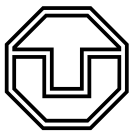


TECHNISCHE UNIVERSITÄT DRESDEN

Dep. of Computer Science Institute for System Architecture, Database Technology Group

Diploma Thesis

CUSTOMER AND PRODUCT MODELING WITH RECEIPT DATA

Lucas Woltmann

Matr.-Nr.: 3758218

Supervised by:

Prof. Dr.-Ing. Wolfgang Lehner

and:

Dr.-Ing. Maik Thiele

Submitted on 18th August 2017

CONFIRMATION

I confirm that I independently prepared the thesis and that I used only the references and auxiliary means indicated in the thesis.

Dresden, 18th August 2017

ABSTRACT

Customer and product modeling has become a major aspect in market research over the last years. With global players trying to optimize their distribution network and profit, there is a high interest in knowing the customer's behavior and predict the customer's course of action. This thesis is modeled around this demand. It provides new strategies for customer and product modeling based on receipt data. It uses several techniques from other domains, like Natural Language Processing (NLP) and Biological Interaction Analysis (BIA). These approaches are employed to a data set containing electronic receipts from different merchants for a large amount of customers. We will show that it is possible to annotate customers and products with information which are not in the original data set. This includes categorizing customers and products into segments and predicting purchase decisions. All this requires a comparable construct for words, customers and products. We embed products into a word vector space given their product titles and we embed customers according to their purchased products into one combined vector space. We will also try to improve our results by modeling customers and products as particles. This leads the way for simulating customers and products as a physical interaction system. We are going to establish the kernel-like structure of BIA and simulation and optimize its performance for better runtimes. We will show the advantage of using the products as a context for the customers. This advantage will be set into the context of further research.

ACKNOWLEDGMENTS

Of course there is a life after death. But then, the real questions are:
How far is it from the city center? And: What are the opening hours?

- Woody Allen (Disputed)

Additionally to the two examining supervisors, this thesis was conjointly supervised by

Dr. Thomas Kirsche,
GfK SE, Nuremberg, Germany

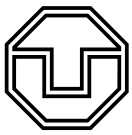
Dr. Susanne Heckmann,
GfK SE, Nuremberg, Germany

Michael Schlueter.
GfK SE, Berlin, Germany

Prof. Dr. sc. techn. Ivo F. Sbalzarini,
Technische Universität Dresden and Center for Systems Biology Dresden, Germany

Johannes Bamme,
Technische Universität Dresden and Center for Systems Biology Dresden, Germany

I would like to thank all supervisors and persons involved for making this thesis an exciting research topic. The thesis is a work across different areas of expertise. From the market research context of the GfK, to Natural Language Processing done by the Database Technology Group at TU Dresden, to the biological concepts employed by the Center for Systems Biology Dresden. Through that, this work can be seen as a synergistic one. The received feedback of the different expert groups was always profound and productive. It lead me to the things you can read on the following pages.



TECHNISCHE
UNIVERSITÄT
DRESDEN

Dep. of Computer Science Institute for System Architecture, Database Technology Group

EXPOSÉ DIPLOMA THESIS
USER BEHAVIOR MODELING WITH CUSTOMER RECEIPTS

Lucas Woltmann

Supervised by:
Prof. Dr.-Ing. Wolfgang Lehner

and:
Dr. Ing. Maik Thiele

Submitted on 10^{th} March 2017

GENERAL INFORMATION

Supervision:

Prof. Dr.-Ing. Wolfgang Lehner,
Technische Universität Dresden, Germany

Dr.-Ing. Maik Thiele,
Technische Universität Dresden, Germany

Prof. Dr. sc. techn. Ivo F. Sbalzarini,
Technische Universität Dresden, Germany

Johannes Bamme,
Technische Universität Dresden, Germany

Dr. Thomas Kirsche,
GfK, Nürnberg, Germany

Dr. Susanne Heckmann,
GfK, Nürnberg, Germany

Michael Schlueter,
GfK, Berlin, Germany

Duration: 10.03.2017 – 18.08.2017

PROBLEM STATEMENT

The main part of the thesis will give an example on how to use known methods in data analysis and simulation on an unknown example data set. The context of the data and thesis is customer/user behavior modeling. The focus of the analysis will be the understanding, modeling and prediction of user purchases and cash flow.

The thesis will elaborate hypothesizes around the given context. Hypothesizes will be established for current questions in market and consumer research. Additionally, the hypothesizes can be updated throughout the thesis for better specification of problems and their understanding.

RELEVANCE AND MOTIVATION

User Behavior Modeling (UBM) is a very important part in data analysis. Global players on the market are always interested in how to place and sell their products in a variety of options. Expenditure and shopping cart analysis are just two examples in the variety of methods. This variety makes it hard to find the right (i.e. a high quality) model. It can take several approaches with different ideas to get to a point of satisfactory insights. The question always remains the same: What impels the market? In our case the problem can also be described with the question: What will the customer do/purchase next? The data we are using are e-mail recipes and order confirmations detailing the consumerism of people over a period of time. The data is not restrained to one supplier, but gives information from the point of view of the customers. Therefore the data models a more complete part of the market, compared to the data collected by just one supplier.

A general problem is the variety of ideas on how to explore the data. There are many report and modeling techniques which could be applied. To give the thesis a more specific focus, we will

concentrate on two groups of methods. The first is segmentation. Customers and items can be put in different groups, according to their purchase characteristics. customers can be defined by age, gender or location. Items will be put into their general market segments. A segmented customer base opens the way for specialized advertisement and product recommendations. The second approach is behavior predicting. With a unconstrained time series of receipts, one can simulate a sequence of purchases as the customer's behavior on the market. It would be favorable for any supplier to predict the customers purchase decision. Especially logistical planning for storing and shipping profits if the next purchase of the customer is already known within a limited amount of choices. It reduces the overhead in organizing a large scale distribution network. Distribution is not the only driving factor. Suppliers are interested in consumer choices to place products, through ads and marketing, most effectively. So, consumer decisions can be put into perspective for the whole market, giving a certain insight into the market's behavior at large.

BACKGROUND

UBM can be comprehended in different ways. The methods for modeling are sheer endless. To get a preliminary feeling for the data, it is imperative to use reports. Simple report measures with respective visualization can give an overview over the properties of the data. They can be used to find hypotheses to be examined with different techniques. Following the overview, the two more advanced techniques can be applied. The first one should give insights into the segmentation of item and customers. A suitable method could be clustering, like DBSCAN [EKS+96]. Another possibility would be to use deep learning for classification [GBB11] [Cho16]. Other publications have shown that deep learning can bring insights to topics with easy model structures [BGJM16]. It is to be noted that for a reasonable good segmentation, we need to do some feature engineering [TFLW99]. In this case, it would be recommended to use Markov Random Fields [Roz82] for behavior prediction. The bases are the purchase decisions, which can be modeled as a graph. The graph contains the events and decisions as nodes and a network of probabilities between those as edges. This will give the opportunity to predict user behavior and spendings. Based on the data, there are a lot more different approaches possible, like expanding the models to make predictions about the whole market. The Particle Method [HE88] can simulate interaction between individual nodes (particles) in a network. Given the fact that the market can be seen as a network of customers and merchants, it should be possible to define a model giving the possibilities to simulate the market and predict purchase decisions.

APPROACH

In this thesis we want to give an example on how to work with unknown data in UBM. Since we cannot say what the significance of the information is yet, we will use exactly this point of reference to build around. The challenge is not to make it look like we are just "*poking it with a stick*", but to use a forward-oriented structured approach. The structure is proposed as follows:

1. Descriptive work:
   Generate reports describing static information of the data This is necessary to get a general understanding for the data set. Possible questions to be answered:

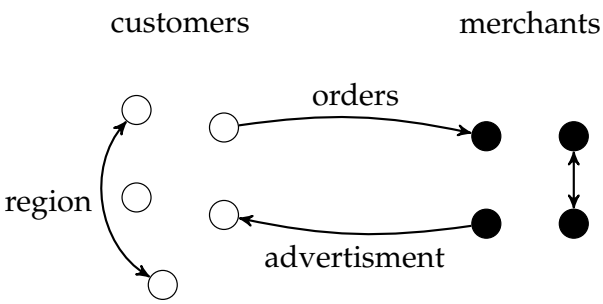


Figure 1: Model

- What is the general distribution of item categories?
- How is the cash flow between the item categories?
- What does the average customer spend where and when?

2. Advanced descriptive work:
   Generate insights using methods giving information which cannot be inferred directly. Possible questions to be answered:
   - What kind of customers/expenditures are there? (clustering)
     - What item categories are there in general (e.g. electronics, healthcare, etc.)?
     - Is there a constrain between time and purchased categories?
     - Is there a constrain between customer and purchased categories?
   - What kind of target group does the customer belong to? (classification)
     - What age is the customer?
     - What is the gender of a customer?
     - Where does the customer live?

3. Predictive work:
   Generate insights using methods giving information about the future or unknown customers. Possible questions to be answered:
   - How does a customer behave? (Markov Random Fields)
     - Can we predict the next purchase?
     - Can we predict the time of the next purchase?
   - What does/will impel the market? (Particle Method)
     - Can the market be simulated?
     - Can we simulate unknown states of the market?

The third point is the main focus. The proposed model for simulating the market with the Particle Method is given in Figure 1. There are two types of nodes (particles): customers and merchants. The particles are color-coded. The customers interact with the merchants via orders and between each other by their regional composition, like a shared postal code. The merchants can satisfy customers' need by delivering a product or create needs by placing ads. The properties and actions of each particle type and interaction are listed in Figure 2.

The first and second point are equally important, but should be seen as a "hygienic measure" and therefore a must-do. It is necessary to get the information from the first two phases before

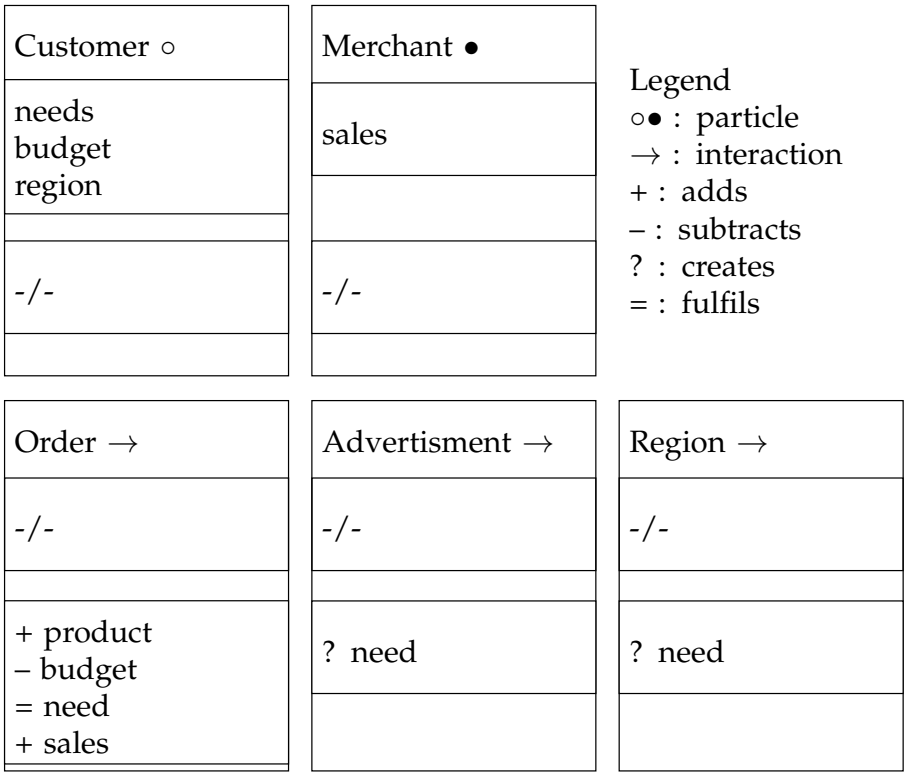


Figure 2: Action cards

building more complex models. It helps to find a structured way of modeling the data by giving an overview and first insights. The reports can give bases for how the merchants satisfy the customers' needs and how the customer interacts with the merchant thereafter.

EVALUATION

The results of the data analysis part are easy to evaluate. There are certain measurements accessing the quality of a model. In machine learning these are *accuracy*, *precision* and *recall*, just to name a few. If applied, those measures can give a good quality assessment. In the course of writing the thesis, definitions will be established about what good quality is and if a model outmatches others. The used measurement is defined per subtask (i.e. accuracy of the age classification). All findings should be compared to a baseline model. The thesis should contain the reports of the descriptive work including a textual description of the insight and how it was inferred from the table or diagram. Additionally, the results of the higher order algorithms in part two and three are explained and put into context.

CONTENTS

1 INTRODUCTION

Modern market research faces a lot of challenges. A more globalized view on every aspect of consumerism brings new ways of understanding markets and their impact on every day life. Alongside with a global market, there is a vast growth in data collected in connection with consumerism. One can assume an ubiquity of data in online and retail shopping. The large amount of data must be processed in split seconds to provide valuable insights into the markets. A profound model of the market can give a merchant the edge over the competition. The core questions for successful marketing are: Where should I launch my product? When should I launch it? Who will buy my product? These are the current challenges for data-driven market research [HWH+16]. Single producers and merchants only have a limited insight into the market. This is why companies in market research, like GfK, put a lot of effort into collecting and analyzing data across a broad spectrum of sources, including experienced market experts. This builds the center for reliable market forecasts. A modern market analysis can answer all core questions giving a producer the confidence of placing his product on the right markets at the right time. Still, there is a lot of research done in this area. A lot of insight is handcrafted. Market experts use their long lasting experience to help modeling customers and markets. A general task is the reproduction of such knowledge in an automated way. A recurring topic over the last years is Customer Modeling (CM) [HCDK17]. *Customer Modeling* is the superordinate term for trying to understand the customer as an individual. A customer can be characterized into target groups exposing collective patterns in purchase behavior. The more similar customers are according to their target group the more similar is their purchase behavior. On the other hand, behavior modeling is changing dramatically right now. It moves away from group-oriented analysis towards single individual behavior. Nevertheless, the information about a customer based on a group are still highly relevant. The individual shopping behavior should be seen as an additional tool to model customers. In individual shopping behavior analysis, market experts speak of so called „shopping missions“ [HCDK17]. *Shopping missions* model the intention behind every choice a customer makes. This can be the decision to buy a product or the choice of store to go to. The current entry point for this research are receipts. Receipts can model an individual very well because they have a personal relation to a single customer. Every shopper has different shopping patterns. It is most desirable to find these patterns and make them accessible to analytical systems. The aggregation over all customer patterns in a market can model the market and therefore answer our three initial questions.

In this thesis, we want to focus on the customer modeled by language. A major part in market research is the search for possibilities to represent a user in analytical ways. The current advances show that it is not possible to completely reconstruct a human in a machine accessible form, yet. Simplified models are needed. Most of the time, algorithms only depict a part of human behavior. A conglomerate of these models, tools and algorithms builds the base for CM [HWH+16]. CM is a very important part of market research. CM can be comprehended in different ways. The most basic definition for a user based behavior model comes from the context of Human-Machine-Interaction. It is described as the conceptual understanding of an user [Fis01] within a computational environment. This includes all the necessary techniques and tools. The methods for modeling behavior are sheer endless. Nevertheless, global players on the market are always interested in how to place and sell their products in a broad spectrum of options. The variety of methods makes it hard to find the right (i.e. a high quality) model. It takes several approaches with different ideas to get to a point of satisfactory insights. The core questions, as stated in the previous paragraph, always remain the same. In our case the problem can also be described with the questions: What kind of customers are there? What will the customer do/purchase next? We want to answer those question by employing methods of Natural Language Processing and Bioinformatics. These methods are not commonly used in this context and will yield new approaches for market research.

NEW METHODS IN MARKET RESEARCH

The general problem is the variety of ideas on how to explore data. There are many analysis and modeling techniques which can be applied. To give the thesis a more specific focus, we will concentrate on three groups of methods. They are called segmentation, recommendations and purchase prediction. The selection of methods was chosen according to the demands described in the previous section. *Segmentation* can be divided into customer and product segmentation. Customers and items can be put in different groups in respect to their purchase characteristics. Customers can be defined by marital status, gender or location. Items can be put into their general market segments. A segmented customer base opens the way for specialized advertisement, product recommendations and shopping cart analysis. *Product recommendations* are also an important part in market research. They are used to display purchase options to the customer which he could like. The purchase options are personalized for the customer. Good recommender systems will guide the customer through his shopping experience. Recommendations therefore help customer to come to a purchase decision. On the other hand, well placed recommendations can be used to efficiently model supply and demand for a merchant. If the merchant already knows what the customer is most likely to buy, it is easier to maximize profits by intelligent marketing. Recommendations can help to give information about the customers future choices by actively induce the desire to purchase particular products. The last approach we want to introduce is *purchase prediction*. With an unconstrained time series of receipts, one can simulate a sequence of purchases as the customer's behavior on the market. It would be favorable for any supplier to predict the customers purchase decision. Especially logistical planning for storing and shipping profits if the next purchase of the customer is already known within a limited amount of choices. It reduces the overhead in organizing a large scale distribution network. Distribution is not the only driving factor. Merchants are interested in consumer's choices to place products, through ads and marketing, most effectively. Effective marketing will lead to maximized profits. If the decisions of all customers in the market could be combined, one get a global prediction pattern

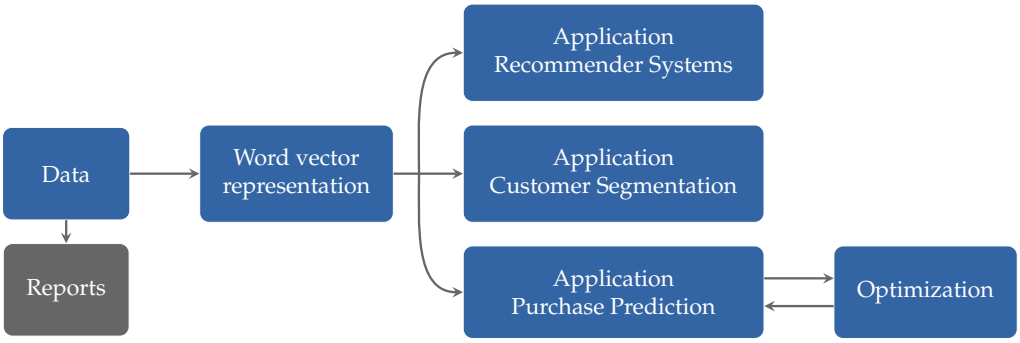


Figure 1.1: Structure of research

for the market. So, consumer decisions can be put into perspective for the whole market, giving a certain insight into the market's behavior at large.

THESIS STRUCTURE

This thesis is divided into three major parts. These are: Data Description, Natural Language Processing (NLP) and Biological Interaction Analysis (BIA). The chapters depend on each other in the given order. They visualize the logical train of thought of this research. Figure 1.1 gives an overview about our progress and the structure of the thesis.

