Alma Mater Studiorum
Università di Bologna

DOTTORATO DI RICERCA IN
SCIENZE STATISTICHE

CICLO XXXII

Settore Concorsuale: 13/D2
Settore Scientifico Disciplinare: SECS-S/03

Text Based Pricing Modelling

An application to the fashion industry

Presentata da: Federico Crescenzi

Coordinatrice Dottorato:
Prof.ssa Alessandra Luati

Supervisore:
Prof.ssa Marzia Freo

Esame Finale Anno 2020

ABSTRACT

In this study we propose a class of hedonic regression models to predict prices of fashion products using attributes obtained *featurizing* text. Using the internet as a source of data, we developed web-scrapers to collect data on prices and product descriptions of items sold in the websites of five famous fashion retailers and producers. For a set of scraped items, given the pair (price, description) our goal is to estimate hedonic regression models by leveraging the information about the product contained in the description. After each description is mapped to a point in a high-dimensional vector space, our estimation strategy uses sparse modelling, as well as text mining techniques of dimensionality reduction and topic modelling to find the model with the best out-of-sample predictive performance. We refer to this approach as Hedonic Text-Regression modelling. With this approach, we estimate the implicit price of words that are used in descriptions. To the best of our knowledge no previous work has been conducted in the Fashion industry. Empirically, the proposed models outperform the traditional hedonic pricing models in terms of predictive accuracy while performing also consistent variable selection.

ACKNOWLEDGEMENTS

This **PhD project** has been made possible thanks to the **investments** made by **Regione Emilia-Romagna** in the three year plan: *Improving research and technology transfer between firms and universities operating in the same territory.* The commercial parter of this project is Imperial Fashion s.p.a.

The project funded by the region was hinged on **creating value for enterprises and society through the analysis of Big data**. In particular the project of this work is focused on **Statistical Methodologies for sparse data**.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER

1

INTRODUCTION

In the last years, the expansion of e-commerce and online markets has acquired a great importance in consumers' purchasing behaviour. More and more products are being sold online and consumers can buy directly from producers' websites who are partially substituting to local retailers. This expansion offers researchers the opportunity to have at disposal high-dimensional collections of data on different collections of products. Therefore, it opens up the possibility to a much wider range of statistical methodologies for data analysis. As pointed out by Einav and Levin [2014], the emergence of Big Data requires researchers in economic statistics and empirical economics to develop new data management and data collection skills. To give some examples on how much interest is growing on the analysis of web data, the biggest community powered shopping application in Japan, Mercari, launched in 2017 a Kaggle competition to develop pricing algorithms to automatically suggest product prices to sellers based on textual description. The competition rewarded the best three pricing algorithms[1] with a monetary prize of 60k, 30k, and 10k dollars respectively. Einav et al. [2013] and Einav et al. [2011] collaborate with eBay to study internet pricing and sales strategies. The researchers of the Billion Prices Project at the Massachussets Institute of Technology coordinate with retailers

[1] in terms of Root Mean Squared Logarithmic Error

to produce daily prices indexes that closely resemble the Consumer Price Index provided by the Bureau of Labour Statistics [Cavallo, 2018, 2017].

On such premises, the aim of this study is to estimate pricing models of italian fashion products by web-scraping the e-commerce websites of several fashion brands, being both producers and retailers. As these models regress the price of one good on a function of its attributes, they may be interpreted as hedonic models[2] [Feenstra and Shapiro, 2007]. Therefore, a hedonic model requires a collection of control variables for measurable and observable attributes.

Apparel items sold in online stores are always equipped with a textual description providing many details on attributes like materials, finishes, comfort and design which are supposed to drive consumer preferences. Moreover, products descriptions provide details either on easily observable attributes (i.e. color, material) as well as on other attributes that are more difficult to measure and consequently to take into account in a model. Therefore, the main objective of this study is to understand how one can use product descriptions to set up a hedonic pricing model of fashion products. In turn, this means to give a hedonic value to the description by estimating the implicit price of the most relevant words therein contained. To give an example, consider the following description:

> *A dappled print invades this lightweight blouse in silk Georgette fabric. A style with a young, wild spirit that will sublimely complete casual looks with jeans or tailored trousers*

The resulting hedonic model should estimate both the marginal price of materials, in this case silk, and the effect of details about the weight (*lightweight*) and the design (*young*, *wild*). Out of this framework, the effect of these attributes concerning the desing would probably have been dropped from the analysis and absorbed into an error term.

In recent statistical applications, it is common to refer to these sources of data as *unstructured* data sources. This is a way to say that this kind of data do not naturally come in the form of a matrix and need a specific algebraic model to be used for statistical applications [George et al., 2016, Gupta and George, 2016]. Given that, our modelling strategy starts by *featurizing* text, which means to text-mine product descriptions to embed them into a feature or vector space. Once that text have been featurized, the set of features

[2]The appelative hedonic was first introduced by Court [1939]

can be used as attributes for Hedonic Pricing Modelling. Therefore, we refer to the resulting model as a Hedonic Text Regression Pricing Model.

From a methodological point of view, the main challenge that is posed by text data is high-dimensionality. Text data is intrinsically high-dimensional. The number of features that can be extracted from text - and so the set of covariates - tends to increase with the sample size and, depending on the rate, can become even greater. In this framework, traditional Ordinary Least Squares estimation - if feasible - is likely to overfit the data and give bad predictions of prices. Additionally, not all the features that one can extract from text need to be significant determinant of prices. Some words can simply add nothing to prices (think of the word *invades* in the previous example). Thus, some form of variable selection or dimensionality reduction is needed to identify the most appropriate set of attributes. Regarding variable selection, traditional approaches like best subset selection are not applicable or too expensive from a computational point of view. In the last years, sparse modeling has become a very popular tool to improve predictive accuracy [Friedman et al., 2001]. These procedures are based on solving convex optimization problems that encourage the solution to be sparse. Regarding dimensionality reduction, we leverage text-mining techniques for *unsupervised* dimensionality reduction and topic modelling to obtain a new set of covariates. These techniques are unsupervised in the sense that they not consider the presence of additional variables other than text variables. Consequently, we propose Partial Least Squares as a way to derive supervised topics.

Concerning model selection and evaluation, traditional hedonic regression modelling relies on goodness of fit to select the best functional form of the model [Cassel and Mendelsohn, 1985]. In this work, we use some results from statistical learning literature to show that this measure is not the most appropriate for model selection. Thus, model selection is performed by selecting the model that achieves the lowest prediction error for the price of new observations based on their set of characteristics.

A similar approach to that presented in this study has been proposed by Nowak and Smith [2017] in real estate. The authors show that text, when used in combination with a traditional set of *structured* covariates, leads to a better predictive accuracy. In this work, we do not assume to have at disposal any structured auxiliary information other than text, which puts ourselves in a less favourable condition. Also, our attempt to develop hedonic

models leveraging topic modelling and aggregated predictors was previously unattempted in this field.

