maps and their individual normalization is the hypothesis that similar features compete strongly for saliency, while different modalities contribute independently to the saliency map. At any given time, the maximum of the saliency map (SM) defines the most salient image location, to which the focus of attention (FOA) should be directed.

The Saliency map will be modeled as a 2D layer of leaky integrate-and-fire neurons. A 2D "winner-take-all" (WTA) neural network will be employed in order to detect the most salient location and suppress all others (move the focus of attention) as figure 5.7.4 shows.

5.8 Example

As figure 5.8 shows: The most salient location is the orange telephone box, which appeared very strongly in C ; it becomes the first attended location (92 ms simulated time). After the inhibition-of-return feedback inhibits this location in the saliency map, the next most salient locations are successively selected.

Like the Multimedia Information Retrieval methods (MMIR), the Saliency-Maps model deals with features, features that are similar somehow to the ones identified by the MMIR. Saliency-Maps use extracted features extensively at various resolutions on finer and coarser scales, in a combination similar to what believed to be the way the mammals percept visual scenes. Since the Saliency-Maps model was originally developed for the sake of object categorization, it would make a prefect sense to use it in the multimedia search and retrieval. The Saliency-Map is a topological representation of the most salient locations in the image, so it could come in handy when working with methods such as the Smart Keypoint Matching where important points from both the query and the target will be compared, and since the Saliency-Map holds the most important i.e. salient regions in an image, this could fasten and enhance the multimedia search not to mention providing reduced versions which is in the end the goal of reduction. The computation of Saliency-Maps can be time and resource consuming, especially on embedded systems with platforms often offer insufficient amounts of RAM and other resources. Coming next is a case study from