In Chapter 2 one can find all preliminary explanations for the used models and algorithms in the following chapters. The first thing, we have decided, was to use reports to get a preliminary feeling for the data. Simple report measures with respective visualization will give an overview over the properties of the data in Chapter 3. They can be used to find hypotheses to be examined with different techniques. Properties being important in a market research context are called *Key Performance Indicators* (KPIs) [FG90]. They are predefined calculations giving standardized comparable information about the data. KPIs include, but are not limited to, the sales per customer, the sales per product group or the average number of items purchased by a customer. We will employ such methods to our data and build the connection between them. Following the overview, we will use Natural Language Processing (NLP) on the data. The chosen subcategory of NLP concepts is called word vector representation. Its main feature is the modeling of language in an algebraic vector space. We used product titles as the starting point of our analysis. The preliminary assumption is that the language of product titles is a subset of the English language and that the word vector representation can maintain the structural significance of words in a language. Using the vectors of the product titles' words, it is possible to build one unified vector space for words, products and customers. We will introduce three applications for this combined vector space in Chapter 4. These are recommendations, customer and product segmentation and purchase predictions. As the final step, we will apply methods from Bioinformatics to the combined vector space to optimize the results of the purchase prediction in Chapter 5. For this, we have chosen BIA because of its close connections to *Particle Mesh Methods* (PMM). BIA is a subcategory [HPS10] of PMMs. PMM [HE88] can simulate interaction between individual particles in a space. Given the fact that the market can be seen as a space of customers, merchants and products, it is possible to define an interaction model. This leaves us with the possibility to alter the unified vector space via simulation and improve the results for the purchase prediction.

2 RELATED WORK

In this chapter we are going to lay the foundations for the needed understanding of models and algorithms in use. We will mainly concentrate on two things. The first one is word vector representation as a subcategory of Natural Language Processing (NLP). To be more specific, we employed a technique called word2vec and its successor fasttext. Section 2.1 will show the workings of both concepts. Additionally, we will highlight some specialties the algorithms have which are important for further analysis. The second point we want to introduce is Biological Interaction Analysis (BIA). Section 2.2 will focus on BIA and its embedding in the research of Particle Mesh Methods (PMM) and Simulation. The concepts in this part of the thesis will be expanded in Chapter 4 for the word vector representation and in Chapter 5 for BIA.

2.1 WORD VECTOR REPRESENTATION

Natural Language Processing (NLP) is not necessarily the first approach you will find in market research. But in recent years, NLP has gained quite a lot of attention. NLP encapsulates all research done for understanding and modeling natural languages within computational environments. The subcategory we are taking into account in this thesis is *word vector representation*. Starting with tf-idf [SB88] for text retrieval in search engines, the idea behind vector representations for words is simple. Every word or document in a given language is not depicted as a set of letters, but as a set of numerical components. These collections are called *vectors*. The most profane representation of a vector is a point in an n-dimensional space. The subject called word vector representation or word embedding [SWY75] uses this thought to represent words as vectors. The main idea is to map words into a vector space preserving the semantic information of a text. A single vector for one word is called a *word vector*. The collection of all word vectors for a vocabulary V is called *word vector space* $\mathcal{V}$. So, each word in a vocabulary must be represented by a geometrical point or vector. Geometrical distance or proximity between these points should show to some sort of *semantic similarity* or dissimilarity between the represented words. The mapping of words to vectors can be done via various concepts, like neural networks, pointwise mutual information or word and context occurrences [SB88]. We will focus on the work done with neural networks by Mikolov et al. [MSC+13] [BGJM16]. Word vector analysis opens the way to powerful

tools, which can be applied to large data sets. The word vector space gives access to methods of linear algebra because it is still a vector space in general. With such a powerful toolbox one can access proximity, similarity and context measures which were not there in the first place. In other words, this means that one can actually add, subtract and compare words or documents like they were points in a vector space.

Using word vector representations for recommendation is a current area of research. It can be employed from sentiment analysis [DSG14] to restaurant recommendations [Das15] to user review analysis [ML13]. Traditional approaches with Markov Models only work on predestinated prediction models [PL99], the new approaches try to take the flexible nature of human behavior into account. A good indicator for this is the natural language. It is as adjustable as the speaker wants it to be. Therefore, it is closely aligned to the flexibility of behavior modeling. The main ideas behind the publications is the similarity of language between individuals. Users use either the same tone for good or bad aspects of things or even for describing them as a whole. Restaurant reviews can detail good service or tasty food, but a customer can also describe the restaurant's atmosphere. This leaves us with two main results. We can access the quality of different aspects of the restaurant with the general tone of the review. Positive language shows a good trait, whereas negative sentiment discloses flaws. The second result gives the opportunity to compare restaurants. If users describe different restaurants similarly, one can assume that they have something in common. This could be the ambiance, the type of service or the food offered. Through the properties of the vector spaces in use, one can use algebraic operations to retrieve restaurant recommendations for users. These properties will be explored in Chapter 4. The basic idea is the *homomorphism* between words, concepts of natural things and their language based description. The homomorphism represents the similar representation of aspects of natural things with similar language. All assumptions described above also apply to the sentiment and review analysis.

WORD2VEC AND FASTTEXT

In recent days word vector representations are closely aligned to Neural Networks and Deep Learning. Mikolov et al. showed with *word2vec* [MSC+13] how powerful vector representation can be for understanding languages. They used a simple neural network with a collection of preprocessing algorithms to model languages in a multidimensional vector space. The neural network transforms sentences or consecutive text to a vector space with an n-dimensional vector for each word in the vocabulary of the text. We will focus on a general explanation of word2vec and fasttext. So, all following concepts are both used in word2vec and fasttext.

Figure 2.1 details the general process. Let the collection of all words in a text be the *vocabulary* V. Each *sentence* s as a subset of words from the vocabulary V will be expanded to a *n-hot vector* $\vec{s}_{01}$ over the complete vocabulary. The dimension n represents the number of distinct words in the vocabulary. So, every vector $\vec{s}_{01}$ for each sentence has n dimensions. An index in the vector is set to 1 if the word at this particular vocabulary index occurs in the sentence s. The order of the vocabulary and the indexes of words are set globally by a preliminary scan of the input text. The algorithm then tries to predict either the context for each word (*skipgram*) or the word for each context (*continuous bag of words*, CBOW). In the first case the input data is the current word and the labels are the words surrounding the current word. In the second case the input data is the context of the current word and the label is the current word. In both instances the loss

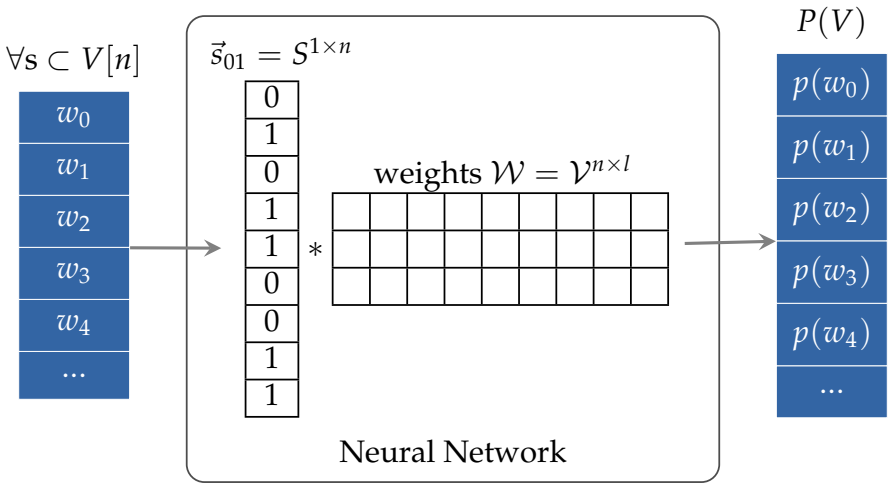


Figure 2.1: Learning word vector representations

target is the accuracy for finding the exact word or context. Both concepts rely on the gradient descent optimization. The neural network reduces the loss in respect to the ground truth over several iterations. The loss in use for word2vec and fasttext is given by the difference between the prediction and the *probability distribution of words* $P(V)$. This distribution is a continuous sampling of how likely it is to get each word given the current input sequence. $P(V)$ is a floating point vector of length n. Skipgram uses the top entries for one input word to model its context. On the other hand, CBOW only optimizes towards the highest probability for one word given a context sequence as input. For further references we used the skipgram approach. The optimization is done for each word in every sentence s. This leads to the typical matrix multiplication concept of neural networks. Unlike other applications for neural networks, word2vec does not return the complete network to be used as a classifier, but only the *weight matrix* $\mathcal{W}$. The final result is a high dimensional vector for each word in the vocabulary. This is called the word vector space $\mathcal{V}$. The dimension n is preserved to represents the number of distinct words in the word vector space. Through the matrix multiplication, the weight matrix $\mathcal{W}$ has an entry for each word in the vocabulary. It is clearly visible that for the multiplication of $S^{1\times n}$ the second matrix $\mathcal{W}$ needs to have n entries in its first dimension. This leads to $\mathcal{W}$ and $\mathcal{V}$ having n entries, one for each word in the vocabulary. l is an arbitrarily chosen number of dimensions for $\mathcal{V}$. It usually spans from 100 to 300. It is set at the initialization of the neural network and defines the number of dimensions for each word vector. l can be tuned via hyperparameter optimization. Due to all the optimizing done by the neural network, one gets a globally stable vector space for words with the domain $[-1, 1]$. The *domain* is derived from the gradient descent with a difference-based loss on probabilities. The gradient only changes the values of the weight matrix $\mathcal{W}$ in a range of -1 to 1 [Sny05].

Words similar to each other in a semantic sense appear closer to each other in the vector space and vice versa. The main assumption is that if two words are similar, they appear in the same context. Therefore, the target probability distributions of both words are similar or the same. The weights of the neural network for the two words are optimized in the same way. So, the two word vectors in the weight matrix approximate to the same values. The vectors themselves have a close proximity to each other. This assumption only holds for a very large training corpus. Mikolov et al. recommend at least 100 million words. This recommendation is derived from the fact that languages have a broad vocabulary. Small texts might only use a small percentage of

it and might lack some vital semantic information. Another influence is the performance of the neural network. Neural networks generally need a lot of iterations for optimizing. A short text for training would lead to *overfitting* [Haw04] of the network to the training text. Overfitting means that the trained model is too specialized on the training data and cannot be generalized to other data sets. An overfitted model would give poor results on other data whilst performing well on the training data. This effect can be averted by using a larger text representing nearly all possibilities of words and their contexts in a language. This would create a generalized model, i.e. a word vector space, for the language.

The concept of word2vec can be improved by a subword n-gram extension, called *fasttext*. fasttext [BGJM16] also uses a simple neural network for transmuting a word in a sentence to a vector in a vector space. Contrarily to word2vec, fasttext does not build the input sequence and $\vec{s}_{01}$ by using words, but n-grams. A sequence contains subparts of words rather than whole words. This gives access to a finer syntactical granularity. It is much easier for fasttext to represent different spelling of words and orthographic errors by correlating similar n-grams in the vector space. A general improvement for the spatial placement of word vectors is reached. The length of the n-grams is given by the user and just like m can be part of hyperparameter optimization.

PROPERTIES OF WORD2VEC AND FASTTEXT

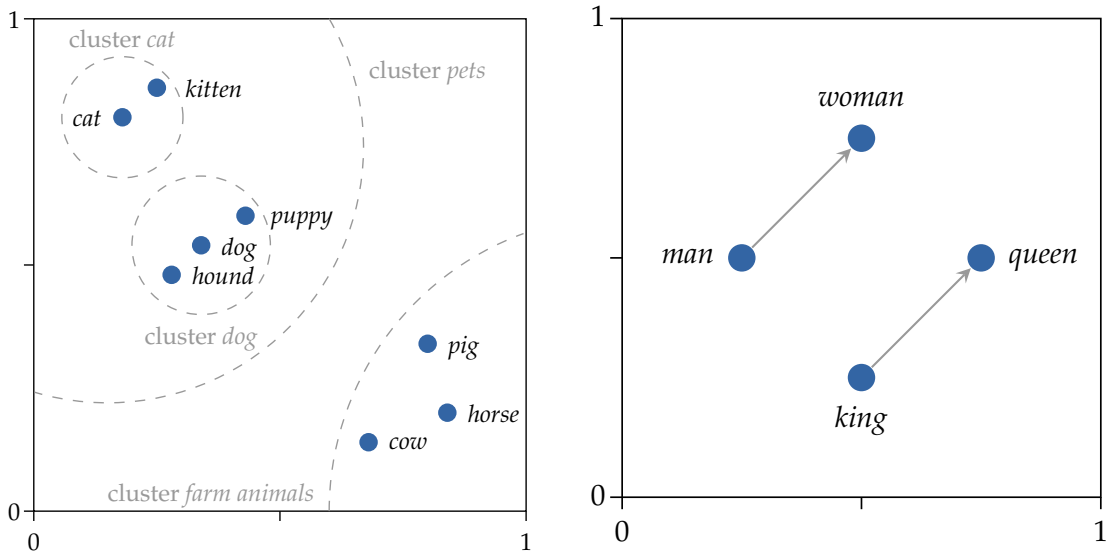
As mentioned in the previous section, the word vector spaces produced by word2vec and fasttext have interesting properties. We want to use this section to introduce two of them. They build the bases for Chapter 4. The first property is the *positioning* of word vectors and their respective words in the vector space. The vector space has local clusters for similar words and synonyms. Additionally, global clusters emerge for similar words in a semantic manner, like words from the semantic concept of *pets*, as shown in Figure 2.2a. This means that directly interchangeable words appear very close in the vector space, but even words with a larger distance can have a semantic connection. It is clearly visible that synonyms for *cat* and *dog* are close to the original words. Both clusters for *cat* and *dog* lie in the super cluster *pets*. Whereas *pig*, *horse* and *cow* have no synonym clusters, but a shared super cluster called *farm animals*. It should be said that there still is a close proximity between all word vectors, because all words and the two super clusters belong to the super cluster *animals*. So, proximity is always measured in a semantic context. A generalization to say far things are dissimilar only holds with this assumption. In our example *horse* is more dissimilar to *dog* than *cat*, but all words are similar in the context of *animals*.

The other property of this vector space is the ability to express *semantic concepts* via vectors. A *concept vector* $\vec{v}_c$ can be expressed with word vectors of antonyms $\vec{a}_1$ and $\vec{a}_2$, like in Equation (2.1).

$$\vec{v}_c = \vec{a}_2 - \vec{a}_1 \tag{2.1}$$

The subtraction renders the distance vector between the two converse word vectors. This concept vector can be used to find antonyms for words representing the same contrary. Given a word vector $\vec{w}$, we can find its antonym $\vec{a}_w$ by adding the concept vector $\vec{v}_c$ as shown in Equation (2.2).

$$\vec{a}_w = \vec{w} + \vec{v}_c \tag{2.2}$$



(a) Example for proximity in 2D

(b) Example for concept *gender swap* in 2D

Figure 2.2: Properties of word2vec and fasttext

Keep in mind that the direction of $\vec{v}_c$ is important for the understanding of the concept. For example, take the vector for *woman* and subtract the vector for *man*. The resulting vector now gives a representation for the concept *gender swap*. Now add the concept vector to the vector of *king* and one will end up with same vector as for *queen*. Figure 2.2b shows a geometric interpretation in 2D for this example. This example shows that the direction of $\vec{v}_c$ points from a male word to a female word. For getting the reverse direction, one has to subtract the two word vectors for *man* and *woman* the other way around. Nevertheless, this opens the way for expressing words or concepts that were not there in the first place. Whereas the word *gender* could be part of the vocabulary, it is just a word with no expressive binary label. The distance between *man* and *woman* however masks the binary concept of biological genders through the vector addition. One can take any word representing a hidden gender, like *ewe*, and add the vector for *gender swap*. In this case, it will lead to the word *ram*. Since the addition is independent from the vectors used, it renders a powerful tool for finding concepts. The vectors in the arithmetic can be exchanged with any concept vector. We will use concept vectors and positioning extensively in Chapter 4.

2.2 BIOLOGICAL INTERACTION ANALYSIS

Biological Interaction Analysis (BIA) is a subcategory of a concept called *Particle Mesh Methods* (PMM)[1]. PMM is a very broad subject. The core idea is the usage of objects as particles in a space. Particles are simplified versions of biological objects. This includes molecules, cells and animals. Particles are placed in a space, called mesh or *context*. Meshes can be very diverse. From simple positions in a n-dimensional space over grids to population habitats, the possibilities to put particles into context are manifold. The context is a very important part of PMM. It enables analysis over distributions of particle positions not only by their own characteristics, but also by the relations to their surroundings. We will use this advantage to build a context-aware ker-

[1] All definitions and assumptions in this chapter are adapted from [HPS10] and [HE88].

nel method in Chapter 5. Particles can contain additional information, like mass, acceleration or type. These are used to enrich the possibilities of modeling. The modeling of parameters with BIA opens the way for simulating the particles in a mesh. We will describe the general idea of BIA and simulation and the connection between them. It is shown how the simulation affects particles. This will yield the bases for all the research presented in Chapter 5.

INTERACTION ANALYSIS WITH CO-LOCALIZATION

Most Interaction Analysis use a concept called *co-localization*. Co-localization-based approaches mainly rely on the density measure of particles in space. This also applies to Biological Interaction Analysis (BIA). The number of measurements for co-localization is broad [MVA93] [ZZO07]. BIA only uses a subset of available measures for density representation. The two core measures for BIA are the *distance co-localization measure* C^t and the *state density* $q(d)$ [HPS10]. C^t is a simple count over all particles i having the distance to their nearest neighbor d_i in a radius of t.

$$C^t = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(d_i < t) \tag{2.3}$$

$\mathbf{1}$ is the indicator function. $q(d)$ gives information about the distribution of particles in a static state. It models the relative frequency of possible distances occurring in the data. In a biological context, this is based on the euclidean distances between objects.