The remainder of this thesis is organized as follows. Chapter 2 focuses on the usage of text for regression purposes. After a review of the related studies in this field, the chapter introduces the theory of hedonic models and continues with the major challenges posed by text data in this context. Also, the chapter gives an introduction to web scraping and the preliminary operations to process text. Chapter 3 illustrates the most common methods in text mining as well as a review of statistical estimators for regression models with high-dimensional data. The data analysis is presented in Chapter 4. The analysis is repeated separately for two categories of products, dresses and trousers. Also, the role of the brand in the proposed hedonic model is discussed. Eventually, we report our conclusion and directions for further research in Chapter 5.

CHAPTER

2

TEXT AS DATA

Text Mining is the a process of automatic extraction of information from text. Depending on the community of the practitioner, text mining is also referred to as *Text Analytics* or *Statistical Natural Language Processing*. These methodologies have become popular over the last years mostly because of the increasing availability of this type of data and, consequently, of the need of specific analysis. Web-pages, electronic news, digital libraries and social media are some examples of data sources that contain lots of text data.

In this study, we make use of text data in the form of descriptions accompanying products found in e-commerce website to build several predictive pricing models. For this reason, this chapter begins with a review of previous studies on the usage of text data for predicting continuous outcomes - that is, *text-regression*. Afterwards, we offer a brief overview of the preliminary procedures to set-up regression models that build on text data. This will cover *web-scraping*, data pre-processing and the procedures to set-up matrices of textual predictors. Also, we briefly introduce the fundamentals of hedonic models. In fact, *text-regression* may be interpreted in terms of a hedonic model.

2.1 Literature Review

While there is plenty of methodological and applied contributions on the usage of text data for document clustering and classification (see Weiss et al. [2015], Berry and Castellanos [2004], Berry and Kogan [2010], Steyvers and Griffiths [2007] for some reviews), not as many research has been produced on the usage of text for regression purposes. Recently, many documents (mostly from the web) are frequently associated with quantitative variables. To give some examples: in a collection of movie reviews, each document is summarized by a numerical rating; in a collection of news articles, each document is assigned to a section of the newspaper; in a collection of on-line scientific articles, each document is downloaded a certain number of times. In this study, listings of products sold in online fashion stores contain descriptions assigned to product prices. Therefore, the task of predicting discrete or continuous outcomes using text as input has been referred to as *text-regression* [Gentzkow et al., 2019].

In general, text data contains a lot in information which can be hardly leveraged. Computer scientists were the first to improve the interpretation of documents by introducing machine learning methodologies to discover latent *concepts* - or *topics* - inside collections of documents. Among these, the most widely used method is probably the Latent Dirichlet Allocation (LDA) [Blei et al., 2003], a generative probabilistic model for topic modelling that has also been used to create features for classification and regression purposes. In fact, it can be regarded to as a method for dimensionality reduction via probabilistic matrix factorization [Fei-Fei and Perona, 2005],[Aggarwal and Zhai, 2012]. Still, unsupervised topic modelling and dimensionality reduction like LDA have been criticized because the extracted features may not be correlated to the outcome to predict. Mcauliffe and Blei [2008] proposed supervised LDA as an extension of traditional LDA. The authors jointly model documents and response variables to find topics that best predict the response given a set of new observed documents. As alternative approach to supervised LDA, Taddy [2013] proposed Multinomial Inverse Regression (MNIR) as a new model of annotated text based on the influence of metadata and response variables on the distribution of words in a document. Motivated by previous work by Gentzkow and Shapiro [2010], the author shows that logistic regression of phrase counts onto document annotations (like ratings) can be used to obtain lower dimensional document represen-

tations that are rich in *sentiment* information. The author perform three different text analysis. First, they take constituent percentage vote-share for G.W. Bush and Republican party membership regressed onto speech for a member of the 109th US congress. Second, they take users' overall ratings regressed onto the content of their we8there.com restaurant review. Rabinovich and Blei [2014] proposed an extension of MNIR, the Inverse Regression Topic Model (IRTM), combining the strength of the former with LDA. They apply the model to a corpus of 73K congressional press releases and to a corpus of 150k yelp reviews. Their results show that IRTM outperforms both MNIR, supervised LDA and LDA based regression. Büschken and Allenby [2016] provide another extension of Latent Dirichlet Allocation by introducing a *bag-of-sentences* model in contrast to the standard bag-of-words assumption. They use data from Italian restaurants reviews from we8there.com and hotels from expedia.com to predict consumer ratings through regression.

In the field of empirical research, many contributions have been published in different domains which are summarized in Table 2.1. It is evident that most applications are concerned with finance. In general, these aim at predicting price volatility or stock returns based on news documents. The majority of these studies try to understand whether financial fenomena are related to news, intended as the mood of the markets. These methods are mostly concerned with linear regression model for time series analysis, with some exceptions that use Support Vector Regression (SVR). An early example on analyzing news text for stock price prediction appears in Cowles 3rd [1933]. The closest modern analog of Cowles's study is Antweiler and Frank [2004], who take a generative modeling approach to ask how informative are the views of stock market prognosticators who post on internet message boards. The authors classify postings on stock message boards as *buy*, *sell*, or *hold* signals. In Tetlock et al. [2008], the authors use word counts in the Wall Street Journals' widely read "Abreast of the Market" column. Using the Harvard VI psychosocial dictionary, they map counts from each article to a high dimensional sentiment score vector and derive a "pessimism factor" using Principal Components to forecast stock market activity. Dictionary based applications can be found in Loughran and McDonald [2011] where the authors demonstrate that the widely used Harvard dictionary can be ill-suited for financial applications. Bollen et al. [2011] instead analyze how the mood of daily Twitter feeds can be used to predict the Dow Jones Industrial Average over time. Kogan et al. [2009] use a company's

annual financial report to predict the financial risk of investment in that company, measured empirically by a quantity known as stock return volatility. They use support vector regression and find text-regression model predictions to correlate with true volatility nearly as well as historical volatility. Wisniewski and Lambe [2013] show that negative media attention of the banking sector, summarized via ad hoc pre-defined word lists, Granger-causes bank stock returns during the 2007-2009 financial crisis and not the reverse, suggesting that journalistic views have the potential to influence market outcomes, at least in extreme states of the world. The use of text regression for asset pricing is exemplified by Jegadeesh and Wu [2013]. They estimate the response of company-level stock returns to text information in the company's annual report. The authors' objective is to determine whether regression techniques offer improved stock return forecasts relative to dictionary methods by proposing a specific regression model. Manela and Moreira [2017] take a regression approach to construct an index of news-implied market volatility based on text from the Wall Street Journal from 1890-2009, by applying support vector machines. Bandiera et al. [2020] apply Latent Dirichlet Allocation to a large panel of CEO diary data. They uncover two distinct behavioral types that they classify as *leaders* who focus on communication and coordination activities, and *managers* who emphasize production related activities. Thorsrud [2018] shows how textual data collected from a major Norwegian business newspaper can be used to construct a daily coincident index of the business cycle within a mixed frequency time varying dynamic factor model. A related line of research analyzes the impact of communication from central banks on financial markets. Lucca and Trebbi [2009] use the content of Federal Open Market Committee (FOMC) statements to predict fluctuations in Treasury securities. Born et al. [2014] extend this idea to study the effect of central bank sentiment on stock market returns and volatility. Hansen et al. [2018] research how FOMC transparency affects debate during meetings by studying a change in disclosure policy. The authors use topic modeling to study 149 FOMC meeting transcripts during Alan Greenspanas tenure. Some studies use text to infer causal relationships or the parameters of structural economic models. Engelberg and Parsons [2011] measure local news coverage of earnings announcements, then use the relationship between coverage and trading by local investors to separate the causal effect of news from other sources of correlation between news and stock prices.