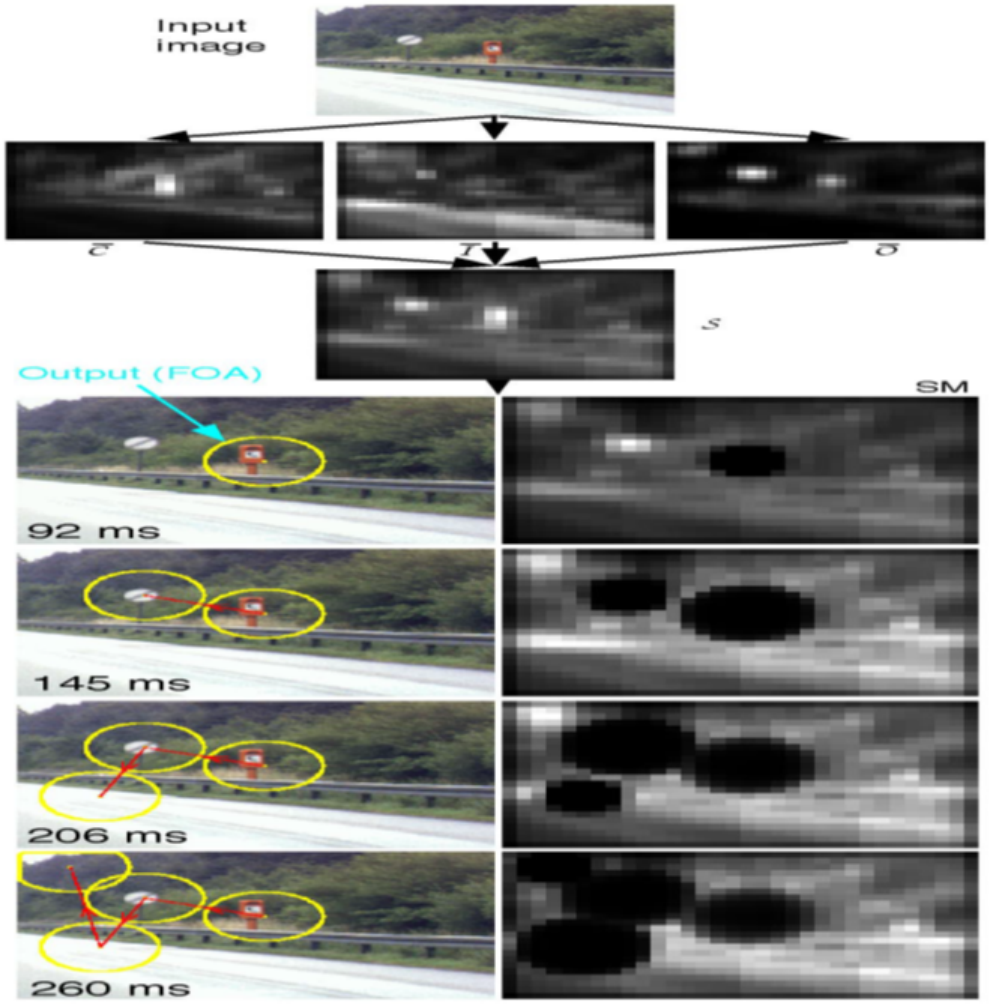


Figure 5.6: Computation of a Saliency Map

KNIME with multiple reduction techniques explained, combined and tested against real-life scenarios.

Chapter 6

Case Study: KNIME-KDD Challenge

KNIME [1] is an open source platform for data analysis, predictive analytics and modeling. KNIME has a graphical interface, it allows implementing procedures by means of workflows. A workflow is collection of nodes that represent processes. Several data mining and machine learning techniques have been implemented and they are ready to use as nodes. In this chapter KNIME will be used with the KDD challenge. Several data reduction methods will be tested with a real-life vary large database.

The KDD challenge is about Customer relationship prediction, where data about three factors will be analyzed [15]:

- Churn: Represents a contract severance by a customer.
- Appetency: The propensity to buy a service or a product.
- Upselling: The possibility of selling additional side products to the main one.

The data set came from the CRM system of a big French telecommunications company. It is a huge data set with 50K rows and 15K columns. The problem is not the size of the data set, but rather the number of input columns. The dimensionality reduction will be performed based on supervised classification algorithms. As first phase, only a small portion of data will be considered, this will give insight how to deal with the huge data set. A cascade of the most promising techniques will be used with the huge data set, these techniques are detected in the first phase of the project i.e. when working with the smaller data set.

Data columns reduction is evaluated based on the following:

- High number of missing values
- Low variance
- High correlation with other data columns
- Principal Component Analysis (PCA)
- First cuts in random forest trees
- Backward feature elimination

[1] https://www.knime.org

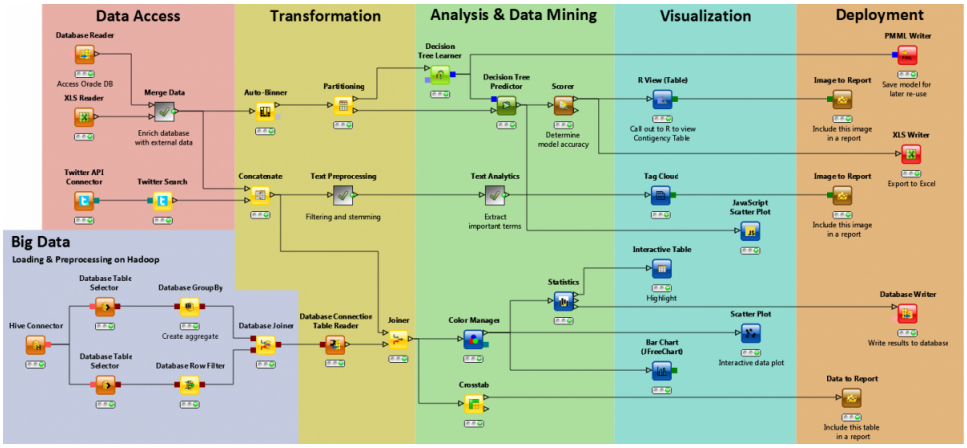


Figure 6.1: KNIME Work-Flow

- Forward feature construction

For the methods that use a threshold, an optimal threshold can be defined through an optimization loop maximizing the classification accuracy on a validation set for the best out of three classification algorithms:

- MLP
- Decision tree
- Naive Bayes

6.1 Missing Values

Principle: Removing Data Columns with too Many Missing Values. **Example:** If a data column has only $5-10\%$ of the possible values, it will likely not be useful for the classification of most records. Remove those data columns with too many missing values, i.e. with more missing values in percent than a given threshold. In order to count the number of missing values in all data columns, we can either use *SQL* select statements with *Groupby*.

Ratio of missing values = No of missing values / total number of rows

The bottleneck of this whole operation, in terms of execution time, occurs when the missing values are counted for all input data columns. This is due to the underlying data sorting.