The co-localization analysis can be generalized to *interaction analysis*. This is necessary because $C^t > 0$ is neither a reliable indicator for interaction, nor gives it information about the properties of the interaction, like the strength. Additionally, we want a model where the context of a particle can be modeled into an influence for the interaction. As a general idea for interaction between two datasets X and Y, BIA uses a binary Gibbs process. It allows the interaction to be expressed by an abstract pair-wise function Φ. It models the *energy* of interaction between two particles $x_i \in X$ and $y_j \in Y$. The Gibbs process can be expressed with a Boltzmann distribution.

$$p(X, Y) \propto \exp\left(- \sum_{i=1}^{N} \sum_{j=1}^{M} \Phi(x_i, y_j) \right) \tag{2.4}$$

With the help of the Boltzmann distribution and mathematical transformations [HPS10], one gets a distribution for the joint probability density of observations D with Z as the partition factor and $\Phi(d_i)$ as an *interaction potential* or *potential function*.

$$p(D|q) = Z^{-N} \prod_{i=1}^{N} q(d_i) \exp\left(-\Phi(d_i)\right) \tag{2.5}$$

This joint probability gives us information about how likely an interaction is given the current density distribution $q(d)$. The so called potential $\Phi(d)$ gives information about the characteristics and likeliness of an interaction between objects. $\Phi(d)$ can model different *shapes* f, *strengths* ϵ, *length-scales* σ and *thresholds* t for the interaction given a distance d. The general form is given in Equation (2.6).

$$\Phi(d) = \epsilon \, f\left(\frac{d - t}{\sigma}\right) \tag{2.6}$$

With parameter estimation we can get a potential function $\Phi(d)$ fitted to the current distribution of particles. Choosing a shape f yields different potential models to be used. The simplest model is a *step function potential* between to particles x_i and y_i with distance d_i defined in Equation (2.7). $\phi(d_i)$ expresses the strength of the interaction depending on the distance. A simple example would be $\forall d_i : \phi(d_i) = 1$, which makes the function a step response signal.

$$\Phi(x_i, y_i) = \begin{cases} \phi(d_i) & \text{if } y_i \text{ is nearest neighbour of } x_i, \\ 0 & \text{else.} \end{cases} \tag{2.7}$$

For our purposes this model would be to simple. It can only model the interaction between objects and their nearest neighbor, but a more general model is desirable. It should model the interaction based on all neighbors surrounding a particle. We have chosen the *Plummer potential* [Plu11] which has this more detailed approach for modeling interaction. The threshold t is set to 0. So, $\Phi(d) = \phi(d)$ reads like Equation (2.8).

$$\begin{aligned} \phi(d) &= \epsilon\, f\left(\frac{d}{\sigma}\right) \quad \text{with} \\ f(z) &= \begin{cases} -(z^2+1)^{-0.5} & \text{if } z > 0, \\ -1 & \text{else.} \end{cases} \end{aligned} \tag{2.8}$$

Other interaction potentials are available. Their performance is subject to future research and not part of this thesis. A selection of potentials can be found in Section B in the appendix.

As a last step, one can transform a potential $\Phi(d)$ to a *force function* $F(d)$. We need forces and accelerations for simulating particles later on. As we stated earlier, the potential is the energy of an interaction. The common laws of physics say that the first derivative of any energy is the force. The connection of forces $F(d)$ and *accelerations* $a(d)$ is also known from Newton's second law of motion. The chain of derivations is shown in Equation (2.9).

$$\Phi'(d) = F(d) = m \cdot a(d) \tag{2.9}$$

SIMULATION

The *simulation* is a system applying rules (i.e forces) to a set of particles over a number of time steps. Every advance in time renders a new state of the simulation. A different distribution of particles and their properties is visible to the observer. In our case, we will use the forces and accelerations derived from the potential as a set of rules. The simulation will then apply the rules to the particles moving them around in the simulation space. The forces in dependency on distances $F(d)$ allow calculating the new position $\vec{x}_{i+1}$ of a particle with mass m_i given a distance d to one other particle.

$$\vec{x}_{i+1} = \vec{x}_i + \frac{1}{2} \cdot \frac{\vec{F}(d)}{m_i} \cdot \Delta t^2 + \vec{v}_i \cdot \Delta t \tag{2.10}$$

Note that the vector form of $F(d)$ is derived by applying Equation (2.9) to every dimension of the particle vector. It is also possible that the properties of a particle could change. These could be the acceleration or velocity of a particle. The position of a particle is influenced by a number of other particles in its neighborhood. The *neighborhood* is defined by the radius r_t. All forces from

the other particles in this radius act independently on the particle. This adds up to a sum of forces over all neighbors N with their respective distance d_j in Equation (2.11). This makes a simulation with p particles and $\bar{n}$ neighbors per particle on average run in $\mathcal{O}(p \cdot \bar{n})$.

$$\vec{x}_{i+1} = \vec{x}_i + \frac{1}{2} \cdot \sum_{j=1}^{|N|} \frac{\vec{F}(d_j)}{m_i} \cdot \Delta t^2 + \vec{v}_i \cdot \Delta t \tag{2.11}$$

2.3 SUMMARY

For summarization, we want to introduce a simple example to show the application of BIA and simulation. Imagine animal cells and viruses distributed on the mucosa of a nose. The cells and viruses are modeled as particles with a different type attribute to distinguish cells and viruses. All particles have a 2D-point as their position. The mucosa is the mesh and gives a context and boundary for the processes. BIA tries to describe the course of infection of the cells through the viruses with the interaction potential. This give information about the probabilities for when and how an interaction could occur. Taken the potential, one can transfer it to a physical simulation model with the force derivation. The simulation will show the possible course and outcome of the infection. This gives us a set of tools to better understand and predict virus trafficking.

This chapter has shown the general bases for the research presented in future chapters. We have illustrated the workings of word2vec and fasttext and put their properties into context of their opportunities. This includes similarity measures and algebraic calculations leading to concept vectors. Both properties can be used for segmentation, recommendation and prediction, as we will introduce in Chapter 4. For this, we will combine words, products and customers into one vector space. As a general remark, word2vec has a high impact on current research without a complete understanding of the reasons for its good performance [GL14].

As a second model we introduced Biological Interaction Analysis (BIA). We detailed its general concept and its entailment with simulations. It can alter vector spaces in many different ways. Through the different positioning of objects, external algorithms can interpret the vector space differently. We will use this side effect to improve our purchase prediction. We will gain a higher accuracy in a classification task and therefore a more precise prediction of purchases. Given our market research use case, we are going to set up the simulation accordingly. The use of BIA for our purposes is depicted in Chapter 5.

3 DATA CHARACTERISTICS

As mentioned in the motivation in Chapter 1, large amounts of data are a common source of information in market research. To get a close use case for our research, we used a data set from the real world. This data set is already used for many different purposes in market research by the GfK. Therefore, we can model our algorithms and concepts around a practical example. The data at hand are parsed customer receipts from online purchases from different suppliers and merchants. The main source are email accounts of US citizens where orders are extracted from incoming order receipts. The orders span from 2014 to 2016. The data is not restrained to one supplier, but gives information from the customer's point of view. Therefore the data models a more complete part of the market, compared to the data collected by just one supplier. The data consists of 300 million tuples resulting in 2.1 million distinct customers and 40 million distinct products. Each tuple contains one product purchase with a certain price and quantity. This is called an *observation*. Several product purchases make up an *order*. Each order is directly connected to a customer. Every tuple also contains detailed information about the order and the purchased product. This includes taxes, discounts, vouchers and product titles. Table 3.1 shows an excerpt of mentioned columns.

First of all, one should get a better understanding for the data at hand. A better understanding for the data is vital for proposing hypothesis and modeling concepts around it. We assume that a broad describing overview over the data will help us in our research. In market research, there are some common indicators. This chapter should be used to present some important indicators. These *Key Performance Indicators* (KPIs) [FG90] can also be used to transcribe hypotheses about the data and the models build on it. We will access five major categories of information: purchases per customer, purchase frequencies over time, purchases per product category, co-occurrences of product categories in orders and language information in the product titles. All five indicators are chosen accordingly to our further work. They will be explained thoroughly in this chapter.

user_id	product_id	order_id	order_total	product_title	product_price	quantity	...
eads-34	tef-34	09ed	28.98	Shampoo No. 5	18.99	1	...
eads-34	bhf-5	09ed	28.98	Adm. Ackbar's Cornflakes	9.99	1	...

Table 3.1: Data Example

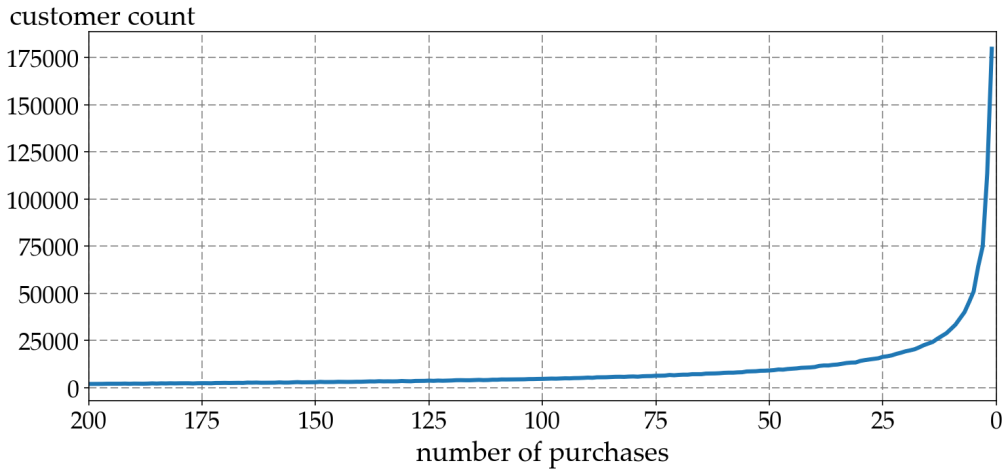


Figure 3.1: Distribution of purchases per customer with less than 200 purchases

NUMBER OF PURCHASES PER CUSTOMER

The first indicator relates the number of customers to the number of their purchases. One purchase is defined as one observation in an order. An observation can have a quantity greater one for a product. Figure 3.1 shows this distribution. It is clearly visible that there are many users buying less than ten items in total. Even though the graph declines fast towards higher numbers of purchases, there are still around 2000 customers with 200 purchases. We assume that there are enough so called *high frequency buyers* with more than 25 purchases for further analysis. The maximum number of purchases per customer is approximately 5000. For better visualization the x-axis of the figure is cut off and not shown in its complete extend.

PURCHASE FREQUENCIES

Our next data reconnaissance task will be about the number of orders per hour. Different daytime hours produce different shopping behavior. Traditional retail shopping has a peak around 16:00 o'clock, when people come from work and do their grocery shopping. The low point of purchases occurs between 2:00 and 3:00 o'clock in the morning. Most shops are closed and most people are asleep at that time. The first major problem we run into are time zones. The x-axis has two labels, one for Pacific Standard Time (PST) and one for Eastern Standard Time (EST). The USA (main land) consists of four time zones, but we cannot access the time zone from the order's timestamp. We have to assume that our data is collected through all four time zones. This makes it hard to correlate time to peaks and lows. The shopping patterns in different time zones are not distinguishable anymore. Purchase anomalies can be overlapping or canceling each other. The resulting diagram can be found in Figure 3.2. Despite the issue with time zones, there are some interesting patterns to be found. We can observe a peak at 16:00 o'clock PST. This is unexpected because we assumed different patterns for online shopping compared to retail shopping. Through the accessibility of online shops from everywhere at any time, people can order products via mobile devices or from their workplace. The all day open access to online shops would lead to the assumption of a flatter curve. Still, this typical observation for online shopping can be found as a steady high of orders during daytime in our data. This makes the data look like a plateau

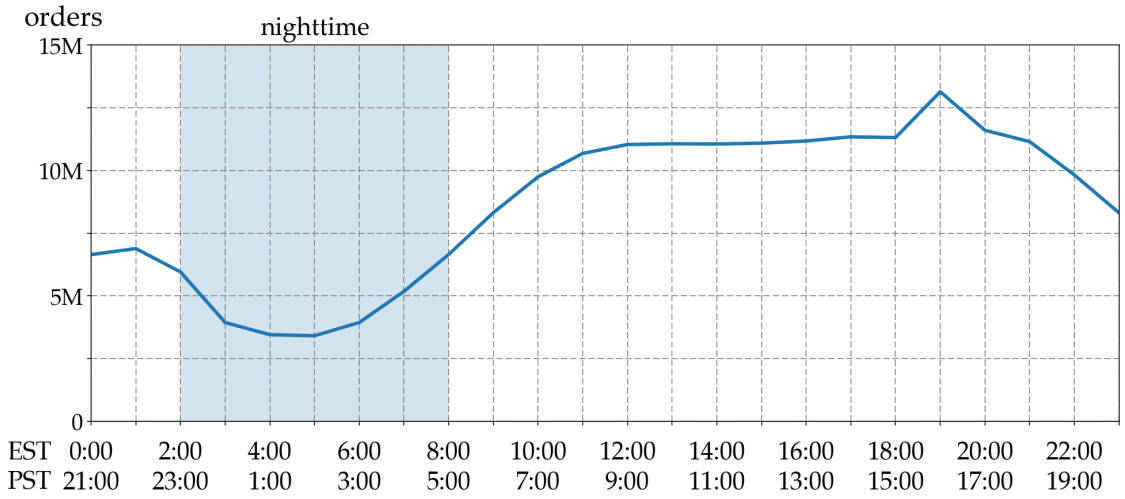


Figure 3.2: Purchase frequency per hour

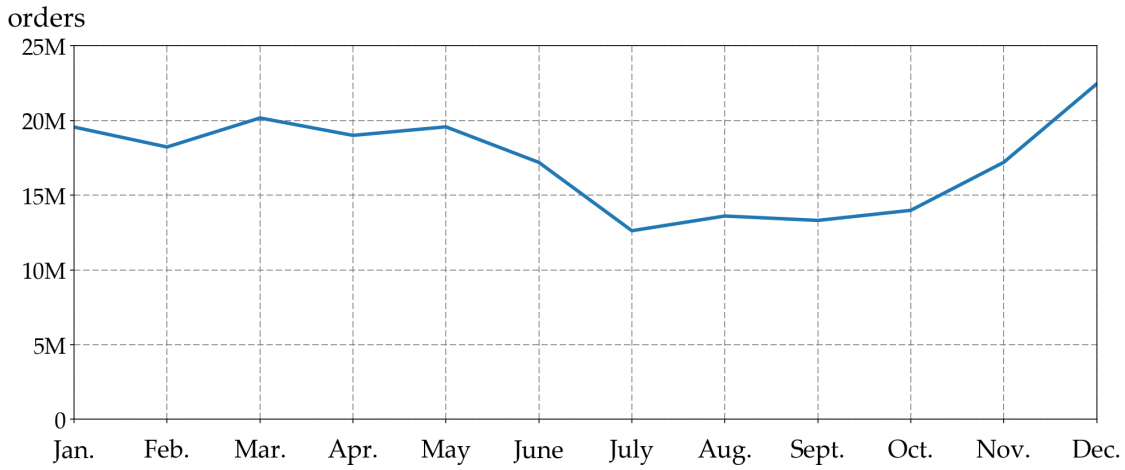


Figure 3.3: Purchase frequency per month

between 10:00 EST (7:00 PST) and 22:00 EST (19:00 PST) o'clock. Another shared thing with retail shopping is the drop in purchases at night. In our timeline, we have defined nighttime as the time where most US-citizens are probably asleep. The ranges span from 2:00 to 8:00 o'clock in EST and 23:00 to 5:00 o'clock in PST. This scope encloses the drop in sales very well.

Another time aspect are the sales over a year. Figure 3.3 shows the orders per month for the data's three year cycle. There are two things visible right away. The first one being the increase in sales during December because of Christmas shopping. Just like traditional retail shopping, online shopping has the same reaction to this particular season. Christmas is generally known as the time of the year with the highest sales. This rush can be seen every year and it is not surprising seeing it in our data as well. The other interesting sequence is the decline in sales during July. This is the holiday season. Most people are away and do not purchase products at this time. Or, they have spent a large amount of their yearly budget on the holiday and now need to regain funds for purchases. Again, this is not a special scenario, but a general observation in online and retail shopping.

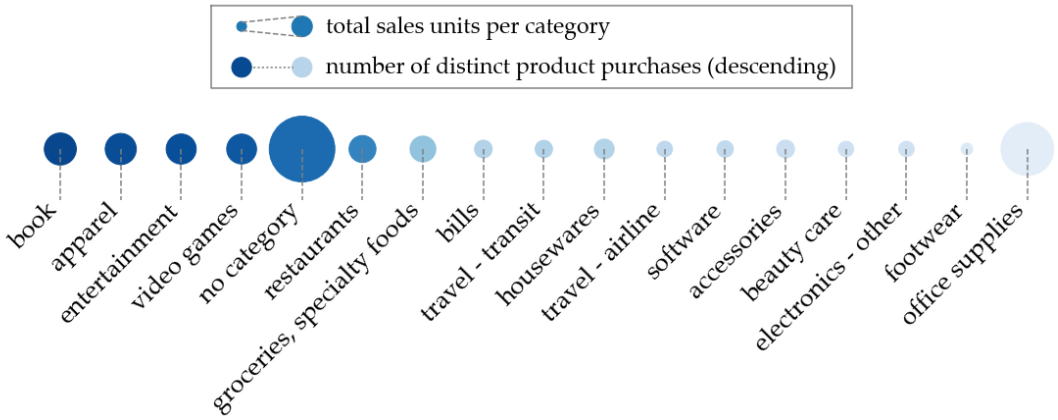


Figure 3.4: Distribution of total sales units and distinct purchases per category with more than 4M purchases

PRODUCT CATEGORIES

Further analysis requires a closer look at product categories. The data contains of 39 major product categories which are divided into 53 minor product categories. These can be found in the appendix, Section A. The next two indicators are used for a joint interpretation. Figure 3.4 details the total number of units purchased and the distinct products purchased for each category. The total sales are represented by the radius of the point. Ignoring the amount of not categorized products, one can see that office supplies have the highest count of units sold. In order of their magnitude, two general groups of products can be observed in the following data points. The first one contains typical *online categories*, like books, apparel and online food orders, having a relatively high count of units sold. The second one is made of so called *emerging categories*. These categories are products which are currently in the transition of becoming an online product. This includes travel bookings and beauty care. The data is also color-coded. The bluer a point, the more different products were purchased in that category. The data is sorted by intensity from left to right. The main online categories stand out in this visualization, too. They offer a wide variety of different products to be purchased. The best examples are different book titles, giving the book category the superiority over others. It is distinctive that the distribution into two groups already present in the sizes, also exits in the coloring. The only exception are office supplies. Office supplies might be purchased in large amounts, but only a few different product types are put into an order. This details the contrariety between the size and color of this node. One can easily imaging paper being ordered for an office. Usually this happens in large amounts and there are no other products purchased with this order.

Another interesting find is that Figure 3.5 shows a similar order of categories for distinct products offered, like the previous one did with the color coding for distinct product purchases. This points towards a directly proportional dependency between distinct products purchased and distinct products offered for each category. This a natural thing to assume. The customers will buy diversely if the product range of a category allows it. In the opposite direction, the market will try to fulfill the need for a divergent product range if the demand is there. One outlier are transit travels. These product titles always include a start and end end point of a journey. Therefore they have a high count of distinct routes, which are counted as separate products. Compared to other categories the sales for transit travels are low. That is why they do not appear in Figure 3.4.