Another domain of application can be found in studies concerning political slant and policy uncertainty. In this category, the methods adopted build mostly on logistic regression and multinomial logistic regression, which underlies the Multinomial Inverse Regresion (MNIR) and the Inverse Regression Topic Model (IRTM). These approaches are also close to the applications in the field of opinion mining, in fact, both are closely related to sentiment analysis [Liu, 2015]. Baker et al. [2016] provide one of the most influential applications of text analysis in the literature to date defining a measure of economic policy uncertainty (EPU). The authors define the unit of observation to be a country-month and the outcome of interest is the true level of economic policy uncertainty. The authors apply a dictionary method to produce estimates of uncertainty based on digital archives of ten leading newspapers in the United States. Groseclose and Milyo [2005] offer a pioneering application of text analysis methods to this problem. They compare the text of large US newsmedia outlets to speeches of congresspeople in order to estimate the outlets' political slant. Gentzkow and Shapiro [2010] use congressional and news text to estimate each news outlets' political slant in order to investigate the supply and demand forces that determine slant in equilibrium. Stephens-Davidowitz [2014] uses Google search data to estimate local areas' racial animus in order to address the causal effect of racial animus on votes for Barack Obama in the 2008 election. Saiz and Simonsohn [2013] use web search results to estimate the current extent of corruption in US cities using a dictionary approach.

In terms of the data used as input, the contributions above share some similarities with those in the field of nowcasting. Following a bayesian approach, Scott and Varian [2014] and Scott and Varian [2013] use data from Google searches to produce high-frequency estimates of macroeconomic variables such as unemployment claims, retail sales, and consumer sentiment that are otherwise available only at lower frequencies from survey data. Zeng and Wagner [2002] note that the volume of searches or web hits seeking information related to a disease may be a strong predictor of its prevalence. Johnson et al. [2004] provide an early data point suggesting that browsing influenza-related articles on the website healthlink.com is correlated with traditional surveillance data from the Centers for Disease Control (CDC). A similar approach has been carried out in [Ginsberg et al., 2009].

In the field of industrial organization and market definition, many important questions hinge on the appropriate definition of

product markets. Nevertheless, standard industry definitions can be an imperfect proxy for the economically relevant concept. Hoberg and Phillips [2016] provide a novel way of classifying industries based on product descriptions in the text of company disclosures. Kelly et al. [2018] use cosine similarity among patent documents to create new indicators of patent quality.

Eventually, we report the major contributions in the field of empirical economics. Foster et al. [2013] provide an example on the usage of text for regression. The authors use listings for real estates that report the price of a property, a description about its key features and details such as the number of rooms, bathrooms and square-feets to build a hedonic regression model. Nowak and Smith [2017] provide another application of text regression in real estate. They incorporate text data from MLS listings into a hedonic pricing model and found text, used in combination with other covariates, to be an important determinant to reduce pricing error. They compare the performances of two common penalized regression techniques, namely the Lasso [Tibshirani, 1996] and the procedure proposed in [Belloni et al., 2011]. Joshi et al. [2010] provide an experiment in text regression on movie reviews and revenue. They use a linear regression to predict the gross revenue aggregated over the opening weekend. They obtain data from movie reviews from 2005 to 2009 by crawling metacric.com as well as data about budget and revenue by crawling the-numbers.com. They use the elastic net [Zou and Hastie, 2005] for model estimation. Ghose et al. [2007] set up an econometric model to infer the economic value of text by giving a value to each opinion phrase using transaction and reputation data from Amazon.com. Archak et al. [2007] and Archak et al. [2011] use text-mining to study the influence of textual product reviews on product choice decisions. Following Chevalier and Mayzlin [2006], they model the impact of product reviews on sales by directly incorporating product review information in a linear model for the sales rank. To this purpose, they use data from Amazon web services on 41 digital cameras and camcoders.

The present study joins to the aforementioned contributions in empirical economics. In particular, our approach is connected with the studies by Foster et al. [2013] and Nowak and Smith [2017]. However, beyond the domain of application, our approach is different in the sense that both these studies use text data as supplement of another set of *structured* covariates while we use just the attributes that we obtain from text. In terms of the methods discussed in [Nowak and Smith, 2017, Foster et al., 2013], our work

goes in their same direction but in addition, our approach explores supervised dimensionality reduction, topic modeling via LDA, as well as variants of penalized linear model that have not been addressed before.

Table 2.1: Review of studies in regression using text data

Author	Year	Outcome	Input	Method
Finance				
Cowles 3rd	1933	stock prices	news text	LR
Tetlock et al.	2008	Firms' earnings, stock prices	Sentiment	LR
Antweiler and Frank	2004	stock prices	message boards	LR
Wisniewski and Lambe	2013	stock returns	Text	LR
Jegadeesh and Wu	2013	stock returns	companies ann. rep.	LR
Kogan et al.	2009	Volatility	Text	SVR
Loughran and McDonald	2011	Returns	Sentiment	LR
Tetlock et al.	2008	Firms' earnings, stock prices	Sentiment	LR
Engelberg and Parsons	2011	Stock prices	earning announc.	LR
Manela and Moreira	2017	market volatility	WS Journal	SVR
Bandiera et al.	2020	Managements type	CEO diary data	LDA
Economics				
Chevalier and Mayzlin	2006	sales ranks	reviews	LR
Archak et al.	2011	Sales ranks	Text	LR
Foster et al.	2013	Price	Text	LR
Nowak and Smith	2017	Price	Text	LR

Continued on next page

Table 2.2 – *Continued from previous page*

Author	Year	Outcome	Input	Method
Thorsrud	2018	Business cycle	business newsp.	DFA
Marketing				
Büschken and Allenby	2016	Ratings	Topic-content	Topic modeling, LR
Hoberg and Phillips [2016]	2016	Industry type	Product desc.	LR
Kelly et al.	2018	patent	Text	Clustering
Political slant, policy uncertainty				
Gentzkow and Shapiro	2010	Doc. Annotations	Phrase counts	Log. Regression
Groseclose and Milyo	2005	political slant	US newsmedia outlets	Mult.reg
Stephens-Davidowitz	2014	google search	vote shares	LR
Saiz and Simonsohn	2013	web searches	corruption	LR
Taddy	2013	vote share, ratings	text	MNIR
Rabinovich and Blei	2014	party affiliation, ratings	Text, Sentiment	IRTM
Nowcasting				
Scott and Varian	2013	macroeconomic indexes	google searches	Bay.Reg
Scott and Varian	2013	macroeconomic indexes	google searches	Bay.Reg
Zeng and Wagner	2002	disease mapping	web searches	LR
Opinion Mining				