Result: A threshold value 0.4 led to a dimensionality reduction of 71% and a corresponding accuracy of 76% by the MLP.

6.2 Low Variance Filter

Principle: Measure the column variance and remove those columns with a variance value below a given threshold. This method is limited to **numerical** columns only. There is still a need to normalize data column ranges (to$[0,1]$ for instance).

Result: Best threshold value: 0.03. Classification accuracy: 82%. Dimensionality reduction :73%. Higher threshold values actually produce worse accuracy values.

6.3 High Correlation Filter

Definition. Correlated features: Are those features that depend on one another and carry similar information.

Principle: One of the two correlated columns can be removed without decreasing the amount of information available for future tasks dramatically.

In order to determine correlation, there are two **correlation coefficients** that can be used:

- Pearson's Product Moment Coefficient ρ for numerical data.
- Pearson's chi square value for nominal or categorical data.

No correlation coefficient is defined between a numerical and a nominal data column; and like with other techniques, normalization for uniform data ranges is needed.

6.4 Principal Component Analysis PCA

This techniques has already been explained in a previous chapter, in terms of principal and computation. In this chapter however the technique will be evaluated and used against real-life data sets.

PCA transformation is sensitive to the relative scaling of the original variables. Data column ranges need to be normalized before applying PCA. The new coordinates i.e. the Principal Components PCs are not real system-produced variables anymore. Applying PCA to your data set causes it to lose its interpretability. If interpretability of the results is important for the analysis, PCA is not recommended.

PCA only affects the numerical columns and does not act on the nominal ones, not to mention that it also skips the missing values. When used against a data set with many missing values, PCA will be less effective.

Result: In the first phase (PCA was used only on a small portion of the original data set)

- With loss of 4% of the original data set information, PCA Reduced the number of columns from 231 to 87.
- 62% reduction rate and an accuracy of 74%.
- Threshold at 96% to keep accuracy over 70%.

6.5 Backward Feature Elimination

Principal: The Backward Feature Elimination loop performs dimensionality reduction against a particular machine learning algorithm. At each iteration, the selected classification algorithm is trained on n input features. Then we remove one input feature at a time and train the same model on $n-1$ input features n times. Finally, the input feature whose removal has produced the smallest increase in the error rate is removed, leaving us with $n-1$ input features. The classification is then repeated using $n-2$ features $n-1$ times and so on. The algorithm starts with all available N input features and continues till only 1 last feature is left for classification.

Each iteration k produces a model trained on $n-k$ features and an error ratee(k). As for selecting the maximum tolerable error rate, we define the smallest number of features necessary to reach the desired classification performance with the selected machine learning algorithm. The default classification algorithm is **Naive Bayes** (can be other classification algorithms).

The main drawback of this technique is the high number of iterations for very high dimensional data sets, possibly leading to very long computation times. **It is better to apply it only after some other dimensionality reduction had taken place.**

Result: Only 77 features took circa 10 hours on a quad-core laptop with $8GB$ RAM.

6.6 Forward Feature Construction

Principal: Builds a number of pre-selected classifiers using an incremental number of input features. A loop starts from1 feature and adds one more feature at a time in the subsequent iterations.

Drawback: Long computational time for high dimensional data sets.

Result: Using the Naive Bayes as a classification algorithm and 20 final input features (91% reduction rate): 83% accuracy on the validation set.

6.7 Comparison

Working with the small data set gave the following results:

- The machine learning algorithm with the highest accuracy on the validation set is the neural network MLP.
- The missing value ratio seems a **safe starting point**. It obtains a good reduction rate without compromising performances and applies to both numerical and nominal columns.
- The random forest based method achieves the highest reduction rate.
- Backward elimination and forward construction techniques produce great figures in terms of accuracy. However, their reduction rate proved to be the highest.
- PCA seems to perform the worst in the compromise between reduction rate and model performance

6.8 Combining Dimensionality Reduction Techniques

In order to work with the huge data set a combination of the best-evaluated techniques was used.

- **First:** Missing Values with threshold 90% reduced the number of data columns from15000 to14578 (3% reduction).
- **Second:** Low Variance Filter on a **subset** of the data rows with a 0.03 threshold value further reduced the number of input columns from14578 to 3511 (76% reduction).
- **Finally:** PCA with allowing a loss of 1% of the information, the data set dimensionality could be reduced from 3511 to 2136 data columns (39% reduction).