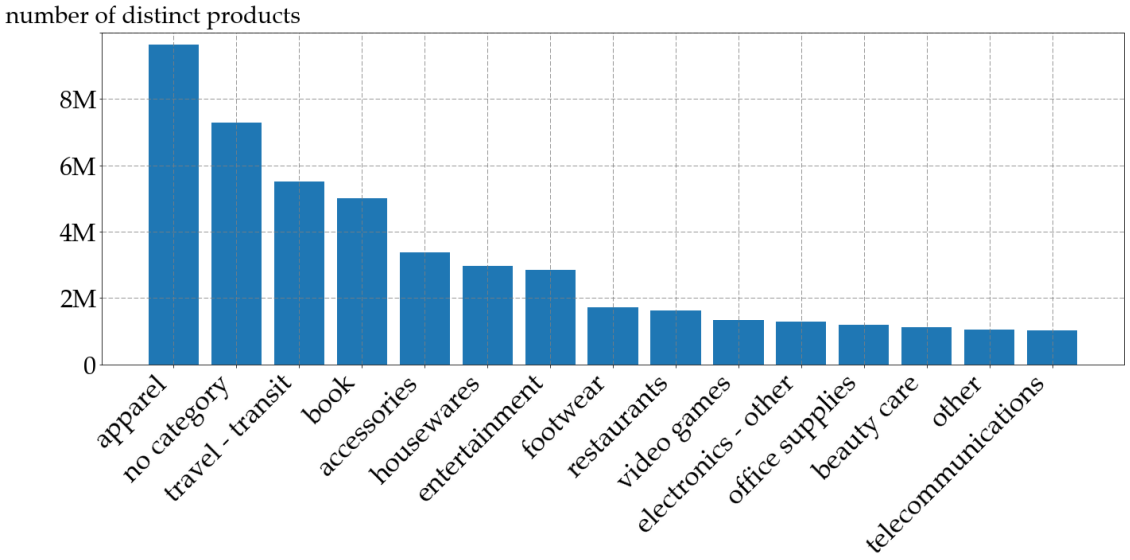


Figure 3.5: Distribution of distinct products per category with more than 1M products

PRODUCT CATEGORIES CO-OCCURRENCES

For the last insight, we examined the *co-occurrence* of different product categories in orders. Figure 3.6 shows an overview over the most common categories and their co-occurrence in an order. A category co-occurs with another if they being bought together in a single order. We summed all single orders to get the total co-occurrence of two categories. The main diagonal is colored differently because the scale for these points is much larger, but we still wanted to compare co-occurrences of the same category with co-occurrences of different categories. The alignment of the two ranges makes the diagonal comparable to the surrounding data. Another point is the axial symmetry of the matrix considering that co-occurrences are counted symmetrical. There are three types of categories to be found. The first one contains categories with a closely aligned identification, like apparel, accessories and footwear. The same is valid for entertainment, software and video games or beauty care and personal care. All categories in this type feature a common topic. The second type are the self-referential product groups, like apparel, books, airline tickets, groceries, event tickets and video games. These are products, which are most often bought in a bulk of the same category. This seems logical for most of them. For example, air travels are booked together if the customers needs a connecting flight. And then there is the last category, which we called interesting co-occurrences. These are products purchased together, where one would not assume it, e.g. housewares are apparently often sold with books or apparel.

WORD COUNTS

In this section, we want to take a look at the product titles and their language. Product titles can be seen as very dense sentences of an artificial language. The density relates to the amount of information found in them. The titles can transfer a lot of content to a customer without using standard paradigms of language. For example, the product titles do not include a predicate like normal sentences in the English language. They also do not follow the traditional subject-verb-

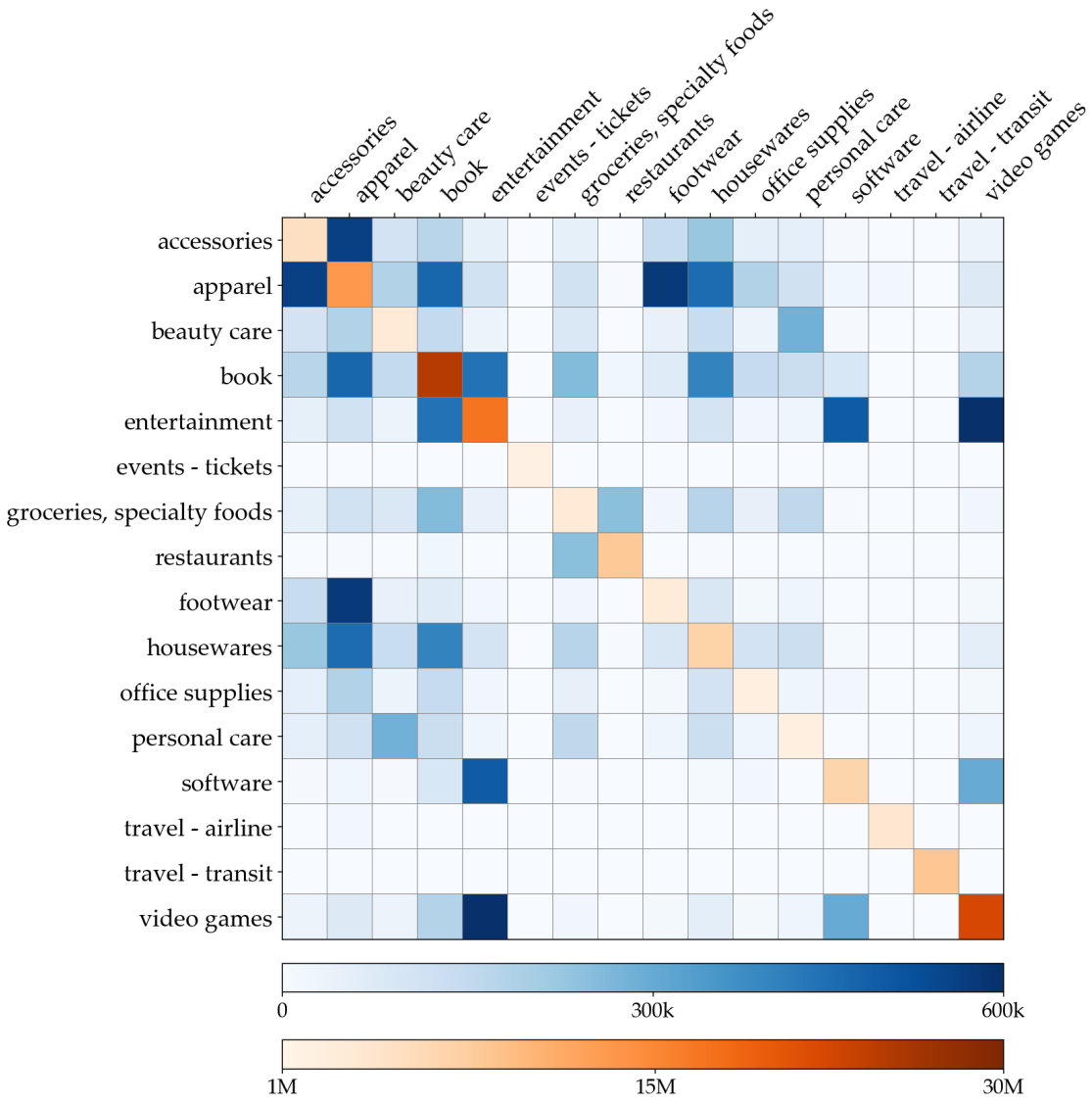


Figure 3.6: Co-occurrence of different product categories in a single order

object order. There are a lot more restrictions which we did not name. We will explore two indicators for the language of product titles called total word counts and the word counts per product category.

Figure 3.7 details the absolute counts of the top 30 words in the product titles. This is without numbers and special characters. The top places are hold by stop words which do not give any additional information to the product title, but can be seen as filler words. That is why they occur very often. The following words are from several major word categories. We can find representatives of products, product characteristics and product categories. Examples are *case* for the product itself, *black* for the product characteristics and *book* for the product categories. Additionally, the range of semantic meaning of the top 30 words is very broad. We cannot identify large synonymous groups of words. This is because the word counts are collected over all categories which have a different vocabulary each.

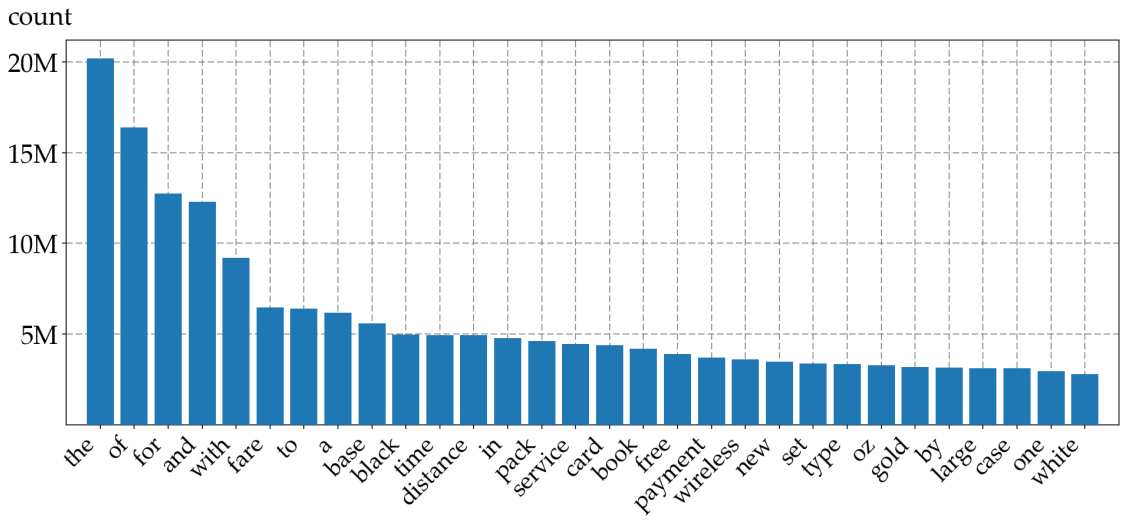


Figure 3.7: Count of the top 30 words in the product titles

category	top word
book	book
apparel	women's
entertainment	season
video games	gold
restaurants	pizza
groceries, specialty food	organic

category	top word
travel - transit	fare
housewares	set
accessories	silver
beauty care	sample
foot wear	women's
office supplies	number

Table 3.2: Most used words for some product categories (without stop words)

After seeing the total counts of words, we are interested in the most common words in each product category. Table 3.2 shows the most frequent words for some of the product categories from Figure 3.4. We excluded stop words from the selection. They would have been the most used words in every single category. We can see that most of the words have a logical connection to the product categories. A lot of words describe a specific subpart of a category. The word *season* shows the purchases of a television series in a seasonal format. It is therefore distinctive for the entertainment category. The same applies to *organic* as a product type for specialty food and *pizza* as a restaurant dish. This details how closely words are aligned to their language-based context. The basic idea is that each word is a representative of its category because of its typical use in this category. We will use this to our advantage for finding customer and product segmentation in Chapter 4.

SUMMARY

We have gathered a broad spectrum of information in this chapter. We analyzed the purchases per customer, product category and month. Additionally, we looked into the co-occurrence of categories and the language of the product titles. With all the presented insights, we feel confident about how to use the data for our research. This is not only for conventional methods, but for off-

id	no. customers	no. products	no. purchases per customer	usage
1	121,000	–	>2	customer segmentation
2	696	500,000	>25	purchase prediction, kernel trick
3	2000	500,000	>30	purchase prediction, kernel trick
4	24,000	500,000	>25	purchase prediction
5	53,000	500,000	>30	purchase prediction
6	910	37,000	>30	performance enhancement
7	–	310	–	product segmentation

Table 3.3: Test sets used for our models

the-spectrum solutions, too. We gained more information about the structure of the data at hand. The five main points in our data exploration give a broad scope for analysis. This will help us modeling different concepts with the data. For example, in Chapter 4 we will use our knowledge about the product title language to put words, products and customers in a combined vector space where we can compare them. Additionally, we will utilize product categories for predicting the next purchase of a customer.

For further references, we will use several test sets for our models and algorithms. They are detailed in Table 3.3. The first data set is used in Section 4.3 for customer segmentation. The second, third and fourth data set are high frequency buyers. The will be used in Section 4.5 for purchase prediction with classification and in Section 5.2 to show the improvement of the classification with BIA and the simulation. The second data set gives us a small example of the data to test our algorithms on. The larger tests set with 24,000 and 53,000 customers are used to verify the findings from the small test set. Both data sets contain products as a context for the customers. The fifth data set is a smaller test set of high frequency buyers. It is used for performance improvement in Section 5.2. It has a reduced number of products to show their effect onto the performance of our algorithm. The last data set are product titles from Amazon's apparel category. They are labeled with the gender of the desired target group. The data will be used in Section 4.4 for product segmentation.

4 NATURAL LANGUAGE MODELING

The word vector spaces as derived in Section 2.1 can be used in a market research context. Since our data is not collected with the purpose of Natural Language Processing (NLP), we need to adapt word vector representation to a format suitable for our research. The only language to be found are the titles of products. They are made up from words and therefore can be seen as a *bag of words* [Har54]. The bag of words can be used for representing the products as vectors as well. Additionally, one can use the purchased products of a customer to put them into the same vector space. The main idea remains that the use of language can give a better understanding of similarity between products and customers. We will show an approach to put words, products and customers into one vector space in Section 4.1. A major part of market research are recommender systems. We will employ our vector space to give language-based recommendations. Customers can get recommendations by finding the products most similar to them. We will show that the word, product and customer vectors can be used for recommendations in Section 4.2. We will also model users against concepts, like social states, and therefore classify users into customer groups with similar purchase behavior in Section 4.3. A similar approach for classifying products into concept categories is shown in Section 4.4. As a last proposal, we will use purchase behavior of a customer throughout different product categories to predict next purchases. This will lead to a concept for a recommender system sampling product categories which are likely to be purchased by the customer. The purchase prediction is introduced in Section 4.5.

4.1 PRODUCT AND CUSTOMER VECTORS

As a first step, we need to make customers and products comparable. Products and customers usually have very different sets of properties. This makes comparing them directly complicated. By using the word vector space introduced in Section 2.1, we have a strong base for comparing vectors. Since the dimensions of the words are chosen arbitrarily, we do not need to use properties of the products and customers to model single dimensions of the vectors. However, the disad-

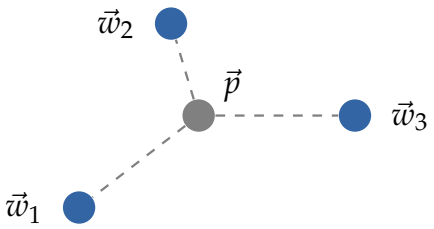


Figure 4.1: Centroid calculation in 2D

vantage is to find a mapping for the products and customers to the word vector space. We have used fasttext to build a vector space as described in Section 2.1 from the product titles. fasttext expects a continuous text, but the product titles are not a natural language. They do not occur in prose-like form. We had to manipulate the titles to resemble an usable input for fasttext. Every single product title is handled like a sentence in natural languages. This is done by listing all products line by line with line breaks separating them. Of course, one cannot assume the same structure as in natural phrases. Product titles are missing vital parts of sentences in a natural languages, like predicates or subject-verb-object order. We used the structural stability of fasttext over a large amount of words to avoid complications with this lack of syntactical objects. The nature of fasttext and the structure of the data allows this simplification. If the context structure of the input data is the same for all training sentences, fasttext can still preserve the reduced information. Words in the product titles still appear in similar contexts. These contexts are completely different to the ones found in natural languages. The main assumption is that they still have some sort of globally structured context. After training the word vector space over all product titles, we employ a simple method for mapping the products to the vector space. For this we take each word from a product title, get its vector from the word vector space and calculate the centroid for all the single words. This gives us a representation for product in the word vector space. The *product vector* has the same dimensions as the word vectors and is therefore directly comparable. Given a set of word vectors $\vec{w}_i \in \mathcal{X} \subset \mathcal{V}$ with $\mathcal{V}$ as the vector space and $i \in \mathbb{N}$ the *centroid* $\vec{p}$ (cp. Figure 4.1) is defined as [Wei02]:

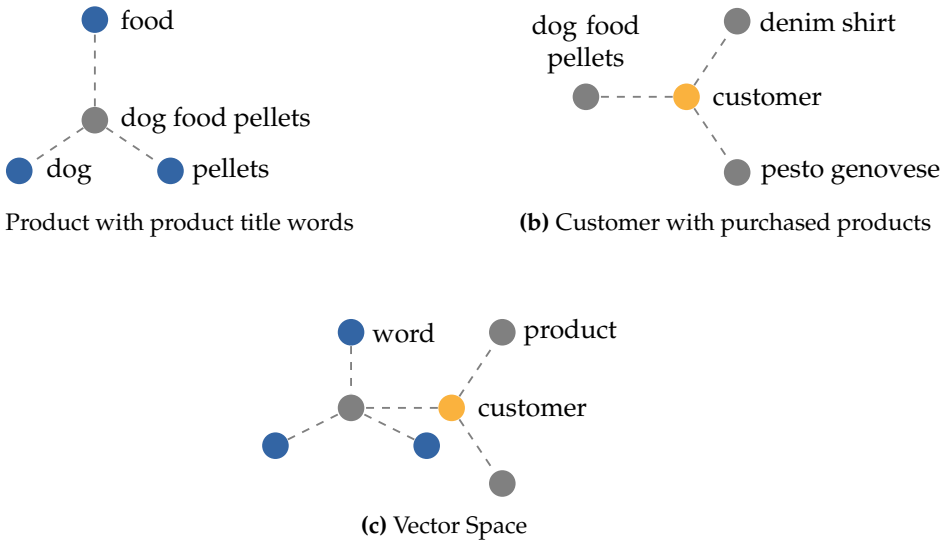
$$\vec{p} = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{|\mathcal{X}|} \vec{w}_i \tag{4.1}$$

The product vectors are added to the vector space $\mathcal{V} = \mathcal{V} \cup \mathcal{C}_P$ with $\mathcal{C}_P = \bigcup \vec{p}$.