Continued on next page

Table 2.2 – *Continued from previous page*

Author	Year	Outcome	Input	Method
Bollen et al.	2011	DJIA	Feeds sentiment	LR
Ghose et al.	2007	Price premium	Reputation	LR
Joshi et al.	2010	Movie revenue	n-grams	Elastic Net
Mcauliffe and Blei	2008	rating	Topics	sLDA

2.2 Microeconomic background of hedonic models

A Hedonic regression model regresses the price of one unit of a commodity on a function of the characteristics of the model and a time dummy variable [Feenstra and Shapiro, 2007]. It is assumed that a sample of model prices can be collected for two or more time periods along with a vector of the associated model characteristics. An interesting theoretical question which is discussed by Rosen [1974] is whether we can provide a microeconomic interpretation for the function of the characteristics on the right hand side of the regression. Here, we illustrate the model in [Feenstra and Shapiro, 2007] which develops on the former based on the following assumptions

- every consumer has a separable sub-utility function $f(\boldsymbol{z}), \boldsymbol{z} \in \mathbb{R}^n$ which gives sub-utility $Z = f(\boldsymbol{z})$, from the purchase of one unit of a commodity that has a vector of characteristics $\boldsymbol{z}$
- The sub-utility that consumer gets from consuming Z units of a commodity can be combined with the utility derived by consumption of another commodity X to give an overall (*macro*) utility $u^t = U^t(X, Z)$ in time period t.

The set $\{(X, Z) : U^t = u^t\}$ is the consumer indifferent curve for period t. It is further assumed that utility can be solved for X (explicitly or implicitly) and that $\partial g^t(U, Z)/\partial Z < 0$. Now, let p^t and P^t be, respectively, the prices for one unit of X and Z, then the consumer expenditure minimization problem is

$$\min_{X,Z}[p^t X + P^t Z : X = g^t(u^t, Z)] = \min_{Z}[p^t g^t(u^t, Z) + P^t Z] \quad (2.1)$$

The price of the hedonic aggregate P^t can be derived by setting the gradient of the right hand side of 2.1 to zero to obtain

$$P^t = -\frac{p^t \partial g^t(U, Z)}{\partial Z} := \omega^t(Z, u^t, p^t) \quad (2.2)$$

where ω is interpreted as the consumer willingness to pay price function. In order to come at the final hedonic regression model, it assumed also that there are K^t models available at time t, where model k sells at unit price P_k^t as has characteristics $z_k^t \in R^{n_k}$. If the consumer decides to purchase from model k then we can equate his willingness to pay for one unit to price P_k^t so that

$$P_k^t = -\frac{f(\boldsymbol{z}_k^t) p^t \partial g^t(u^t, f(\boldsymbol{z}_k^t))}{\partial Z} \quad (2.3)$$

where $f(\boldsymbol{z}_k^t)p^t\partial g^t(u^t, f(\boldsymbol{z}_k^t))/\partial Z = Z\omega^t(Z, u^t, p^t)$ is the amount of money that a consumer is willing to pay for a model with characteristics $\boldsymbol{z}$ (where we used the fact that $Z = f(\boldsymbol{z})$). No statement on g has been made so far, it can be assumed that every consumer has the same subutility function f and that the i-th consumer is given the following linear indifference function

$$g_i^t(u_i^t, Z) = -a^t Z + b_i^t u_i^t$$

where $a^t, b^t > 0$. Substituting in 2.3 we get easily

$$P_k^t = p^t a^t f(z_k^t) = \rho^t f(\boldsymbol{z}_k^t) \tag{2.4}$$

This equation will consitute the basis to derive the most used functional forms of hedonic regression models, namely the log-log, the semilog and the linear. Assume that

$$\log f(z) = \alpha_0 + \sum_i \alpha_i \log z_i \tag{2.5}$$

the log-log hedonic model can be derived easily by taking logs of 2.3 plugging 2.5 and adding an error term ϵ

$$\log P_k^t = \log \rho^t + \alpha_0 + \sum_i \alpha_i \log \boldsymbol{z}_{ik}^t + \epsilon_i^t{}^{1} \tag{2.6}$$

On the other hand if it is assumed that

$$\log f(\boldsymbol{z}) = \alpha_0 + \sum_i \alpha_i z_i \tag{2.7}$$

subsitututing in 2.3 plugging 2.5 and adding an error term ϵ we obtain the following semilog hedonic regression model

$$\log P_k^t = \log \rho^t + \alpha_0 + \sum_i \alpha_i z_{ik}^t + \epsilon_i^t{}^{2} \tag{2.8}$$

In the linear model it is assumed that f is a linear function of the characteristics so that the hedonic regression model turns out to be

$$P_k^t = \rho^t(\alpha_0 + \sum_i \alpha_i z_{ik}^t) + \epsilon_i^t \tag{2.9}$$

which is non-linear in the parameters and thus it is not usually estimated. This model is usually approximated by its linear version

[1] $\log \rho^t = 0$ is set for idendifiability
[2] $\log \rho^t = 0$ is set for idendifiability

$$P_k^t = \rho^t + \alpha_0 + \sum_i \alpha_i z_{ik}^t + \epsilon_i^t \tag{2.10}$$

The choice of the most suitable functional form has been object of debate, see for example Rosen [1974], Cassel and Mendelsohn [1985], Freeman III [1979] and Halvorsen and Pollakowski [1981]. In what follows we use a linear specification as in 2.10 since our focus is the actual level of price.

2.3 Web-Scraping

Web scraping is the automated collection of data from the internet. Although web scraping is not a new term - in past years this practice has been more commonly known as *screen scraping*, *web data mining*, *web harvesting* - general consensus today seems to favour web scraping. Occasionally, we will make use of *web-crawling* as synonymous.

A complete description of how to do web scraping would be impossible, since there are too many different ways and solutions to that, depending on the specific needs. These start from the programming language to be used to its actual implementation. Of course, web scraping is not a statistical procedure per-se, yet, the most common statistical programming languages are coming with several good packages to perform scraping routines. For example, data scientist or statisticians may prefer using R or Python. For the purpose of this work, we used Python Scrapy as environment, although other packages like beautiful soup (in Python) and rvest (in R) are excellent alternatives. The main steps that a web-crawler needs to accomplish are outlined below, with more technical details reported in Algorithm 1. These are:

1. Retrieving HTML[3] data from a domain name
2. Parsing that data for target information
3. Storing the target information
4. Moving to other pages and repeat (*optional*)

It is worthwhile to examine the specific issues that arise in extracting text from the Web. To begin with, several aspects of text extraction are highly platform-specific. Broadly speaking, this

[3]Hyper-Text Markup Language

means that each website has its own structure so that it needs a specificly designed crawler to get its data collected. A Web page may often be organised into content blocks that are not related to the primary subject matter of the page. For example, irrelevant blocks such as advertisements, disclaimers, or notices, that are not very helpful to mine. The web-crawler has to be designed so that when it scrolls a web page looking for data, the information contained in these blocks is not considered.