Total dimensionality reduction : 86% (15000 original data columns to 2136 reduced ones).

Summary and Discussion

In the past few years, with the rise of new technologies and the extensive use of social media, the world witnessed a data flood on biblical levels. This data flood was a great opportunity for the enterprises to gain more insights and delve into deeper business analysis. However this came with new challenges. The computer systems had to evolve on both hardware and software levels. Keeping up with the new data age doesn't come cheap. Even if the computer systems were powerful enough to handle massive amounts of data, unrelated or bad data can drive the analysis in a completely wrong direction. Before going in for deeper analysis, data sets need to be reduced in terms of attributes and records. This reduction is no luxury, it is a must. It can significantly ease the process of data management and enhance the overall performance, not to mention stripping away many elements that may affect the validity of the analysis.

Although it is not typically suited to handle this type of tasks, the traditional relational model can still offer some help using native techniques originally developed to perform general-purpose queries. When working with these techniques, reduction may affect the data itself, attributes or both. While selection usually involves some condition and may be used alone with no need for extra coefficients as long as the condition is well formed, projection on the other hand can't be simply used blindly, just for the sake of reduction. It has to be accompanied with some metrics to asses the data loss. For instance, the data engineer can experiment with the columns and asset the information loss with each combination using Kullback-Leibler divergence. User-defined or generated thresholds can be always used to set the necessary borders, either with the projection or any other technique. MLP neural networks offer a good option in this regard.

Mathematical transformations introducing data in new perspectives such as Principal Component Analysis can detect where most of the data is concentrated. They can - accompanied with the right threshold - eliminate the columns or attributes that have little variance. As mentioned earlier, PCA will not be the best option in the case of too many missing values or if the interpretability of the data is essential.

All the techniques mentioned so far in this summary would work perfectly fine either on numerical data, textual data or both, however they can't affect the multimedia data. Multimedia data reached new peaks in this data surge, and given the fact that the size of a single multimedia document could exceed the size of hundreds of records containing textual and numerical data combined, or even more; it makes a perfect sense to include this type of data in the quest for efficient reduction techniques. Tons of images and videos are produced and uploaded every day which could easily cause a shortage in data storage. While the reduction in the case of numerical and textual data will be for the sake of better insights, visualization and performance, the reduction in the case of multimedia data would be rather more for the sake of capacity and content-based retrieval. Saliency maps and similar bio-inspired models can help in this matter by extracting valid representations of an image based on its most salient locations. You can consider the saliency map somehow to be equivalent to the principal components when working with collections of numerical data.

The techniques discussed in this thesis are just a few of the methods available to perform data and dimensionality reduction. The thesis didn't not particularly investigate the rich world

of machine learning. Techniques such as Random Projection proved to be effective with fewer errors compared to other techniques. Statistical methods similar to principal component analysis such as Exploratory Factor Analysis can be also used, or even extensions to the already-discussed PCA, such as Kernel PCA.

Which technique (or which combination of techniques) to use highly depends on the data set itself. There is no ready-to-use recipe that would work the same for all data sets. When using a combination of techniques, the order plays a significant role, especially in terms of performance since some methods tend to be time-consuming, so using less time-consuming techniques before would alleviate the path better for such techniques. The data engineer should experiment with a portion of the data, study patterns and assess on a small scale before moving to the larger data set.

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *ESANN*, 2013.

[2] Danilo Bargen. Programming a perceptron in python. Danilo Bargen Blog, March 2013. Online; accessed 21-September-2016;.

[3] Jason Brownlee. Crash course on multi-layer perceptron neural networks. Machine Learning Mastery Website, May 2016. Online; accessed 21-September-2016;.

[4] "Kenan Systems Corporation". Introduction to multidimensional database technology, 1995.

[5] George Dallas. Principal component analysis 4 dummies: Eigenvectors, eigenvalues and dimension reduction. George Dallas Blog, October 2013. Online; accessed 28-June-2016;.

[6] H. Garcia-Molina, J. Ullman, and J. Widom. *Database System Implementation*. Prentice Hall, 1999.

[7] Logan Harbaugh. Big data ssd architecture: Digging deep to discover where ssd performance pays off. Samsung Business Insights, January 2016. Online; accessed 12-September-2016;.

[8] Stacey Higginbotham. Sensor networks top social networks for big data. Gigaom Official Website, September 2010. Online; accessed 28-June-2016;.