With the product vectors we can also model the customers by calculating their centroid. This time, instead of using the words of a product title, we use the vectors of the products the user has purchased. This changes the Equation (4.1) for a centroid of a customer $\vec{c}$ to:

$$\vec{c} = \frac{1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} \vec{p}_i \tag{4.2}$$

With $\vec{p}_i \in \mathcal{P} \subset \mathcal{C}_P$ and $\mathcal{P}$ as the customer's purchased products. The *customer vectors* are also added to the vector space. It follows $\mathcal{V} = \mathcal{V} \cup \mathcal{C}_C$ with $\mathcal{C}_C = \bigcup \vec{c}$. The *combined vector space* now consists of word, product and customer vectors. Figure 4.2a illustrates the calculation of a product with three word in its title. Figure 4.2b shows a customer after purchasing three different products. All three vector types share the same dimensionality l used in fasttext. We have chosen $l = 100$ for our purposes. This is the smallest recommended value for l [BGJM16]. We assume that if our algorithms work with $l = 100$ they will also work with $l = 300$. With the reduction to 100,



(a) Product with product title words

(b) Customer with purchased products

(c) Vector Space

Figure 4.2: Examples for centroid calculation in 2D

calculations are faster and the memory footprints of our test set are smaller. All types of vectors have therefore the length 100. The interchangeability of word, product and customer vectors and the resulting direct comparability is shown in Figure 4.2c. This opens the way for comparable distance measurements between the three types of vectors. A measure comparing vector $\vec{x}$ and vector $\vec{y}$ is for example the *cosine similarity* in Equation (4.3).

$$d_c(\vec{x}, \vec{y}) = \cos(\theta) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|_2 \cdot \|\vec{y}\|_2} \tag{4.3}$$

4.2 PRODUCT RECOMMENDATIONS

Our centroid-based vector space makes customers, products and words comparable. This can be utilized for several market research tasks, like product recommendation. The main goal of *recommendation* is to find products, which are similar to ones purchased before. We will focus on the *book* product group as an example for suggestions. First, we utilize our vector space to arrange the products. We choose a customer as the starting point. We then use a *nearest neighbors approach* to get all products close to the starting point. For finding the most similar neighbors of a customer, we used the cosine similarity from Equation (4.3) between each book and the customer. Items already purchased by the customer were excluded. Table 4.1 shows an example of previous purchases on the left and recommendations for the customer on the right. This is a very simple approach. Its accessibility only holds for single product groups via filtering. It can be used for all products close to customer, but then we might get false positives through the high variance in products close to a customer. These are recommendations of products, the customer is not interested in. The quality of such a recommender system is hard to access without the verification of experts. As of publication of this work, we do not have a gold standard for the recommendations. We only can access the quality of it by sampling some test customers. We have presented one of them in Table 4.1.

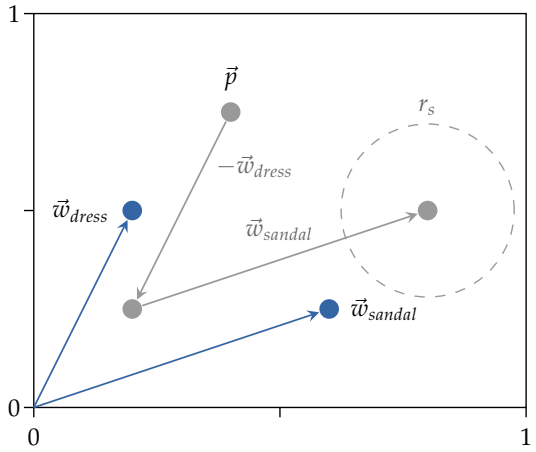
marriage: tips and advice on how to maintain a healthy relationship [...]
self-help box set: techniques & strategies for greater mind power [...]
learn how to use focus to improve your concentration [...]
the art of becoming clutter free: decluttering your life in minutes [...]
time management [...]

(a) Book purchases

help guides to help you improve your skills and make your life easier [...]
amazing tips on how to improve your emotional maturity [...]
50 tricks to live a happier and successful life [...]
33 techniques to unlock the power of your mind [...]
the simple approach to health: a practical guide to understand your body [...]

(b) Book recommendations

Table 4.1: Book recommendations with nearest neighbors approach for an example customer



(a) Operations in the vector space

	h&m peplum dress
	⇓
1	guess women's kymma strappy platform dress sandals
2	carter's strappy sandals
3	bcbgeneration ronny chunky heel sandal
4	b.o.c emmeline dress sandals
5	emporio armani black leather espadrille wedge heel t-strap sandals

(b) Top 5 recommendations

Figure 4.3: Product recommendations for finding matching sandals for a dress

Our second approach utilizes the semantic concept algebra (cp. Figure 2.2b). This time, instead of a customer as starting point, we use a product the customer already bought. We will focus on the apparel product group. Let the customer choose a specific product $\vec{p}$. Our algorithm takes the product vector, subtracts the word vector for the original concept of the item and adds the word vector for the targeted concept of items. The result is a vector around which we employ a nearest neighbor search (cosine similarity) with a threshold r_s. The nearest neighbors are returned as recommendations. The result vector $\vec{r}$ is obtained from the word vector for the original concept $\vec{w}_o$ and the word vector for the targeted concept $\vec{w}_t$ via Equation (4.4).

$$\vec{r} = \vec{p} - \vec{w}_o + \vec{w}_t \tag{4.4}$$

Imagine the customer bought a summer dress and is now looking for a matching pair of sandals. Figure 4.3a shows the calculation in the vector space. First we subtract the the concept of dress from the product. Therefore, we subtract the vector for the word *dress* $\vec{w}_{dress} = \vec{w}_0$ from the product vector $\vec{p}$. After that, the vector of the target product group is added. This is done by adding the vector for the word *sandal* $\vec{w}_{sandal} = \vec{w}_t$. The resulting vector is the starting point for a search of nearest neighbors in r_s. Figure 4.3a also illustrates the proximities between products and concepts. One can clearly see how close the original summer dress is to the concept *dress* and

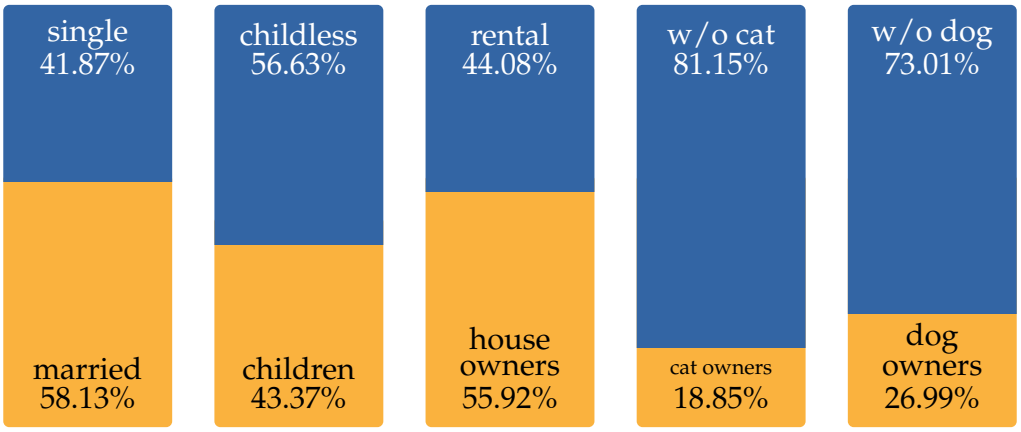


Figure 4.4: Customer composition

how the new product must be near the concept of *sandal*. This means that the suggested items, found by the nearest neighbor search, are sandals. This leaves us with the results in Table 4.3b. It is clearly visible that all returned products are indeed sandals. A quick lookup confirms that the sandals match the style of the dress. Therefore the algorithm can deliver recommendations, which preserve stylistic information just by analyzing the language of the titles. This can be helpful in a recommendation context.

A recommender system could not only suggest products according to stylistic properties, but also can be used in an exploratory way. The customer can add concepts, like colors, which can be translated to a single word and the recommendation would prefer products with this color. For example, the user only wants sandals in black, so the algorithm adds the vector $\vec{w}_{black}$ to the final result from the algorithm and returns the products in the neighborhood of this new point. Note that this only works if the color can be found in the titles. Once again, we do not have a gold standard for the recommendations, so we had to access the quality by sampling customers.

4.3 CUSTOMER SEGMENTATION

Even with good item recommendations one cannot cover the complete needs of a customer. Advertisement and recommendation are usually chosen according to the customer or target group, which the customer belongs to. *Customer groups* can be *married women over 30* or *families with more than two children*, just to name a few. The general process of classifying customers into groups is called *customer segmentation*. In our case, we got information about the composition of customers on a global scale from a third party source. This source mainly uses email hashing and register lookups for classifying customers. The customers in our data set are distributed like in Figure 4.4.

The main goal is to predict the customer's profile in regards of the target groups he belongs to. We have chosen the ten groups from Figure 4.4 and try to classify every customer into those. This is done by using concepts. A concept, introduced in Section 2.1, is a category of objects described by one or more words. For reproducing this customer segmentation, we used single words to describe each concept. For example, the concept of being married is represented by the word *marriage*, whereas the concept of being single is described by the word *alone*. Note that single

in this case also includes divorced and widowed. Therefore the concept is better modeled by the word *alone* than by the word *single*. This also applies for the next concept pair. Customers with children are modeled around the word *children*, but childless customers are also represented by the word *alone*. The argumentation is similar to the previous one. Childlessness can occur in different forms and needs to be addressed in a broader way. This is why we have chosen *alone* over *childless* and other words. The house ownership concepts can be expressed with *house* and *rental*. The vectors for the contrary words in each concept are called *concept vectors*. A concept described by two opposing words is called *contrary concept*. To get a classification into one group for each opposing pair of words, we calculated the cosine similarity to both concept vectors describing the concept. The customer is then classified into the group of a pair whichever yields the higher similarity. Equation (4.5) formalizes the algorithm with G_c as the resulting group, d_c as the cosine distance, $\vec{c}$ as the customer vector and $\vec{w}_{pos}$ and $\vec{w}_{neg}$ as the concept vectors for the words of one concept.

$$G_c = \underset{\vec{w}}{\operatorname{argmax}}\, d_c(\vec{c}, \vec{w}) \quad \text{with } \vec{w} \in \{\vec{w}_{pos}, \vec{w}_{neg}\} \tag{4.5}$$

Contrarily, owning a cat or a dog are *unary concepts*. Unary concepts are not alternative. These cannot be expressed with contradictory words. There are no matching words for not belonging to a unary concept. For example, owning a cat does not automatically mean that the customer owns a dog, whereas having no children automatically means that one is childless. To tackle this issue we calculated the cosine similarity to just a single word $\vec{w}_{pos}$ representing the concept for each customer vector. We then normalized the similarities for one concept over all customers. The decision if the user belongs to this concept is true if the normalized similarity to the word is greater than 0.4. The customer is assigned the label 1 for this concept. Otherwise the customer will be labeled with 0. Equation (4.6) formalizes the process with G_u as the label for an unary concept.

$$G_u = \begin{cases} 1 & \text{if } \left\| d_c(\vec{c}, \vec{w}_{pos}) \right\| > 0.4, \\ 0 & \text{else.} \end{cases} \tag{4.6}$$

We used this approach for the cat and dog ownership. The words for the two concepts are *cat* and *dog*. The algorithm assigns the label for both concept to each customers. To get a global view onto the concept, we add up all customers with label 1. This gives us the percentage of customers with a cat or a dog. We like to point out that choosing an arbitrary threshold of 0.4 can lead to overfitting. An idea to overcome this issue is the usage of statistical analysis of the distribution of cosine similarities for an unary concept. We could use the average or other statistical measures over the similarities as a threshold. This would make the classifier less prone to overfitting.

For proofing our concept, we have sampled 120,000 customers with more than two purchases and calculated their segmentation. This refers to dataset 1 in the summary of Chapter 3. All the mentioned approaches lead to the results in Figure 4.5. With an absolute deviation of two to four percent points and a relative deviation of four to ten percent, we have shown that the algorithm gets marginally close to the original ground truth. This means that customers can be modeled into social groups just by looking at their purchases. More specific, the language behind the product titles gives us an understanding of how close a customer is to a social construct, like marriage. This assumption is quite clear for two reasons. The first one being the fact that the product titles mask several concepts by co-occurring words. The words *white dress* seem to be used more often for wedding dresses and are therefore closely aligned with the concept of marriage. The second reason is the origination of product titles. Products are always labeled for representing

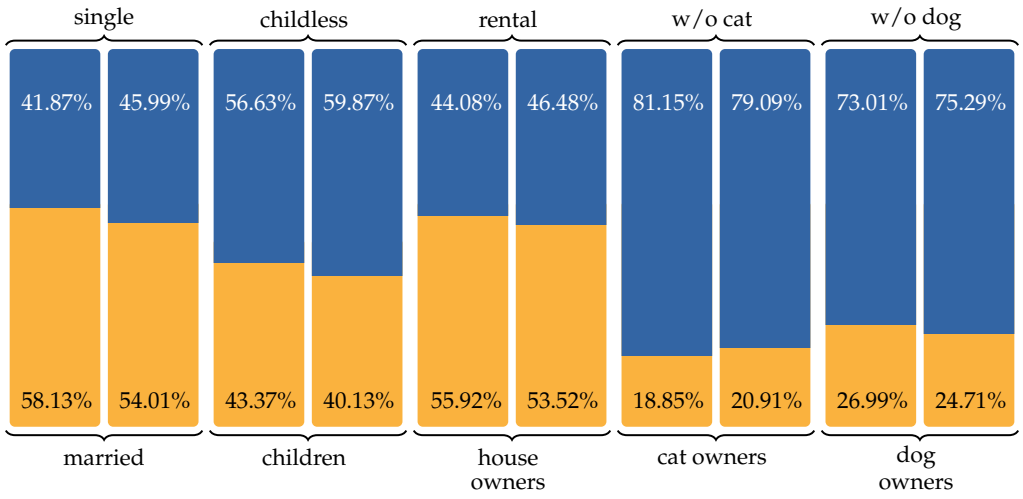


Figure 4.5: Original segmentation (left bar) and predicted segmentation (right bar)

the maximum amount of information in as few words as possible. This makes product titles semantically dense. Thereby a product manager will give the product a name suitable for the desired target group. This aids our algorithm to find not only obvious concepts, but also words and products, which are not connected to the concept at a first glance. Note that the algorithm only works on a global level. For detailed information for a single customer one would need to find a better ground truth. For example by letting market experts label a subset of customer. Another critical point is the problem of shared accounts. If two or more people use one account for shopping and they belong to two different social groups, like male and female, one cannot make any assumptions about their social background. The purchases in this particular account would be too different and would eradicate each other in the customer's centroid calculation.

There are still possibilities for improvement. A first step would be a more flexible threshold for unary concepts, like stated before. We could also think of a more complex classifier the contrary and unary concepts. The concept vectors could be expanded to include multiple words. Our approach only uses one word for $\vec{w}_{pos}$ and $\vec{w}_{neg}$. We could apply our centroid calculation to average two or more words to represent one side of a contrary concept. The concept of not being married can be described with the centroid of the word vectors *single*, *divorced* and *widowed* instead of the word vector for *alone*. The expectation would be a better segmentation through the finer description of the concept with more words. This assumption can also be reversed. The classification can model more than five concepts. One can easily employ the same algorithms to an additional contrary concept of *divorced* and *married* and classify customers accordingly. This would require a better test set where the labels for the concepts are available for every single customer and not on a global scale.

4.4 PRODUCT SEGMENTATION

The concept of word representation for segmentation also works for products. The goal is to put products into different segments. This called *product segmentation*. We can use the concepts vectors on products in the same way as we did on the customers because of the combined vector

product titles	baseline algorithm	concept vectors
w/ *women* and *men*	88.7%	97.7%
w/out *women* and *men*	0.0%	93.5%

Table 4.2: Accuracy product segmentation

space. Equation (4.5) can be rewritten to find the desired segment G_p of a product $\vec{p}$.

$$G_p = \underset{\vec{w}}{\operatorname{argmax}}\, d_c(\vec{p}, \vec{w}) \quad \text{with } \vec{w} \in \{\vec{w}_{pos}, \vec{w}_{neg}\} \tag{4.7}$$

The algorithm stays the same otherwise. We expand the example from the previous section. The words *women* and *men* are representing their respective gender group. Since we do not have any information about the gender orientation of a product, we crawled Amazon's apparel category for 310 product titles with gender labels. This data set is also known as data set 7 in Table 3.3. We also have chosen a baseline classifier, which looks directly for the words *women* and *men* in the titles and classifies accordingly. Our approach is the same as previously described. Calculate the cosine similarities to both words and whichever has the the higher similarity labels the title. As a last step, we removed the words *women* and *men* from the product titles and rerun the classification. The removal prevents the classifier from overfitting to the two concept words. Table 4.2 details all outcomes. One can clearly see the good performance of our classifier, even without the words *women* and *men* in the title. This points towards a stable classifier capable of distinguishing genders from any product title. We would like to point out that the Amazon product titles are not in the training set for fasttext. It also reinforces the assumptions made about the stability of finding status groups. Product titles have a strong tendency towards concepts described by single words. The prerequisite still is a good model for the vector representation of the language in use. On the other hand, the language does not need to be a natural language. An artificial language, like the one for product titles, is enough to model a vector space.

4.5 PURCHASE PREDICTION

Given the vector space with all its algebraic properties, we can use the centroid-based customer modeling approach for *purchase prediction*. The target is to find the product group of the next purchase of a customer. Usually one utilizes the previous purchases for modeling the customer. The model can then be used for prediction. In our case, we have chosen a stable customers base with at least 25 purchases each. The data sets in use are data set 2, 3, 4 and 5 in Table 3.3. We model our customer by calculating the centroid of those 25 or more purchased products, except the last five. The last purchase is used as the target label, describing the purchase, we want to predict. The other four unregarded purchases are seen as a gap. The division of purchases $P_c = \{p_i | \, i \in \mathbb{N}, 1 \leq i \leq 25\}$ for an example customer is depicted in Figure 4.6. Each purchase represents one product. This can be the product title or the product vector $\vec{p}$ as defined in Section 4.1. We have chosen such a split to relax the evaluation measures in a later step. This leaves us with a point in the vector space representing the customer and a label representing the next purchase of the customer.

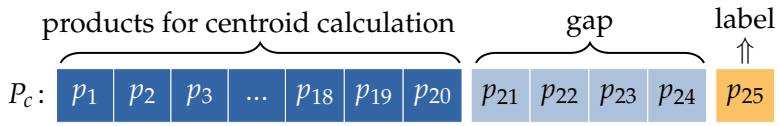
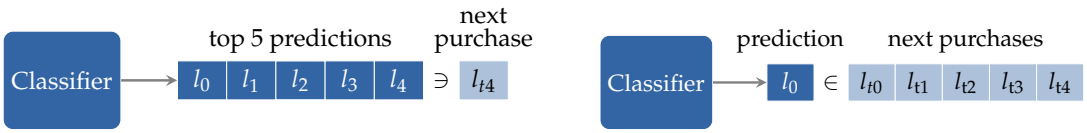


Figure 4.6: Purchase gap



(a) Accuracy@5 **(b)** Accuracy in 5

Figure 4.7: Accuracy measures

Strictly speaking, purchase prediction is not a part of market research, but can be assigned to market prediction. Targeting and personalization make heavy use of predictive behavior modeling. It therefore is in the focus of this thesis.