Algorithm 1: Pseudo code for web-scraping

```
Data: Web pages URL, N - number of search result pages
      (i), n - number of results on search page (j)
Result: JSON file with scraped data
Send request via URL to server;
for i ← 1 to N do
    Set i result page;
    Set n;
    for j ← 1 to n do
        Send request to server for j-th page from the search
         results ;
        Get the data from HTML through X-Path and
         Regex expression;
        Store data in a JSON file;
    end
end
```

In brief, the basic action that a web crawler needs to perform is to scroll the entire HTML of the web page looking for specific *tags* containing the desired information. HTML is a tree structure with nodes and leafs, indexed by tags, where the latter usually contains the information that one needs to scrape. XPath is a querying language that allows to scroll the tree structure to get to the leaf containing the needed information. Regex are programming expressions that can automatically detect specific patterns in text by manipulating text strings with the purpose to isolate the desired information[4]. As discussed, web data like text are unstructured, therefore it is not recommended to store this data in a standard csv file. The most appropriate way to collect data from web scraping

[4]As example, the find and replace function of many software is a front end implementation of some regex

is to store it in Javascript Object Notation (JSON) format which can be easily handled by the most common statistical programming languages [Kwartler, 2017].

2.4 Preprocessing text

Once that text data has been collected, the first step is to convert the raw text into a character sequence. We refer to the raw text as a character sequence that contains a significant amount of meta-information. For example, an HTML document will contain various tags and anchor text, and an XML document will contain meta-information about various fields. Here, the analyst has to make a judgement about the importance of the text to the specific application at hand, and remove all the irrelevant meta-information. The character sequence is then parsed into *tokens*. Consider the following example of a character sequence:

> *After sleeping for four hours, he decided to sleep for another four.*

The tokens from this character sequence are ("*After*", "*sleeping*", "*for*", "*four*", "*hours*", ",","*he*", "*decided*","*to*","*sleep*","*for*","*four*","*hours*",".").
Note that some words are repeated multiple times and some are not consolidated. In addition, some words are capitalized. Thus, a token is a sequence of characters from a text that is treated as an indivisible unit for processing. Tokenization presents some challenging issues from the perspective of deciding word boundaries. A very simple and primitive rule for tokenization is to use white spaces as separators after removing punctuation. This method is simplistic and may lack of human interpretation when words naturally come in pairs or exhibit a specific pattern that has to be captured in the analysis. A much richer tokenization would include also sequences of n adjacent words, the so-called *n-grams*. It must be noticed though that there is no unique way of performing the best tokenization. To give and example, consider the following description of a dress scraped from an e-commerce website of a fashion retailer:

> *Not the usual pencil dress: the V-neck with tulle insert and the star appliques are that something extra you will immediately fall in love with. Ultra-fitted bodycon cut that shows off your silhouette. Perfect with soaring heels at evening parties.*

White-space tokenization will separate the words *pencil* and *dress*. This is not a problem if the analysis were carried out on just this category but it would be very different if we were to model different categories of apparels at once. In this case, the word pencil may not be immediately be associated with dresses but with other categories of apparels. The use of bi-grams would consider *pencil dress* as a distinct token. Of course, reacher tokenizations correspond to better interpretability, but this is not coming without side effects, as it shall be discussed.

Tokens that can be extracted from a character sequence may not be all of practical use. The goal of the analysis is to discard some tokens and to retain the subset that preserve the core content of the text. To this purpose, the following steps are usually recommended. However, these are just indicative so that some can be skipped or performed in alternative order [Aggarwal and Zhai, 2012, Kwartler, 2017].

removing stop-words *Stop-words* are words in language that do not add much content to a sentence. These are usually articles, conjunctions, prepositions and pronouns which occur very frequently in the collection. These words typically occur at one end of the spectrum, they are not discriminative for most mining applications and only add a large amount of noise. Depending on the specific domain This list can be enlarged with other words. For example, when dealing with a collection of reviews on dresses, the word *dress* may not add any specific content to the review.

removing punctuation Punctuation is usually not assumed to be a key role in statistical applications and is often removed. This includes also hyphens, accents, apostrophes.

stemming Stemming is the process of consolidating related words with the same root. For example, a text document might contain the singular or plural form of the same word, various tenses, and other variations. In such cases, it makes sense to consolidate these words into a single one. Consolidation plays a crucial role for subsequent statistical analysis; since terms can be used as covariates, consolidation has a direct impact on the dimensionality of data.

lemmatizations Often referred to as stem-completion. After stemming it yields the dictionary word. The most widely used stemming-lemmatization algorithm are the Porter's stemmer

[Van Rijsbergen et al., 1980], Snowball [Porter, 2001] which applies to other languages and WordNet [Miller, 1995].

By filtering out some tokens, pre-processing text reduces the dimensionality of the problem. In text mining or statistical natural language processing this is known as feature selection [Guyon and Elisseeff, 2003]. In addition to the steps outlined above, pre-processing can be further encouraged by filtering out tokens with low frequencies based on some thresholding rule. Other criteria that have been proposed in the literature, in particular term-variance [Liu et al., 2005] and term-variance quality [Dhillon et al., 2004] consider the variance of term frequencies to define thresholding rules.

The set of remaining tokens after pre-processing is referred to as the *lexicon* or the *vocabulary*. Some text mining books [Aggarwal, 2018] make a further distinction and refer to these left tokens as *terms* to better establish the connection with the idea of a vocabulary.

2.5 Weighting schemes and the vector space model

In order to use text for statistical applications it is necessary an appropriate model. The most famous model in text mining literature is known as *Bag of Words* (BoW). In this model, the documents are given a sparse multidimensional representation where the dimensions correspond to terms or *features*. This is the reason why we stated that reacher tokenizations may have a cost to be paid. In this case, the cost is directly related to the dimensionality of data.

Let $\mathcal{D}$ be a collection of D documents. Suppose that documents have been pre-processed using the criteria discussed above and let $\mathcal{V}$ be a vocabulary of size V. The vocabulary is retrieved according to the tokenization (words, bi-grams, words and bigrams,...) that the researcher believes is the best for the problem at hand. Usually, the most common and straightforward choice is to identify tokens in words so that throughout this study, we will assume tokens to be words. After pre-processing, each document $\boldsymbol{w}$ is represented as a point in a V-dimensional vector space, $\boldsymbol{w} \in \mathbb{R}^V$, whose dimensions correspond to the words in the vocabulary. A corpus $\mathcal{D}$ is then equivalently defined as a collection $\mathcal{D} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_D\}$ of points in this space. Given that the size of the vocabulary is often very large, documents are points in a high-dimensional vector space.

The matrix

$$\mathbf{W} = \begin{pmatrix} \boldsymbol{w}_1' \\ \dots \\ \boldsymbol{w}_D' \end{pmatrix} \tag{2.11}$$

is referred to as the Document-Term Matrix (DTM). Equivalently, its transpose is referred to as the Term-Document Matrix (TDM). Since each document only uses a small portion of the vocabulary, this matrix is highly sparse. Also, the vector space model is based on the implicit assumption that the order of words in documents can be neglected. This is the reason why this model has been named after *bag-of-words* in text mining literature [Jurafsky and Martin, 2014].