[9] Yu Hen Hu. *Handbook of Neural Network Signal Processing*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2000.

[10] IBM. What is big data. IBM Official Website, April 2014. Online; accessed 27-June-2016; Page Date retrieved using: http://centralops.net/co/.

[11] Programmer Interview. How do database indexes work. Programmer Interview Website. Online; accessed 21-September-2016;.

[12] Abbas Keshvani. Principal component analysis in 6 steps. Coolstatsblog, March 2015. Online; accessed 23-09-2016;.

[13] Justin Kestelyn. The truth about mapreduce performance on ssds. Cloudera Official Blog, March 2014. Online; accessed 12-September-2016;.

[14] M.A. Khamsi. Computation of eigenvectors. SOSMath Website. Online; accessed 23-09-2016;.

[15] KNIME-Team. Seven techniques for dimensionality reduction. KNIME Website, 2014.

[16] KELLY LEBOEUF. 2016 update: What happens in one internet minute?, February 2016. Online; accessed 27-June-2016.

[17] Pei Lee, Laks V. S. Lakshmanan, and Jeffrey Xu Yu. On top-k structural similarity search. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, pages 774–785, 2012.

[18] Jiuyong Li, Jixue Liu, Muzammil Baig, and Raymond Chi-Wing Wong. Information based data anonymization for classification utility. *Data Knowl. Eng.*, 70(12):1030–1045, December 2011.

[19] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*, chapter 2, page 46. Cambridge University Press, 2005. Version 7.2 (fourth printing) March 28, 2005.

[20] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute Official Website, May 2011.

[21] Technical Marketing. Ssds for big data - fast processing requires high-performance storage. Technical report, Micron Technology, Inc, March 2016.

[22] Eston Martz. Understanding qualitative, quantitative, attribute, discrete, and continuous data types. MiniTab Blog, December 2014. Online; accessed 21-September-2016;.

[23] Jen Methvin. Does more data = more power? Import.io Website, September 2013. Online; accessed 19-September-2016;.

[24] k. Nwosu, B. Thuraisingham, and P. Berra, editors. *Multimedia Database Systems: Design and Implementation Strategies*. Springer, 1996.

[25] Victor Powell. Principal component analysis - explained visually. Setosa Website. Online; accessed 23-09-2016;.

[26] Sebastian Raschka. Linear discriminant analysis - bit by bit. Sebastian Raschka Website, August 2014. Online; accessed 23-09-2016;.

[27] Eric Roberts. The perceptron. Stanford Computer Science Website, 2000. Online; accessed 21-September-2016;.

[28] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[29] Margaret Rouse. What is data aggregation. Techtarget Official Website, September 2005. Online; accessed 04-August-2016; Web page date obtained using: timetravel.mementoweb.org.

[30] Margaret Rouse. What is multidimensional database (mdb) - definition from whatis.com. TechTarget network Website, 2005. Online; accessed 21-September-2016;.

[31] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[32] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann, 1st edition, 2006.

[33] Scikit-Learn. Scikit-Learn Website. Online; accessed 23-09-2016;.

[34] Åse Dragland. Big data - for better or worse. SINTEF Official Website, May 2013. Online; accessed 27-June-2016.

[35] S. Stanczyk, B. Champion, and R. Leyton. *Theory and Practice of Relational Databases*. CRC Press, 2nd edition, August 2001.

[36] StreamSets. 'bad data' is polluting bidg data. StreamSets Official Website, June 2016.

[37] Techopedia. Data aggregation. Techopedia Official Website, January 2013. Online; accessed 04-August-2016;.

[38] William M.K. Trochim. Types of data. Web Center for Social Research Methods, October 2006. Online; accessed 21-September-2016;.

[39] Linda Tucci. Big data can mean bad analytics, says harvard professor. TechTarget network Website, July 2013. Online; accessed 12-September-2016;.

[40] W3School. Sql group by statement. W3School Official Website, Match 2001. Online; accessed 04-August-2016; Web page date obtained using: timetravel.mementoweb.org.

[41] W3School. Sql functions. W3School Official Website, November 2004. Online; accessed 04-August-2016; Web page date obtained using: timetravel.mementoweb.org.

[42] Ben Walker. Every day big data statistics - 2.5 quintillion bytes of data created daily, April 2015. Online; accessed 27-June-2016.