With the static modeling we explained before, we now can reduce the prediction problem to a classification problem. There are many possibilities for classification in n-dimensional vector spaces. For simplicity we have chosen a simple neural network with one hidden layer with 256 neurons. It takes the customer vectors as inputs and the last purchase as target label. For optimization, the network utilizes stochastic gradient descent to minimize the categorical cross entropy [RK13] [Mur12]. To prevent overfitting, we added a drop-out layer with a 50% rate [SHK+14]. This results in the accuracies found in Table 4.3. All test cases use the 39 possible main product categories. The low accuracy can be explained with the high variation of purchases in the relatively small number of test customers. To loosen the constrains, we used the *accuracy@5* as a subtype of precision@5 [BV00] and the *accuracy in 5*. This is made possible by the gap, we introduced earlier. Accuracy@5 tries to predict if the next purchase category is in the top 5 predictions of the classifier. Accuracy in 5 reverses this concept and tries to predict one of the next five purchase categories of the customer with the best prediction from the classifier. Figure 4.7 gives an overview over both measures. Note that we still use the very last purchase as the target label during training. The purchases from fourth-to-last to second-to-last are ignored whilst training. This prevents the classifier from overfitting to the labels in the gap. For the test sets, we get an increase for the accuracies of 10 to 20 percent points. Refer to Table 4.3 for more details.

The classifier can not give an exact prediction of a user's purchase, but it can still deliver some useful insights. For any customer we calculate two things. The accuracy@5 presents the next categories, where the probability of the customer purchasing is high. We can use this information to offer the user recommendations from these five categories. On the other hand the accuracy

training users	test users	number of purchases per user	accuracy	accuracy@5	accuracy in 5
550	146	>25	53.2%	80.1%	84.2%
1600	400	>30	40.1%	81.2%	69.0%
20,000	4,000	>25	49.8%	78.1%	83.7%
43,000	10,000	>30	44.5%	75.7%	78.7%

Table 4.3: Accuracies for different test sets

in 5 will provide one category, from which the customer will likely purchase in one of the next five purchases. This does not have to be the next purchase, but any of the next five. Imagine a shop scenario. With the top five predictions of the classifier you now can offer your users recommendations from five categories they probably like. Additionally, you have the top category of the classifier, which is likely to be purchased by the customer in the near future. Both of these information can be the bases for an online shop recommender system. But all this shows that single point prediction in this highly fluctuating type of data is not feasible. The suggestion would be to use the status group classification and other rankings for products in the market to make a prediction. This needs further research. We will pick up on this in the outlook in Chapter 6.

Another problem we did not research is the time aspect. Usually purchases are highly time-depended. Through the word vector analysis we have lost any time component except the order in which the purchases occur. We do not know anything about the frequency or intervals of the orders. This makes it hard to infer recurring purchases with methods from time series analysis. Additionally, we had to decouple the time step to the next purchase from the real time between the purchases. This leaves us with no possible to predict when the customer will purchase the next product. A time-dependent model would need a complete new set of algorithms from the domain of time series forecasting.

4.6 SUMMARY

We have presented many different applications of word vector representation for market research. The main point is to show how powerful methods from language analysis can be in areas with simple languages or where the language has a low impact. The product titles of our data resemble exactly this group. The titles are short, but dense, sentences without a predicate or similar language constructs. Still, nearly all proposed methods perform good given the sparsity of the language. The methods are different. Product recommendation, a segmentation for both customer and product and a predictor for next purchases are the main concepts, we have presented. We have shown how flexible and effective NLP can be in all three instances.

The product recommender and the classifier both use the semantic concept algebra and the combined vector space. We have shown that proximity of points can be used to get objects similar to another. Products close to a customer can be used as recommendations. Another case for recommendations is the possibility to add and subtract concepts. The product vector is altered by the algebraic operations and therefore moves to another position in the vector space. A nearest neighbor search reveals proposals for the customer according to the concepts in use. Test samples detail satisfactory product suggestions for both models. A similar approach is used for the segmentation. Here, the algorithm compares cosine distances to words representing concepts. Concepts include gender, marital status and house ownership. The word closest to the customer or product determines the affiliation to one of the concept's labels. This means the algorithm has to check all words masking different characteristic of one concept. For the concept of gender the representatives are the words *men* and *women*. According to the statistics, our concept can model customers' status groups on a global level. This concept can also be used to classify products into groups. These represent those target groups which should be addressed with the product. The last approach is the prediction of upcoming purchase categories for a customer. Despite the low accuracy of the classifier, we managed to use the results to improve the usefulness of our

system. The loosened measures Accuracy@5 and Accuracy in 5 broaden the interpretation for a classification result in such way one can use for a recommender system. Instead of one prediction we return several categories in a ranking. The ranking gives information about how likely a purchase in a category is and how long it probably will take until the purchase in that category. This supports a more flexible scheme for suggestions of categories to the customer.

5 BIOLOGICAL INTERACTION ANALYSIS

As the accuracies of the purchase prediction in Section 4.5 are below 50%, naturally we wanted to improve these scores. One can easily apply hundreds of methods for optimizing vector spaces. A standard method for improving classification in vector space are *kernel tricks*. A kernel trick or *kernel method* is an algorithm to use a linear classifier on non-linear data [HSS08]. The most common kernel tricks will not give us reasonable results on our data, so we wanted to try a non-typical approach for building kernels. For this, we used a method called *Biological Interaction Analysis* (BIA). BIA uses stochastic methods and physical laws to analyses and simulate cellular processes [HE88]. Additionally, co-localization has an important role for finding rules in biological processes [MW03]. At a first glance this does not seam to be related to kernel tricks or market analysis in any way. We will show that BIA in combination with a simulation can transforms the data points in a vector space to make them separable in a higher dimension in Section 5.1. BIA and the simulation can be seen as a kernel. The major difference to common kernels is the *context awareness* of our approach. We learn the parameters for the kernel not only from the data, but also from context information. In our particular case, we will build a kernel for the customers in the vector space for better prediction of their next purchase. We will base our kernel trick not only on the customer vectors themselves, but also on the product vectors as the context of customers. Unlike traditional kernel functions, this gives us a more specialized model for kernel application. We managed to produce a context-aware kernel to enhance our purchase prediction from Section 4.5. We will show that classification in a language-based vector space can be improved by using BIA and simulation of physical laws in a high dimensional space in Section 5.2. We will establish the kernel-like nature of this methodology. As a last measure, we will improve the performance of the simulation in Section 5.3 to make its application more compatible to traditional approaches. We will optimize two different parameters to reduce the runtime whilst keeping the accuracy the same.

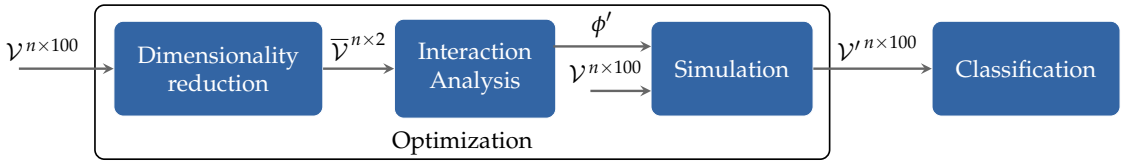


Figure 5.1: Setup for optimizing the classification in $\mathcal{V}$ with BIA and simulation

SETUP

Biological Interaction Analysis (BIA) has not been embedded into the work of kernel methods or market research. Therefore, we need to identify such an embedding into our context. This requires a non-trivial setup for building the data processing pipeline. Our setup has four steps detailed in Figure 5.1. The first step takes the word vector space $\mathcal{V}$ with its 100 dimensions and reduces it to a two dimensional vector space $\overline{\mathcal{V}}$. High dimensionality is a problem for the further proceedings. BIA usually focuses on physical problems in 2D or 3D. The software in use only can model points with up to three dimensions. The word vectors have a length of 100 each. Dimensionality reduction for BIA is needed, on the account of that there could be a loss in information. The second step is the BIA itself. It takes the reduced two dimensional vector space and models a potential ϕ and force function ϕ' given the preliminaries in Section 2.2. This can be used as a set of rules in step tree, the simulation. The simulation will apply the forces to the customers accordingly to the physical laws of motions. This results in acceleration and motion of the customers in the vector space. Contrarily to the Interaction Analysis, the simulation will be run on the 100-dimensional word vector space $\mathcal{V}$. Only the potentials are derived with BIA in the reduced vector space $\overline{\mathcal{V}}$. The simulation produces an altered version of $\mathcal{V}$ called $\mathcal{V}'$. The positions of customers have been changed. The first three steps represent the optimization part of our initial pipeline in Figure 1.1 in Chapter 1. The fourth and last step takes the altered vector space and trains the same classifier used in Section 4.5 on it. The result of the last step then can be compared to our previously obtained accuracies in Table 4.3.

5.1 MODELING

This section shall be used to describe each step in the setup more thoroughly. We will describe the first three steps within our context. These are in chronological order: Dimensionality reduction, Interaction Analysis and Simulation. The last step and its conclusions will be part of Section 5.2.

DIMENSIONALITY REDUCTION

Before starting the interaction potential modeling, there is the problem of dimensionality. As mentioned earlier, the potential can currently only be fitted to points in 2D or 3D[1]. The word vectors have a dimensionality of 100. We have to apply *dimensionality reduction*. Dimensionality reduction embeds a high dimensional vector into a lower dimensional vector space. In doing so, the relative

[1]This is limited by the software in use. A version for arbitrary dimensions is currently in development.

position of the vectors to each other is preserved. We reduce our combined vector space by applying *Principal Component Analysis* (PCA) [Jol11] to the word vector space reducing it to two dimensions. PCA uses covariances and eigenvectors to build a representation of higher dimensional vector spaces in 2D[2]. The general distance information in the vector space are preserved although there is a margin for errors. When one reduces a 100-dimensional space to two-dimensional one there must be some sort of loss. Points in higher space have a finer distance-based comparability given the larger amount of dimensions to compare. The same accuracy cannot be expected from points in two dimensions. Still, we can show that even an analysis in the lower dimensional space is enough to make the simulation feasible. An example for our reduced vector space can be found on the left hand side of Figure 5.4.

INTERACTION ANALYSIS

The first step is to model all parameters of the BIA accordingly to our distribution of customers and products. Our sample data contains 696 customers and ca. 500,000 products representing the combined vector space. This is denoted as data set 2 in Table 3.3. We used the *Mosaic* plug-in [SRS13] for *Fiji* to get a potential curve for the derivation of forces. Fiji is an open source toolbox for biological image and data processing. The Mosaic plug-in uses the concepts from Section 2.2 and is the official implementation of [HPS10]. With it, we can model a function for $\phi(d)$ as given in Equation (2.8) with certain values for the strength ϵ and the length scale σ with the lowest residual. For choosing the shape f the software needs the user to choose a potential function from a list of predefined options. As stated in Section 2.2, we have selected the Plummer potential for modeling $\phi(d)$ and f. From this it automatically follows that the threshold t for interaction is set to 0. For our particular scenario we receive a potential function and a force function like in Figure 5.2. The figure shows that nearby points have a high absolute potential of interaction, whereas distant points will have a lesser interaction potential. This follows directly from the potential curve. Note that we consider the absolute distance to be 0 as the strength of the potential. The potential curve forms the base for the forces used in the simulation. The forces have a peak at around 0.1 because there the potential has its steepest slope. The long tail of the function shows that distant particles will apply little to no force on each other.

SIMULATION

The potential function from the previous section alone does not yield any information for us. We need to use the potentials between particles for reordering the customers in the combined vector space. This is done via simulation. To build a reliable simulation, one needs to integrate the motion of particles into time steps. For our purposes, we used the *Velocity Verlet* algorithm [SABW82] for integrating the particles' motion into time steps. This yields stable simulation results. For simplicity, we set the mass of a particle $m = 1$.[3] So Equation (2.9) can be written as $F(d) = a(d)$. Another simplification is the time equidistance[3] $\Delta t = 1$, because the purchase prediction does not have a direct time component in our scenario. We only want to predict the next purchase and we

[2]Using 3D did not bring any improvement, because the reduction to 3D is as tremendous as a reduction to 2D.

[3]Note that no parameter has any physical unit of measurement because our scenario is an abstract one without any physical properties for the particles.

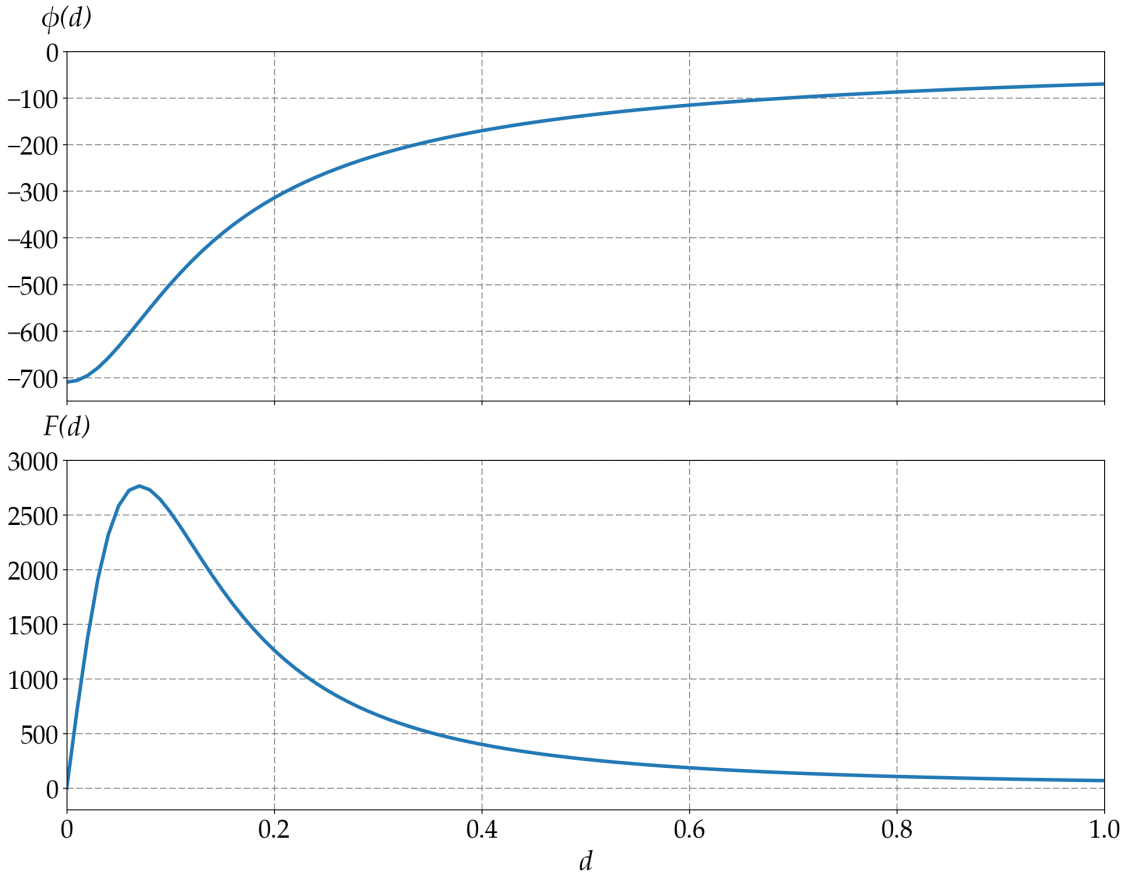


Figure 5.2: Plummer potential $\phi(d)$ and forces $F(d)$ of distance d with $\epsilon = 709.5$ and $\sigma = 0.0987$

do not model the real time between purchases. In our scenario, purchases occur with the same frequency making every purchase in a time line equidistant to the next one. The time step has also no relation to the artificial time in the simulation. The time in the simulation only gives access to the length of vector space optimization. Equation (2.11) from Section 2.2 now reads

$$\vec{x}_{i+1} = \vec{x}_i + \frac{1}{2} \sum_{j=1}^{|N|} \vec{F}(d_j) + \vec{v}_i \tag{5.1}$$

The simulation uses the customers as the primary particles, whereas the products construct the context. This means, the products have static positions and do not move during the simulation. The customers are free to move accordingly to the forces applied to them with Equation (5.1). The forces acting on a customer are the cumulated sums of interactions of all neighboring products of the customer. The neighborhood radius r_t can be chosen arbitrarily. Only products in this radius around the customer are considered in the force calculation. For our test case, we set it to be 1. r_t has a major impact on the runtime of the simulation. If we decrease r_t, we get less neighbors $\bar{n}$ per customer and reduce the number of force calculations. This follows directly from the complexity of the simulation as stated in Section 2.2. A more detailed look onto r_t, accuracy and the performance of the simulation will follow in the next section.

Contrarily to previous methods, the simulation relies on the euclidean distance and not the cosine distance because physical movement is rather expressed by an euclidean vector than an angle.

This abstraction does not degrade our approach but simplifies the calculations. We have used the simulation framework *openFPM* [IL14] for calculating the interaction forces. openFPM allows us to write simulations on a large scale. This gives us access to the potential of simulations required for our purposes.

5.2 EVALUATION

This section is used to present the results of our setup. The focus is on the accuracies of the classification step of Figure 5.1. We will detail the accuracies and put them into context to the results from Section 4.5. Before the comparison, we will establish the kernel-like effect of a simulation with potentials derived with BIA.

BIA AND SIMULATION AS A KERNEL

In mathematical terms, BIA and the simulation encapsulates a *kernel trick*. A kernel trick is defined as an inner product K on the feature map ψ[4] for vectors $\vec{x}$ and $\vec{y}$ in a vector space [HSS08].

$$K(\vec{x}, \vec{y}) = \langle \psi(\vec{x}), \psi(\vec{y}) \rangle \tag{5.2}$$

The most important part is the *feature map* ψ. After deriving it from a given K, a feature map ψ gives a transformation of vector spaces. Usually, K is chosen in a way that the feature map makes the data separable in a higher dimensional space. Examples for kernels are the *polynomial kernel* and the *radial basis function* (RBF) kernel [HSS08]. The mapping between two vector spaces $\mathcal{V}$ and $\mathcal{V}'$ can be described with the function in Equation (5.3).

$$\psi \colon \mathcal{V} \mapsto \mathcal{V}' \tag{5.3}$$

Note that $\mathcal{V}$ and $\mathcal{V}'$ are same as in our setup pipeline in Figure 5.1. $\mathcal{V}'$ comes with a new order of vectors. The new vector space $\mathcal{V}'$ has new properties regarding separability. A simulation can be used as a kernel. The application of Equation (5.1) to all particles in a context generates global movement. The particles and their respective vectors are rearranged. A profound model for a simulation can improve the separability of the vectors with this reordering. A classification on the altered vector space $\mathcal{V}'$ will perform better. To find such a qualitative model, we employed BIA to model a Plummer potential over particles (i.e. customers) and the mesh (i.e. products) as their context. The results of the simulation on our particular data set from market research can be found in the next section.