For Document Term Matrices, entries w_{ij} can be defined in different ways, depending on the requirements of the analysis. Let tf_{ij} and df_j denote respectively, the number of occurrences of word j in document i and the number of documents in the collection containing word j. Three popular ways of defining weights are the following:

Term Frequency captures how salient a word is within a given document. The higher the frequency the more likely it is that the word is a good description of the content of the document. Thus

$$w_{ij} := \text{tf}_{ij}$$

Term frequency is usually dampened by some concave function such that $f(x) = \sqrt{(x)}$ or $f(x) = 1 + \log(x)$. This is because un-dampened occurrences may overestimate the importance of words [Aggarwal, 2018]. Alternatively, w_{ij}-s can be defined as relative frequencies by dividing by the number of words in the document.

Term Presence Sometimes the presence or absence of a specific word is likely to be more important than its frequency. Therefore, a binary weighting scheme, also known as one-hot encoding, or *Bernoulli* is defined as

$$w_{ij} := 1_{\text{tf}_{ij}>0}$$

Term Frequency-Inverse Document Frequency Salton and McGill [1986]. This weighting scheme is widely used in the literature on text mining and information retrieval because it puts

higher weight on words having high frequency in the document and low frequency in the collection. Conversely, it gives less weight to most common terms. The weights under Term Frequency Inverse Document Frequency are defined as:

$$w_{ij} := \text{tf}_{ij} \times \log^{-1}(D/\text{df}_j)$$

For the sake of completeness, a new stream of the literature Mikolov et al. [2013a,b], and Pennington et al. [2014] introduced vector space representations based on Neural-Network embeddings. These representations are becoming quite popular in text-mining applications to statistics [Lenz et al., 2018]. The key features of these methods is to produce very meaningful interpretations of word representations.

2.6 Challenges and difficulties in automatic processing of text

The Hedonic Pricing Model (HPM) assumes that prices of differentiated goods can be described by a bunch of measurable features/attributes and that the consumer's valuation of a good can be decomposed into implicit values of each product feature [Rosen, 1974]. With a slight change of notation with respect to the model in 2.10, let p_i denote the price for item i and let $\boldsymbol{x}_i \in \mathbb{R}^p$ be a bundle of embodied attributes valued by some implicit or shadow prices embedded in the vector $\boldsymbol{\beta} \in \mathbb{R}^p$[Baltas and Saridakis, 2010]. The hedonic model in 2.10 can be reformulated as

$$p_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i \tag{2.12}$$

where ϵ_i is a zero mean and finite variance error term. Note that in 2.12 we suppress the time index which is considered in 2.10 and previous models since we are interested in explaining sectional prices' variations.

The traditional approach to hedonic modeling assumes that the choice of which attributes have to be included in the model is made by the researcher. On the contrary, in this study we aim at a data-driven selection of predictors that leverages the information contained in the descriptions of the items on sale. In other words, we are interested in answering to questions like what tokens are relevant to explain the variations within prices and which method is performing best among the ones that we consider. We are aware

that a punctual evaluation of attributes - for example in terms of a detailed composition of materials - may improve the interpretation in prices' variation, however, this procedure is affordable just for very small datasets and does not scale to bigger ones.

The advantage of using web-data is that products are always provided with a textual descriptions that contains very detailed information about their attributes. This means that if it were possible to leverage this data to obtain this rich set of attributes, it would be also possible to reduce some bias due to omitted variables. In addition, this procedure is applicable to datasets of any size.

Therefore, compared to traditional HMP where data is modeled as a collection of prices and attributes, $\{(p_i, \boldsymbol{x}_i)_{i=1}^n\}$, in this framework the goal is to model the collection $\{(p_i, \text{text}_i)_{i=1}^n\}$ where text_i is the textual description of the good associated with price i. In light of what introduced in Section 2.5, let $\mathcal{D}$ be a collection of D product descriptions and let $\mathcal{V}$ be the vocabulary[5] of size V associated to $\mathcal{D}$. In addition, let $\mathbf{W} \in \mathbb{R}^{D \times V}$ be Document Term Matrix collecting the vector space representation of the descriptions. In the hedonic text-regression pricing model we let the vector of attributes of a given product to be the representation its description in the vector space defined in Equation 2.11:

$$\boldsymbol{x}_i^{'} := \boldsymbol{w}_i' \in \mathbb{R}^V$$

obtaining the following specification

$$p_i = \beta_0 + \beta_1 w_{i1} + \cdots + \beta_V w_{iV} + \epsilon_i \tag{2.13}$$

Remarkably, to estimate 2.13 corresponds to estimate the implicit value of each word in the description, and thus the hedonic value of the description.

The key feature of this model is that it allows for both observable features - like materials - and for intangible product features - quality of product, design, ease of use, robustness - that probably would have not been included following a traditional approach. To see this, consider the example illustrated in Figure 2.1i. The price for this item is 337 euros. The hedonic text regression model will estimate the implicit price of intangible product features denoted by words *delicate, wild, young*, clearly defining the style of the blouse, while allowing for observable attributes like materials (*silk, georgette*). Also, at least in principle, it is possible to estimate the hedonic value of the item combined with another that form a

[5]obtained after the pre-processing steps illustrated in Section 2.4

particular outfit. For example, this would be the marginal effect of the word *heel* for the item in Figure 2.1ii.

(i) *A dappled print invades this lightweight blouse in silk dappled print invades this lightweight blouse in Georgette fabric. A style with a young, wild spirit that will sublimely complete casual looks with jeans or tailored trouser*

(ii) *Lightweight and airy, this classic-cut blouse is made unique by the tulle star insert that adorns the neckline. A delicate design that you will love with a suit and 12 cm heels.*

Figure 2.1: Example of descriptions

Given the importance that words can have in explaining prices, a proper estimator of the vector of coefficients is fundamental. In fact, not all the words that are contained in product descriptions need to be good predictors of prices. Again, consider the example in Figure 2.1i. From an economic perspective, words like *looks, spirit* would not be as strong as *dappled, silk*, *georgette*, or *lightweight* in determining prices. In this case it is assumed that only a subset $\mathcal{S}$ of the set of features (i.e. words) actually determines prices. Thus, to estimate a Hedonic model like 2.13 is also a matter of selecting a suitable statistical learning algorithm to retrieve this subset.

Unfortunately, when dealing with text data, traditional variable selection procedures like best subset selection or forward/backward stepwise selection are rarely of practical use. The first consists in comparing 2^V model which is prohibitive when V - the size of the vocabulary - is greater than 40. The latter can still be applied as

long as D - the documents in the collection - is less than V but they are way computationally expensive. Moreover, if one were to include also non linearities, interactions or a richer representation of text based on n-grams the number of covariates obtained by featurizing text may become larger than the available sample size.

Another challenge of text data is that the bag-of-words model fails to deal with two aspect of natural language, namely *polysemy* and *synonymy* as it gives to synonyms and polysemys different dimensions in the vector space. Also, the high-dimensionality and the sparsity of the DTM matrix sometimes make necessary the usage of dimensionality reduction techniques to improve the interpretation of data.

CHAPTER

3

METHODS

In Section 2.1, we introduced the Bag of Words model to summarise the information in document collection via a sparse Document-Term Matrix. This chapter introduces additional statistical models that use text to explain the variation of a continuous outcome like price. Gentzkow et al. [2019] distinguish between i) *text-regression* models, where they account also dimension reduction, penalized linear models and other non-linear models ii) generative language models iii) word embeddings and iv) uncertainty classification. Given that the interpretability of results is the priority of this study we decided to focus on the first two.