[43] Markus Winand. Concatenated indexes. Use the Index, Luke Website, Novemeber 2011. Online; accessed 23-09-2016;.

[44] Dan Worth. Cern experiments generating one petabyte of data every second. V3 Official Website, June 2011. Online; accessed 28-June-2016;.

[45] Stefanie I Becker and Gernot Horstmann. A feature-weighting account of priming in conjunction search. *Attention, Perception, & Psychophysics*, 71(2):258–272, 2009.

[46] J. Braun, C. Koch, and J. Davis. *Visual Attention and Cortical Circuits (MIT Press)*. A Bradford Book, 2001.

[47] James R. Brockmole and David E. Irwin. Eye movements and the integration of visual memory and visual perception. *Perception & Psychophysics*, 67(3):495–512, 2005.

[48] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

[49] E. Goldstein, editor. *The Blackwell Handbook of Sensation and Perception*, chapter Visual Attention, page 272. Wiley-Blackwell, 2004.

[50] James E Hoffman and Baskaran Subramaniam. The role of visual attention in saccadic eye movements. *Perception & psychophysics*, 57(6):787–795, 1995.

[51] Po-Jang Hsieh, Jaron T Colas, and Nancy Kanwisher. Pop-out without awareness unseen feature singletons capture attention only when top-down attention is available. *Psychological science*, 22(9):1220–1226, 2011.

[52] H. Intraub and L. Nadel. *Encyclopedia of cognitive Science*, chapter Visual Scene Perception, pages 524–527. London: Nature Publishing Group, 2002.

[53] Information Resources Management Association (IRMA), editor. *Image Processing: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*, chapter 22, page 424. IGI Global, 2013.

[54] David E. Irwin and Gregory J. Zelinsky. Eye movements and scene perception: Memory for things observed. *Perception & Psychophysics*, 64(6):882–895, 2002.

[55] L. Itti. Visual salience. 2(9):3327, 2007. revision #72776.

[56] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12):1489–1506, 2000.

[57] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

[58] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, November 1998.

[59] Jean-Christophe Nebel, MichałLewandowski, Jérôme Thévenon, Francisco Martínez, and Sergio Velastin. Are current monocular computer vision systems for human action recognition suitable for visual surveillance applications? In *Proceedings of the 7th International Conference on Advances in Visual Computing - Volume Part II*, ISVC'11, pages 290–299, Berlin, Heidelberg, 2011. Springer-Verlag.

[60] M. Passer and R. Smith. *Study Guide for use with Psychology: The Science of Mind and Behavior*, chapter Sensation and Perception, pages 78–79. McGraw-Hill, 2004.

[61] Boris M Sheliga, Lucia Riggio, and Giacomo Rizzolatti. Orienting of attention and eye movements. *Experimental Brain Research*, 98(3):507–522, 1994.

[62] Daniel Simons. Selective attention test. YouTube, Match 2010. Online; accessed 23-12-2013;.

[63] Jan Theeuwes. Exogenous and endogenous control of attention: The effect of visual onsets and offsets. *Perception & psychophysics*, 49(1):83–90, 1991.

[64] ThePeakPerformanceCenter. Types of attention. Web. Online; accessed 23-09-2016;.

[65] Massimo Turatto, Matteo Valsecchi, Adriane E Seiffert, and Alfonso Caramazza. On the speed of pop-out in feature search. *Journal of experimental psychology: human perception and performance*, 36(5):1145, 2010.

[66] A. Vu, A. Ramanandan, A. Chen, J. A. Farrell, and M. Barth. Real-time computer vision/dgps-aided inertial navigation system for lane-level vehicle navigation. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):899–913, June 2012.

[67] J. Wolfe and T. S. Horowitz. Visual search. 3(7):3325, 2008. revision #145401.

[68] Jeremy M Wolfe. Visual attention.

[69] Jeremy M. Wolfe. What can 1 million trials tell us about visual search?, 1998.

[70] R. Wright and L. Ward. *Orienting of Attention*, chapter Introduction, pages 11–13. Oxford University Press, 2008.

[71] A. Yarbus. *Eye Movements and Vision*, chapter VII, pages 171–196. Plenum Press, 1967.

[72] Carole Yue. Divided attention, selective attention, inattentional blindness, & change blindness. KhanAcademyMedicine on YouTube, September 2013. Online; accessed 04-01-2014;.