Our kernel trick has a function fitted to the particles (customers) and their context (products). That is why we call the kernel context-driven or context-aware. This is different compared to traditional kernel tricks which do not access context information. They only take the distribution of particles (i.e. customers) into account. Additionally, traditional kernels only use a minimal amount of parameters. Most of the time, they use the variance σ^2 of the data as a scaling factor. As another possibility, the factor can be trained on the data, but it does not model any context

[4] Standard literature uses ϕ for the feature map, but in our case this is already the potential function.

training users	test users	accuracy in $\mathcal{V}$	accuracy poly kernel	accuracy RBF kernel	accuracy in $\mathcal{V}'$
550	146	51.3%	67.5%	68.4%	64.1%
1600	400	40.1%	17.5%	40.9%	41.7%

Table 5.1: Accuracies for different test sets

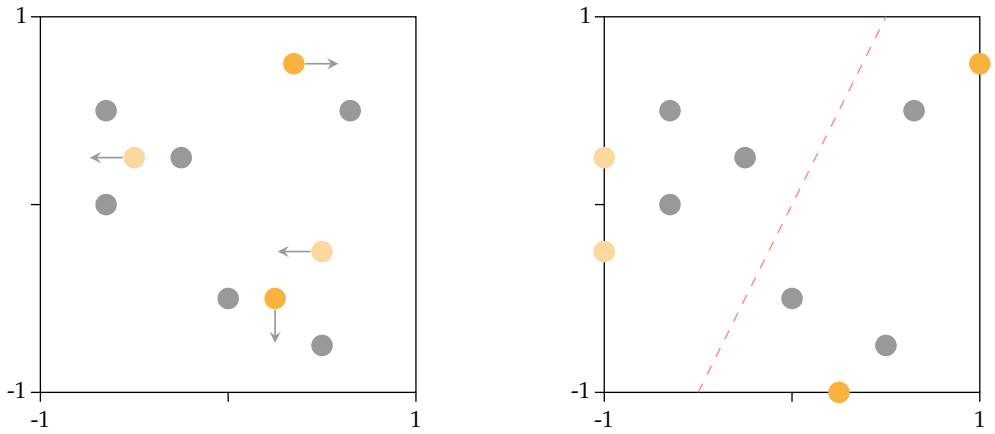
information. Further research is required to formalize BIA and simulation to a kernel K and feature map ψ. This would open the way for context-driven kernel creation. Finding kernels would no longer only rely on the rather static parameters, but could be improved by context information. We will elaborate on this topic in the outlook in Chapter 6.

RESULTS FOR OUR DATA SET

After passing our setup in Figure 5.1, we obtain the accuracies in Table 5.1. The customers in the two data sets are disjunct. Their properties can be found under data set 2 and 3 in Table 3.3. Note that we choose these two smaller data sets because a larger one would have taken to long to simulate. For comparison, we also applied a polynomial kernel (poly) with degree two and a radial basis function kernel (RBF) to the original combined vector space $\mathcal{V}$. Finally, we reran the classification on the kernel-altered vector space $\mathcal{V}'$. $\mathcal{V}$ and $\mathcal{V}'$ are the respective input and output vector spaces from our setup in Figure 5.1. All measures are averaged via a 3-fold cross-validation with an 80-20 split. The original accuracies were obtained in the unaltered vector space $\mathcal{V}$ produced by fasttext as described in Section 4.5. Note that testing the accuracy on the same data set from which we derived the potential is not overfitting because the BIA does not know the labels for the data at any time. The optimization done by BIA and the simulation does only rely on the density distribution of products and customers.

The accuracies from beforehand classification in Table 4.3 were improved by some margin. For the small test set in $\mathcal{V}'$ we can see an increase of 12.8 percent points compared to the results obtained in the unaltered vector space $\mathcal{V}$. The large test set's accuracy is increased by 1.6 percent points. Our kernel trick does not outperform the two standard kernels in the case of the small test set. Both standard kernels have a slightly higher accuracy than our approach. We still can support our general idea of building kernels with context information. Even thought, our approach might not be as good as standard kernels yet, we still can measure a similar improvement of accuracies. This is an indicator that our approach can improve the separability and act as a kernel method. It should be noted that the polynomial kernel is not reliable for the large data set. The low accuracy show that the classification is random. Therefore the kernel does not improve the classification in this case. On the other hand, the RBF kernel improves the classification. Therefore, it is feasible to assume that kernels can bring an improvement to our classification in general. The polynomial kernel is just inappropriate for this one test case.

The increase shows that it is possible to improve vector spaces for classification with BIA and simulation as a kernel method. In our case, the simulation moves all points to the minimum and maximum values of our vector space in each dimension. These are -1 and 1 respectively. This means that the classifier only needs to deal with -1 or 1 in each dimension. It is trivial that thereby the data gets more separable. An explanation in 2D can be found in Figure 5.3. Customers are in shades of orange depending on their class label. Products are in gray. The forces in Figure 5.3a



(a) Not separable **(b)** Separable

Figure 5.3: Example kernel trick in 2D

are marked as arrows. These are induced by the interaction of the products upon the customers as described in Section 5.1. Our approach splits groups of densely positioned customers for better separability and therefore better classification. Imagine two customers who bought similar products and are therefore positioned nearby in the combined vector space. Through their high volume of purchases they are in a cluster with a lot of neighboring products. Nevertheless, their next purchase will not be in the same product category, but the classifier would predict the same category. The proximity of the customer makes them look like they belong to the same category to the classifier. The simulation will now induce a lot of forces onto the customers because of the dense embedding of the surrounding products. The following movement of the customers will separate them to two different borders of the vector space. This is because even small differences in positioning of two particles lead to completely different sums of forces. After that, the classifier can distinguish between the two customers and categorize them into two different categories. If we generalize this assumption to all customers, we can say that the vector space gets more separable.

DRAWBACKS

There are still some downsides to the process. We want to present three disadvantages in this section. One is the decrease in accuracy improvement on the larger data set. The decrease can be explained with the higher information content in the data. More data for the classifier helps to provide a more stable training phase. The classifier can already reduce the loss over more distinctive situations. Therefore kernel trick loses influence on the final result. The improvement through the kernel trick decreases. Another point is the fact that the BIA is made with the assumption of using small data sets. In biological science one usually operates with small amounts of particles. A similar effect can be seen in our test cases. Smaller test set experience a better accuracy improvement through BIA and simulation.

Another discrepancy producing loss is the dimensionality reduction. We assume that a potential derived in 2D cannot model a vector space in 100 dimensions completely. Our assumption is that the reduction generates to much loss and the vector space has a dissimilar form in 2D. It is

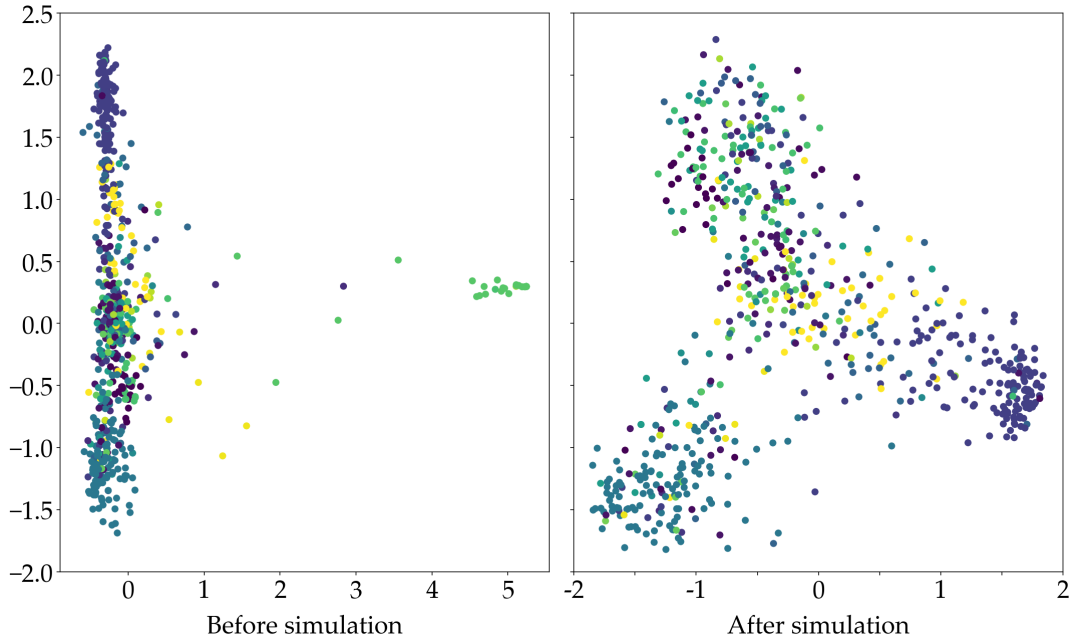


Figure 5.4: Effect of simulation on positioning of customers in vector space.

necessary to model the BIA in 100 dimensions and derive a potential function accordingly. Since Fiji does not support more than three dimensions, this is subject to future research. For example, Figure 5.4 shows the smaller test set before and after the simulation. The colors of the points are the respective product categories of the next purchase. It is clearly visible that the right vector space is much easier separable than the left one. For picturing the points in the vector space on paper, we had to reduce the vector space. This is also done via PCA. Note that we are using PCA in the same context compared to the PCA used in the dimensionality reduction step. The vector space $\overline{\mathcal{V}}$ is the same as the depicted vector space on the left in Figure 5.4. Nevertheless, this means that we lose separability in the illustration as we would in the modeling with BIA. Some separability is lost due to the conversion from 100 to two dimensions as explained in Section 5.1. Still, the right hand site of the figure with $\mathcal{V}'$ has a better separability, but this is after the simulation. BIA is modeled before the simulation, so on the vector space on the left. It is trivial that this vector space $\overline{\mathcal{V}}$ has a different density distribution than the original vector space $\mathcal{V}$. BIA will slightly divert the potential compared to a potential derived in 100 dimensions. Additionally, some clusters are still separable from each other in $\mathcal{V}'$ even if it does not look like it in the 2D representation of $\mathcal{V}'$. For example, the central cluster with the yellow points in the vector space after simulating can still be separated from the others in 100 dimensions. In Figure 5.4 it looks like there is to much overlap with other clusters for a good separation. Additionally, the potentials were derived from the small test set and then applied to the large test sets. This leads to the problem that we use the same Plummer potential function over both data sets. The potential could be malformed for the large data set and hence give poor results. One could derive the potential from the large data set for modeling it. This can be very time and memory consuming.

The calculation time for our kernel is another serious problem. Running the BIA and the simulation takes 20 minutes for our small test set with 696 customers. This increases significantly for more data samples. Compare the measures in Figure 5.5. Note that the runtime for BIA is much smaller in all cases compared to the runtime of the simulation. The x-axis is logarithmic. The slow performance is quite clearly the problem of the high complexity $\mathcal{O}(p \cdot \bar{n})$ of the simulation. In the

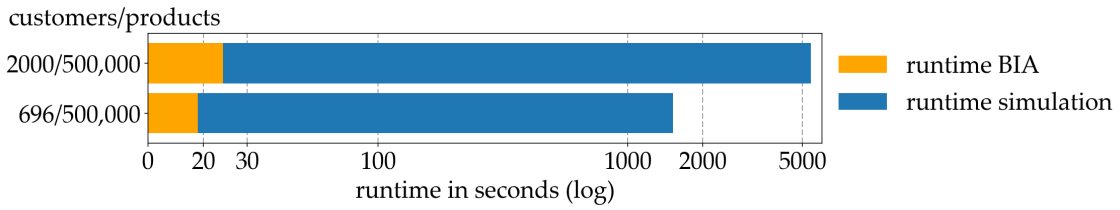


Figure 5.5: Runtime of BIA and simulation for two sets of customers with the same set of products

case of the small data set p is 696, the number of customers (particles), and $\bar{n}$ is approximately 20,000, the average number of products near a customer. We used ca. 500,000 products in total. We think, there is the possibility to reduce the calculating time. The first step would be to improve the simulation to run more efficiently. We will discuss this in the next section thoroughly. A second idea is the mapping of the movement of the particles in the simulation to an approximated feature map ψ_{sim}. We will go further into this topic in Chapter 6.

5.3 SIMULATION PERFORMANCE ENHANCEMENT

In this section we want to tackle the problem of performance issues of the simulation without decreasing the improvement of the classification. Additionally to the products of test set 2, we use test set 6 from Table 3.3. This set test contains 910 customers and has a subset of 37,000 out of 500,000 products. Contrarily to test set 2, these products are only the ones the customers have purchased. The 500,000 products of test set 2 contain all products which were purchased by any customer of all of our test sets. They are therefore modeling our complete test market. The 910 customers are disjunct to the ones in test set 2. We will address both accuracy and runtime with these data sets.

First, we want to take a look at the impact of the number of products in the simulation. The complexity of the simulation $\mathcal{O}(p \cdot \bar{n})$ is given with p as the number of customers and $\bar{n}$ as the average number of neighboring products per customer. The number of neighbors is directly controlled by the number of products, used as the context, in the simulation. More products in the simulation automatically increase the density of products around a customer. There will be more interaction between these products and the customer. This leads to more force calculations per customer and a longer simulation runtime. Our assumption is that it is sufficient to only use products purchased by customers. This would reduce the number of products drastically. In our example from 500,000 to 37,000 for ca. 900 customers. This leads to a shorter runtime compared to the larger test set with additional products. We have to show that with the reduction of products the simulation still improves the accuracy of the classifier. For all experiments the neighborhood radius was set to be 1.0.

Figure 5.6 details the runtimes and accuracies for both test sets. The simulation runs much faster and even the BIA has a better runtime. The total improvement is 1200 seconds with nearly the same accuracy improvement of 6.7%. The absolute values of the accuracies for both test sets are 63.7% for the larger test set and 63.1% for the smaller test set after simulating. The base line accuracy for both cases is 56.4%. The simulation with 500,000 products only delivers a slightly increased accuracy improvement of 7.3% compared to a simulation with 37,000 products. On the

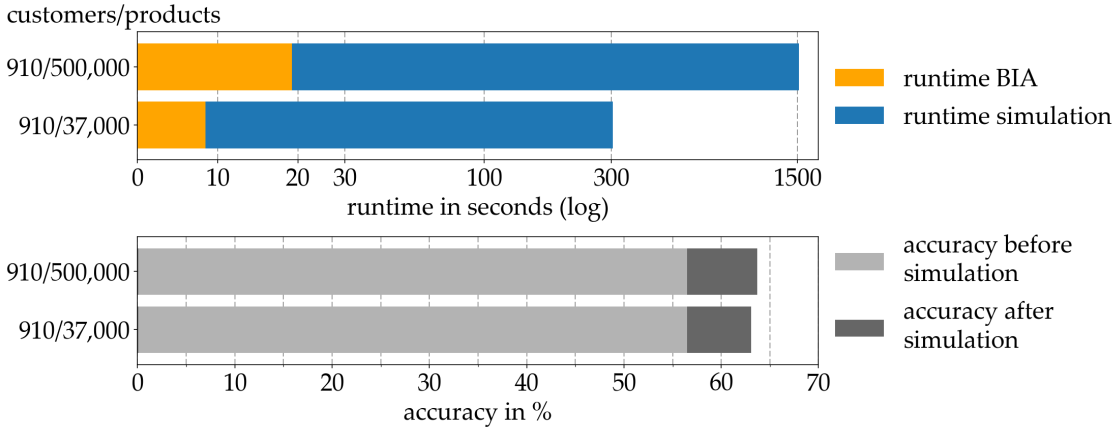


Figure 5.6: Runtime/accuracy of BIA and simulation for the same set of customers with two sets of products

other hand, the runtime of the large test set is much slower. We think, a 0.6% accuracy gain with a 500% increase in runtime is not practical. This shows the sufficiency to only use products which were purchased by the customers other than using all possible products. The stabilization of the simulation with the complete test market does not yield a lot of improvement. It only makes the simulation run slower. The products of the complete market do not add any more value to the positioning of the customer. The natural thing to assume would be that products which are not purchased by the customers still have an impact on them through advertisement or the like. The simulation depicts the opposite. The purchase prediction has the nearly the same accuracy for just the products purchased by the customers. Apparently, for our model the sheer presence of products does not help the classification. The customer has to buy it to make an impact on the next purchase. The advantage of all this is the reduced number of products and therefore the reduced runtime of the simulation.

The second parameter, we want to take a look on, is the neighborhood radius r_t. It has a severe impact on performance of the simulation. The complexity of the simulation is $\mathcal{O}(p \cdot \bar{n})$. It is trivial that r_t directly controls the average number of neighbors per particle as stated in Section 2.2. A smaller radius decreases the number of possible particles to be neighbors. A larger radius encapsulates more neighborhood particles. Since the complexity is a multiplication the runtime will decrease or increase directly proportional to $\bar{n}$ which is proportional to r_t. To find the range for r_t, we have to look at the minimum and maximum value of it. The minimum value is obviously 0, but the range of r_t does not include zero. A radius of 0 would render $\bar{n} = 0$ and therefore no interaction would take place. If the number of neighbors is zero, there are no forces to be invoked on the particle. It does not move in the simulation space. So, the vector space will be the same after simulating as it was before. It is clearly visible that no optimization would take place. The maximum value for r_t between two vectors $\vec{x}$ and $\vec{y}$ is given in Equation (5.4). Its main dependence are the maximum values -1 and 1 for each of the 100 dimensions in the combined vector space. These maximum values are given by the domain $[-1, 1]$ of the vector space as explained in Section 2.1.

$$\max(r_t) = \sqrt{(x_0 - y_0)^2 + ... + (x_{99} - y_{99})^2} = \sqrt{100 \cdot (1 - (-1))^2} = 20 \tag{5.4}$$

The maximum euclidean distance between two vectors is 20. The total range for r_t is therefore $(0, 20]$. We will not use the complete interval for performance measurements. We will choose

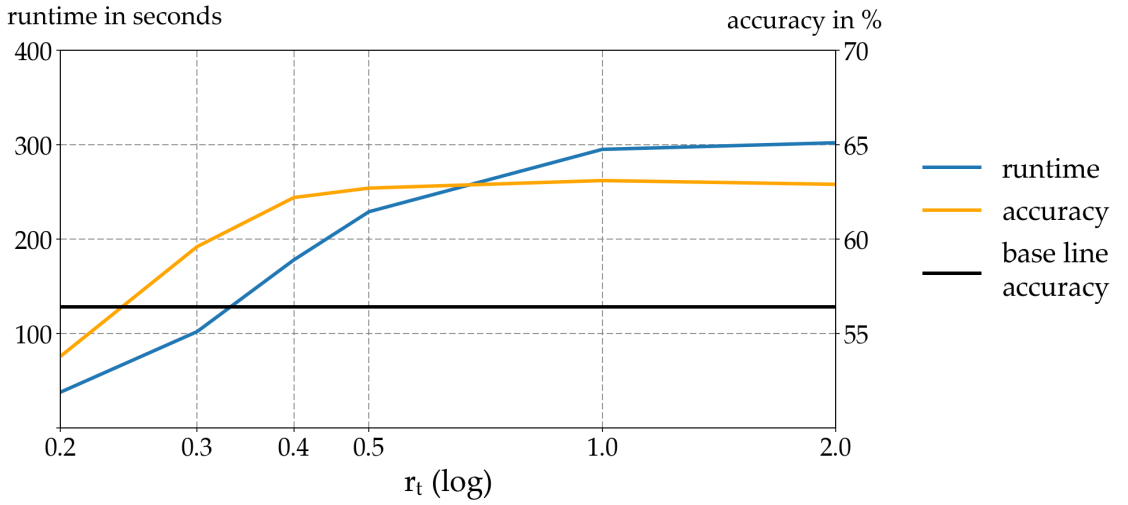


Figure 5.7: Runtime of the simulation and accuracy of the classification for different r_t

different, meaningful values for r_t from the complete range to show the main points of our argumentation. We are still using the 910/37,000 test set 6. The base line accuracy for classification without simulating is 56.4%. If the accuracy of the classification after simulating falls below this threshold, the simulation has no improving impact any more.