In particular, we review two alternative models for text data, Latent Semantic Indexing (LSI) [Deerwester et al., 1990] and Latent Dirichlet Allocation (LDA) [Blei et al., 2003] - which belong to the class of dimensionality reduction and topic modelling techniques - some variations of penalized linear models (Lasso, SLOPE) and aggregated predictors. Eventually, in Section 3.8 we discuss their implementation as well as the procedures for perfomance evaluation.

3.1 Latent Semantic Indexing

The intuition behind Latent Semantic Indexing is that by comprising the DTM with a lower-rank approximation, one may obtain a lower dimensional representation that best represents co-occurrences among words in documents. As example, those due to polysemy and synonymyms. This intuition suggests that not only should retrieval quality not suffer too much from the dimension reduction, but in fact may improve. For this reason, LSI has been regarded also as a topic modeling technique [Steyvers and Griffiths, 2007, Aggarwal, 2018, Crain et al., 2012, Evangelopoulos et al., 2012].

A topic model is a statistical model for discovering latent *topics* - or *concepts* - inside a collection of documents [Blei et al., 2003]. Topic models usually factorize a Document-Term Matrix into a product of two or three matrices. Usually, one of these gives insights about the topics' dominance inside each document, and another explains how much each topic is defined by the words in the vocabulary. LSI is obtained by means of a lower rank approximation $\mathbf{W}_k$ of the DTM $\mathbf{W}$ such that $\mathbf{W}_k$ can be decomposed in the product of three matrices. The value of the chosen rank, k, identifies the number of concepts in data [Manning et al., 2010]. This decomposition can be represented as

$$\mathbf{W} \approx \mathbf{W}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k'.$$

The matrices $\mathbf{U}_k \in \mathbb{R}^{D \times k}, \mathbf{V}_k \in \mathbb{R}^{V \times k}$ in this decomposition represent respectively the coordinates of documents and terms in the lower dimensional latent space of concepts or topics. The remaining matrix instead says how much each topic dominates the collection of documents. This is illustrated in Figure 3.1. The j-th topic is interpreted according to the magnitude and the sign of the coordinates with respect to the j-th column of $\mathbf{V}_k$. Opposed to the sparse bag of words model, this representation is dense.

Singular Value Decomposition Given a generic Document Term Matrix $\mathbf{W} \in \mathbb{R}^{D \times V}$, let $\mathbf{U}$ be a $D \times D$ matrix such that its columns $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_D\}$ are the eigenvectors of $\mathbf{W}\mathbf{W}'$ and let $\mathbf{V}$ be a $V \times V$ matrix whose columns $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_V\}$ are the eigenvectors of $\mathbf{W}'\mathbf{W}$.

Theorem 1 *Let r be the rank of matrix $\mathbf{W}$. Then, there is a singular value decomposition (SVD) of $\mathbf{W}$ such that*

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}' \tag{3.1}$$

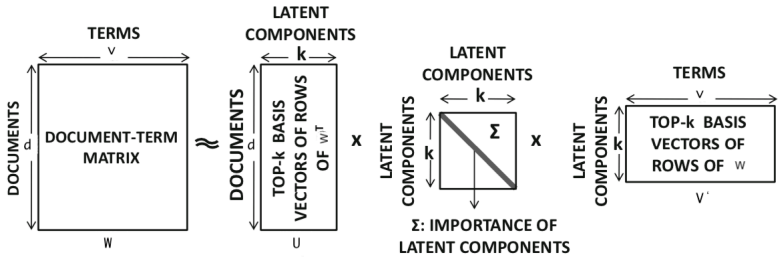


Figure 3.1: Visual representation of LSI for a given Document Term Matrix.

where

1. *the eigenvalues* $\lambda_1, ..., \lambda_r > 0$ *of* $\mathbf{WW}'$ *are the same as those of* $\mathbf{W}'\mathbf{W}$.

2. *Matrix* $\mathbf{\Sigma} \in \mathbb{R}^{D \times V}$ *is a diagonal matrix such that for* $1 \leq i \leq r$, $\Sigma_{ii} = \sigma_i = \sqrt{\lambda_i}$.

The values σ_i are referred to as the *singular values* of $\mathbf{W}$. When practically computing the SVD, it is more convenient to represent $\mathbf{\Sigma}$ as an $r \times r$ matrix with the singular values on the diagonal, since all entries outside this sub-matrix are zeros. Accordingly, it is conventional to omit the rightmost $D - r$ columns of $\mathbf{U}$ (corresponding to omitted rows of $\mathbf{\Sigma}$), and the rightmost $V - r$ columns of $\mathbf{V}$. The cost of computing the SVD is then reduced from $\mathcal{O}(VD^2)$ to $\mathcal{O}(Vr^2)$. This way of representing the SVD is known as *compact* SVD. This is shown in Equation 3.2 below and graphically in Figure 3.2

$$\mathbf{W}_{D \times V} = \mathbf{U}_{D \times V} \mathbf{\Sigma}_{r \times r} (\mathbf{V}_{V \times r})' \tag{3.2}$$

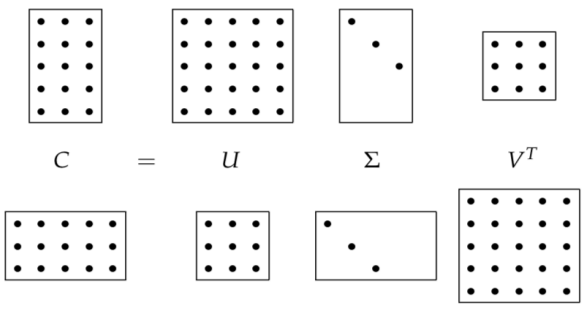


Figure 3.2: Singular value decomposition.

Low-rank approximation Given a Document Term Matrix $\mathbf{W}$ and a scalar k we wish to find a matrix $\mathbf{W}_k \in \mathbb{R}^{D\times V}$ of rank k such that the quantity

$$\|\mathbf{W} - \mathbf{W}_k\|_F = \sqrt{\sum_{i=1}^{D}\sum_{j=1}^{V} W_{ij}^2} \tag{3.3}$$

is minimized. The Singular Value Decomposition is one solution to this problem [1].

The error that is obtained by this appoximation is given by

$$\min_{Z|\text{rank}(Z)=k} \|\mathbf{W} - \mathbf{Z}\| = \|\mathbf{W} - \mathbf{W}_k\|_F = \sigma_{k+1}$$

Thus, the larger is k the lower is the approximation error.

Definition 1 (Trucated SVD) *Let $k \ll r$ then the Truncated SVD of W is defined as*

$$\mathbf{U}_{D\times k}\mathbf{\Sigma}_{k\times k}(\mathbf{V}_{V\times k})' \tag{3.4}$$

thus, it gives the best least square approximation of $\mathbf{W}$ by a matrix of rank $\mathbf{W}$.