Figure 5.7 details all the runtime measurements in relation to the measured accuracies. The x-axis is logarithmic and the y-axis details time on left side and accuracy on the right side. With smaller values for r_t the runtime of the simulation and the accuracy of the classification decreases. The accuracies fall below the threshold for the improving impact at $r_t < 0.3$. The measurements show that r_t should be larger than 0.3 for good accuracy, but less or equal to 0.5 for reasonable runtime. There is no improvement for values higher than 1.0. They have a similar runtime without improving the accuracies. The data is very dense and most customers have a distance of 1.0 or less to their neighboring products. Going up to 2.0 and further only adds few important points to the interaction. The forces in the simulation are nearly the same as they would be for $r_t = 1.0$. This leads to similar accuracies and runtimes. Nevertheless, even on a small data set a radius $r_t < 1.0$ results in a long runtime compared to the standard kernels which both run in about three seconds on the test set. We will propose a solution to reduce the runtime of our approach in Chapter 6.

5.4 SUMMARY

The main conclusion we can draw is that it is beneficial to use novel methods for market research which seem far off the traditional spectrum. This aligns with the discoveries in Chapter 4 where we proposed a similar thought for word vector representations in market research. We have successfully transferred our hypothesis about improving classification accuracies with BIA to a market research context. The introduced setup delivers feasible results on our data sets. We have shown the impact of BIA by depicting its improvement to the accuracies for purchase prediction measured in Section 4.5. We presented the kernel trick which reorders the vectors to make them separable. However, the kernel does not to give any accessible information. After the simulation we cannot use the vector space $\mathcal{V}'$ for the applications described in Chapter 4 anymore. Our

suggestion is to use the unaltered vector space $\mathcal{V}$ for all language related analysis and only use the altered vector space $\mathcal{V}'$ for classification tasks or the like.

In Section 5.3 we have shown the influence of different parameters to the runtime of the setup. We measured runtime and accuracies for different amounts of context particles, i.e. product vectors, and neighborhood radius r_t. Our conclusion was that products which are not bought by the customers do not improve the simulation. This is contra intuitive to the concept of ads. Usually, a customer is influenced by marketing, but our results says that is not the case. One explanation could be that our data is missing information about marketing. We do not know how the customer was influenced by ads and recommender systems before the purchase. Our data only details purchases, so we only can make assumptions about their influence. All other aspects of the market are hidden to us. Additionally, our experiments showed an optimal neighborhood radius of $r_t = 0.5$ which also increased the performance of the simulation.

A general problem is the modeling of real world time. Just like the vector space representation, the simulation cannot remodel the time between purchases. This means we cannot predict when the customers is going to buy the next product. For high frequency buyers this should not be a problem. The amount of products bought by such a type of customer has a regularly shopping scheme. So, we assumed that the time steps between purchases for one customer are equally distributed. Another major problem is the runtime of the simulation. Even with our improvement the simulation is much slower than traditional kernels. We will suggest some ways to improve the performance even more in the next chapter. The connection of NLP and BIA brought up some interesting discoveries. The proposed concepts in this chapter of the thesis bring up a lot of ideas for further research. They will be explored in Chapter 6.

6 SUMMARY AND OUTLOOK

The thesis has proposed a new way for a synergistic combination of market research with other domains to answer important questions. These questions are closely aligned to the modeling of customers and products. Especially user segmentation and recommender systems can be improved with methods which are not in direct correlation with them. We have shown this for two concepts: Natural Language Processing (NLP) and Biological Interaction Analysis (BIA). We can now fill the pipeline from Figure 1.1 with more detailed steps. The pipeline in Figure 6.1 is now filled with the concepts and algorithms in use for each step. One can clearly recognize the general path of thought as detailed in Figure 1.1 in Chapter 1. We matched all proposed ideas with concepts and applied them to the data. The customer and product segmentation were done with concept vectors. Two contrary words describe a concept like having children or not. The largest similarity of a customer or a product to one of the words categorizes it accordingly. Each class of a category is represented by a single word. The recommender system has two possible algorithms. The nearest neighbor approach looks for the nearest products for a customer. After filtering the products the customer already bought, we return the ranked list of item suggestions. Additionally, we used the concept vectors to find matching recommendations to an item already bought. For this, we subtracted the concept of the product and added the concept of the desired target product group. In this case, concepts are product types, like dresses or sandals. The last approach uses the customer and product vectors to predict the next purchase of customer with a simple classification task. We then moved on to improve the classification results by applying BIA and simulation. We have presented the kernel-like nature of this algorithm and its context awareness. This optimization step encapsulates the process detailed in Figure 5.1. It can improve the purchase prediction for small sets of customers. All three word vector approaches and the BIA and simulation were tested on data from a real world scenario. The test sets are closely related to market research. So, we could test our algorithms in a scenario where they might be applied in the future.

We think that our approaches are profitable for market research. The three algorithms produced with word vector representation tie in seamlessly into a market research context. For example, the recommender system based on concept vectors from Section 4.2 can be used in an online shop. The same applies to the purchase prediction with the top five product category recommendations as detailed in Section 4.5. The established link between the applications of NLP and BIA helps

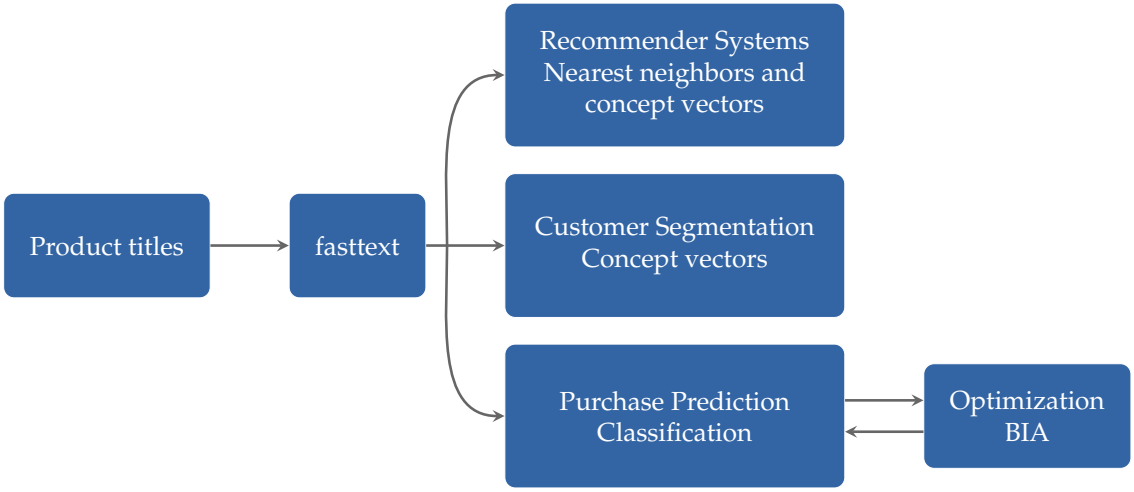


Figure 6.1: Pipeline for data processing done in this thesis

us to consolidate or hypothesis about improving market research with new approaches. Market research has a high demand for innovative techniques. There are only a few areas with a similar high dynamic. This thesis can deliver advances in both, the traditional group-based customer analysis and the individual-based customer analysis. These are two major points of market research as stated in Chapter 1. An example for the group-based approach is the customer segmentation with concept vectors in Section 4.3. It enables the categorizing into user groups via language analysis. The purchase prediction and the product recommendation are individual-based approaches. We like to point out that our concepts cannot model individual shopping missions, yet. We are positive that we can use our approaches for more individual-based contexts, maybe even shopping mission. This still requires a lot of research. The novelty of the approaches leads to a lot of interesting research demand in general. We will present some of these points in this chapter. This list of ongoing research is not complete. We want to highlight the most important possibilities for research.

KERNEL LAYOUTS

The most important point we want to address, is the context-driven feature mapping for kernels as proposed in Section 5.2. We think if a mathematical combination of BIA and simulation can be done, a new way for finding kernel functions would be possible. The potential function from BIA and the movements in the vector space for the simulation can be merged into one function. This function can be used as a feature map ψ for a kernel K. This would make finding meaningful kernels easier. Contrarily to traditional kernels which do not apply context information, context-driven kernels would fill this gap. Kernels could be fitted directly onto the data and context at hand giving them an advantage for delivering insights. As noted in Section 5.2, the feature map ψ_{sim} should also reduce the execution times for applying the kernel to the data. A fitted kernel would be trained once which would take some time. The speedup comes with the application of the derived function to existing or new samples. Using the function will be much faster than running the simulation on each sample. Overall, a kernel learned from the data and context will most likely perform better than a kernel using only the data. A context-driven kernel can employ additional knowledge about the data. We assume that a kernel derived from BIA and simulation

could outperform the traditional approaches. Comparison of our approach to the traditional approach as shown in Section 5.2 is needed to prove our assumption. This requires testing over data sets from several other domains to show the generality of the approach.

POTENTIAL FUNCTIONS

Another interesting point is the modeling of potential functions. As mentioned before, potentials have to be calculated in a vector space with lower dimensionality. This is done via dimensional reduction. We have chosen PCA, but there are other options available, like *t-Distributed Stochastic Neighbor Embedding* (t-SNE) [MH08]. The reduction has a drastic effect on the data. It is trivial to assume that 100 dimensions contain much more information than two. To avoid such loss, it is necessary to expand the current concept of BIA to arbitrary sizes for vector spaces. Ignoring the increased time complexity of the algorithm, this is a simple task. The BIA algorithm only works with distances for the potential estimation. Distances are always one-dimensional for all euclidean or cosine vector spaces. This makes it fairly easy to expand the current two-dimensional case to a 100-dimensional one by allowing vectors with a length larger than two. An expansion to the Mosaic plug-in for handling points with arbitrary dimensionality is currently in development.

Dimensionality is not the only influence. BIA for a special scenario heavily relies on the chosen potential function in Equation (2.6). Finding a general assumption for how objects interact in the scenario can help improving the results. Currently, we do not have a specific potential function for market research. BIA provides a potential which is non-parametric. It can be fitted to a distribution of particles without prior knowledge about their potential function. The estimator calculates all parameters in Equation (2.6) just from the density distribution. The non-parametric potential, in combination with the proposals done in the previous chapter, could improve the classification even more. A test of other provided potentials would be possible. All available potentials in the Mosaic plug-in are detailed in the appendix Section B.

SIMULATED LANGUAGES

A subject, which was not mentioned in this thesis, is the possibility of simulating the word vectors. We could run the simulation only on word vectors and get some results regarding the use of language. This might help to stabilize the word vector space before calculating the centroids of the products and customers. BIA and simulation would also provide information on how words interact in the semantic space. This could include interaction with context and synonyms of words. The context is the general concept of fasttext, so it could yield information about how fasttext sees context and why fasttext actually performs very well. The reasoning behind the good performance of word2vec and fasttext is still subject to research [GL14]. The interaction between synonyms can deliver notes on the quality of the vigor of a word. A word with lot of interaction with close by words leads to the assumption of a high expressiveness. It could mean that the word is very old and has a strong position inside the language. It has been used for a long time, so its meaning has a stable usage in the language of most speakers. New words usually need some time to build a stable foundation for their usage. Neologisms would have a low interaction potential because they do not have a dense neighborhood of synonyms and context, yet. They

need to adjust themselves to the language. The proposals could give a more detailed look on how language evolution works. So, we propose a more linguistic analysis of languages with BIA to support ongoing research about semantic and syntax of languages.

ADDITIONAL DIMENSIONS

Lastly, we think that using additional information about the products can enhance the purchase prediction. Products can be characterized by much more than their product titles. In classical approaches for classification, products are described by physical properties in most cases, really. It is trivial to expand the 100 arbitrary product vector dimensions by characteristic dimensions. These properties could be anything one can use to describe the product, like size, color, number of pages and so on. The classifier would get more information about the preferences of a customer via the properties of purchased products. There should be an improvement in the accuracy because the algorithm can model interests by combining properties into a prediction. This would also open the way to model interests and demands of a customer which could not be correlated from the product titles. A customer buying products with a particular set of properties can show a general interest in a combination of properties. Somebody who buys different products, but from the same brand, will most likely be loyal to this brand in his future purchase. The brand can be seen as his personal interest. We think, this would lead to a better understanding of purchase decisions and shopping missions.

7 APPENDIX

A PRODUCT CATEGORIES

- accessories
- apparel
- auto
- baby
- beauty care
- bills
- book
- business services
- donations
- education
- electronics:
  - audio and video
  - cameras and imaging
  - other
- entertainment
- events:
  - registration
  - tickets
- fees, charges, taxes
- food and drink:
  - alcohol
  - groceries, specialty foods
  - restaurants
- footwear
- general merchandise
- gifts:
  - gift cards and vouchers
  - other
- health care:
  - prescriptions
  - vitamins and supplements
  - other
- hobbies
- household goods:
  - appliances
  - cleaning and paper products
  - furniture
  - housewares
- insurance
- investments, banking
- magazines, journals, newspapers
- memberships
- no category
- office supplies
- payment processor
- personal care
- pet care
- service
- software
- sporting goods
- telecommunications

- tobacco and tobacco alternatives
- toys
- travel:
  - airline
  - auto rental
  - cruise
  - lodging
  - transit
- video games

B POTENTIAL FUNCTIONS

All potential functions have the same equation for $\phi(d)$ with distance d, strength ϵ, shape f, threshold t and length scale σ.

$$\phi(d) = \epsilon \cdot f\left(\frac{d-t}{\sigma}\right)$$

The distinguishable feature of the functions is the different shape f for each of them. Examples for f for different parametric potentials [HPS10] are shown below.

$$\text{Step function:} \qquad f(z) = 1$$

$$\text{Linear potential, type 1:} \qquad f(z) = \begin{cases} 0 & \text{if } z > 1, \\ -(z-1) & \text{else.} \end{cases}$$

$$\text{Linear potential, type 2:} \qquad f(z) = \begin{cases} 0 & \text{if } z > 1, \\ -1 & \text{if } z < 0, \\ -(z-1) & \text{else.} \end{cases}$$

$$\text{Plummer potential:} \qquad f(z) = \begin{cases} -(z^2+1)^{-0.5} & \text{if } z > 0, \\ -1 & \text{else.} \end{cases}$$

$$\text{Hermquist potential:} \qquad f(z) = \begin{cases} -(z+1)^{-1} & \text{if } z > 0, \\ -(1-z) & \text{else.} \end{cases}$$

DEFINITION INDEX

BIBLIOGRAPHY

[BGJM16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[BV00] Chris Buckley and Ellen M Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40. ACM, 2000.

[Cho16] Francois Chollet. How convolutional neural networks see the world. `https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html`, 2016. Accessed: 2017-02-12.

[Das15] Sudeep Das. Making meaningful restaurant recommendations at opentable. `https://de.slideshare.net/SudeepDasPhD/recsys-2015-making-meaningful-restaurant-recommendations-at-opentable`, 2015. Accessed 02.06.2017.

[DSG14] Cícero Nogueira Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78, 2014.

[EKS+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. volume 96, pages 226–231, 1996.

[FG90] C.T. Fitz-Gibbon. *Performance Indicators*. BERA Dialogues Series. Multilingual Matters, 1990.

[Fis01] Gerhard Fischer. User modeling in human–computer interaction. *User modeling and user-adapted interaction*, 11(1):65–86, 2001.

[GBB11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

[GL14] Yoav Goldberg and Omer Levy. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[Har54] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[Haw04] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[HCDK17] Gerhard Hausruckinger, Alessandra Cama, Christian Diedrich, and David Krajicek. GfK Annual Report 2016. `http://annual-report.gfk.com`, 2017. Accessed 31.07.2017.

[HE88] Roger W Hockney and James W Eastwood. *Computer simulation using particles*. crc Press, 1988.

[HPS10] Jo A. Helmuth, Grégory Paul, and Ivo F. Sbalzarini. Beyond co-localization: inferring spatial interactions between sub-cellular structures from microscopy images. *BMC Bioinformatics*, 11(1):372, 2010.

[HSS08] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[HWH+16] Gerhard Hausruckinger, Bernhard Wolf, Matthias Hartmann, Christian Diedrich, David Krajicek, and Alessandra Cama. GfK Capital Market Day 2016. `http://www.gfk.com/investors/capital-market-day`, 2016. Accessed 31.07.2017.

[IL14] Pietro Incardona and Antonio Leo. openFPM - Open framework for particles and mesh simulations. `http://openfpm.mpi-cbg.de`, 2014. Accessed 31.07.2017.

[Jol11] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[MH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[ML13] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM, 2013.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[Mur12] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[MVA93] E. M. M. MANDERS, F. J. VERBEEK, and J. A. ATEN. Measurement of co-localization of objects in dual-colour confocal images. *Journal of Microscopy*, 169(3):375–382, 1993.

[MW03] Jesper Moller and Rasmus Plenge Waagepetersen. *Statistical inference and simulation for spatial point processes*. CRC Press, 2003.

[PL99] Alex Pentland and Andrew Liu. Modeling and prediction of human behavior. *Neural Computation*, 11(1):229–242, 1999.

[Plu11] Henry Crozier Plummer. On the problem of distribution in globular star clusters. *Monthly notices of the royal astronomical society*, 71:460–470, 1911.

[RK13] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[Roz82] Yu. A. Rozanov. *Markov Random Fields*, pages 55–102. Springer New York, New York, NY, 1982.

[SABW82] William C Swope, Hans C Andersen, Peter H Berens, and Kent R Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.

[SB88] Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24:513–523, 1988.

[SHK+14] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[Sny05] Jan Snyman. *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, volume 97. Springer Science & Business Media, 2005.

[SRS13] Arun Shivanandan, Aleksandra Radenovic, and Ivo F. Sbalzarini. MosaicIA: an ImageJ/Fiji plugin for spatial pattern and interaction analysis". *BMC Bioinformatics*, 14(1):349, Dec 2013.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

[TFLW99] C Reid Turner, Alfonso Fuggetta, Luigi Lavazza, and Alexander L Wolf. A conceptual basis for feature engineering. *Journal of Systems and Software*, 49(1):3–15, 1999.

[Wei02] E.W. Weisstein. *CRC Concise Encyclopedia of Mathematics, Second Edition*. CRC Press, 2002.

[ZZO07] Vadim Zinchuk, Olga Zinchuk, and Teruhiko Okada. Quantitative colocalization analysis of multicolor confocal immunofluorescence microscopy images: pushing pixels to explore biological phenomena. *Acta histochemica et cytochemica*, 40(4):101–111, 2007.