Computing LSI Given a Document Term Matrix of dimension $D \times V$ representing documents in term space, LSI is obtained by choosing a rank k and computing the Truncated-SVD of $\mathbf{W}$. So that the quantity

$$\mathbf{W} \approx \mathbf{W}_k = \mathbf{U}_{D\times k}\mathbf{\Sigma}_{k\times k}(\mathbf{V}_{V\times k})' := \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k' \tag{3.5}$$

LSI has the advantage over other matrix factorisation algorithms in text mining (like Non-Negative Matrix Factorisation [Lee and Seung, 1999]) that columns of $\mathbf{U}_k$ provide the top k-basis for the column space of $\mathbf{W}$ while columns of $\mathbf{V}$ provide the top k-basis for its row space (see Figure 3.1). Rows of $\mathbf{U}_k$ and $\mathbf{V}_k$ relates documents and terms to topics, respectively. Therefore, right singular vectors show how much a topic is dominated by the words in the vocabulary. Likewise, left singular vectors map between topic and documents. The higher the weight the higher the predominance. Note that entries of such matrices need not to be positive, thus leaving the interpretation of negative signs unclear. Instead, the magnitude of each singular value represents the relative importance of the corresponding topic in the collection of documents.

[1] under some constraints

3.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a model for text data where documents are considered as mixtures of topics and each topic is defined as a probability distribution over the vocabulary. As LSI, the model assumes the number of topics to be specified before any data has been generated. The data generating process for each document is as follows:

1. Draw topics $\beta_k \sim Dir(\phi), k = 1, ..., K$
2. For each document
   (a) Draw topic proportions $\theta|\alpha \sim Dir(\alpha)$
   (b) for each word
      i. draw topic assignment $z_n|\theta \sim Mult(\theta)$
      ii. draw word $w_n|z_n; \beta \sim Mult(\beta_{z_n})$

Figure 3.3 illustrates this process as a graphical model. Rectangles are plate notation, and denote replication. Each node is a random variable and is labeled according to its role in the generative process. The hidden nodes for topics, topic proportions and assignments are unshaded. The shaded nodes correspond to observed data. The $\mathcal{N}$ plate denotes words within documents while the $\mathcal{D}$ plate denotes documents within the collection. Figure 3.4 illustrates its geometric interpretation for a vocabulary of size 3 and a 3 topic model. The word simplex represents the space of all possible probability distributions on the three words, so that each topic is represented as a point in this simplex. Consequently, being mixture of topics, document are represented as points in the topic simplex.

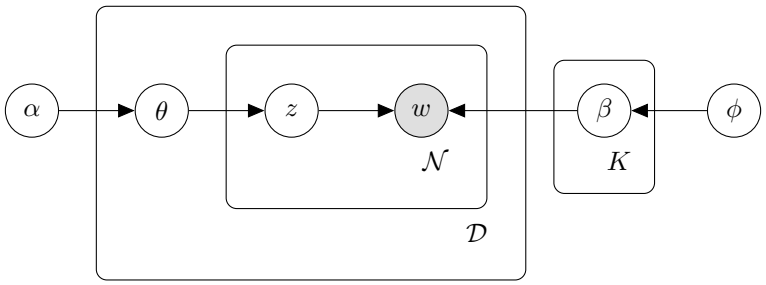


Figure 3.3: The graphical model for Latent Dirichlet Allocation Blei et al. [2003]

.

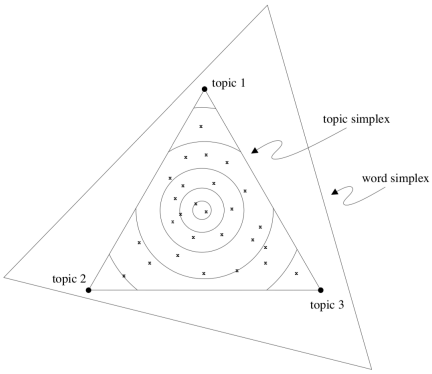


Figure 3.4: Geometric interpretation of LDA [Blei et al., 2003].

Therefore, for each topic $k = 1, ...K$, β_k is a distribution over the vocabulary representing word probabilities under topic k. The topic proportions for the d-th document are denoted by θ_d , where $\theta_{(d,k)}$ is the proportion of topic k in document d. The topic assignments for the d-th document are z_d, where $z_{(d,n)}$ is the topic assignment for the n-th word in document d. Finally, the observed words are w_d, where $w_{(d,n)}$ is the n-th word in document d.Under this model, the joint distribution of both the observed and the unobserved variables is

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{k=1}^{K} p(\beta_k) \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N_d} p(z_{d,n}|\theta_d) p(w_{d,n}|z_{d,n}, \beta_{1:K}) \tag{3.6}$$

where the : notation indicates the whole collection of random variables, for example $\beta_{1:K} = \{\beta_1, \ldots, \beta_K\}$. Inference for LDA involves computing the posterior distribution of the latent variables conditional on the observed word counts

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \tag{3.7}$$

It is worth noting from Equation 3.6 that LDA makes use of the "bag-of-words assumption" that the order of the words in documents can be neglected and that the order of the documents can be neglected as well.

Algorithms for an approximation of Equation 3.7 fall into two categories: sampling based algorithms [Steyvers and Griffiths, 2007] and Variational Methods [Blei et al., 2003, Wainwright et al., 2008]. Algorithm 2 sketches the Gibbs sampler for LDA assuming simmetric dirichlet priors with parameters α, ϕ. The interpretation is as

follows: the first factor in 3.8 expresses the probability of word w_i under topic j. It is approximately the ratio of the number of times the word w_i have been assigned to topic j ($n^{w_i}_{-i,j}$), over the sum of the word assignments, excluding the current instance, The second factor represents the probability of topic j in document d_i, where $n^{d_i}_{-i,j}$ denotes the number of times topic j have been assigned to some words in document d_i excluding the current instance.

In this study, we will use the Gibbs sampler proposed by Steyvers and Griffiths [2007] as we believe to provide more accurate approximations to the posterior. Variational Methods put an alternative family of distributions - called variational - over the posterior and then find the set of parameters that minimize the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior. In brief, instead of computing samples from the posterior distribution, say, $p(\theta|y)$, variational inference (VI) fixes a family of densities $\mathcal{Q}$ and finds the member q^* such that the KL divergence with the true posterior is minimized. Thus is equivalent to maximise the so called Evidence Lower Bound (ELBO) defined as:

$$\text{ELBO}(q) = \int_{\Theta} (\log p(\theta, y) - \log q(\theta)) d\theta$$

The variational algorithm underlying LDA builds on the so-called mean field variational family where it is further assumed that:

$$q(\theta) = \prod_i q(\theta_i)$$

This produces very fast estimates that scale for huge collections of data but loosing accuracy [Blei et al., 2017]. Indeed, the KL divergence is minimized when $q(\theta) = p(\theta|y)$.

Figure 3.5i and Figure 3.5ii illustrate these methods with a brief simulation study. In the simulation in Figure 3.5i we created a collection of 800 documents from vocabulary of $V = 300$ words with $K = 3$ latent topics. For each document, its length is drawn from a Pois(140). This setting is to better resemble the data that we shall be consider in the data analysis. Also, parameter[2] α is set equal to $1/K$ in order to have documents uniformly spread over the topic simplex. Eventually, as each topic is a mixture of words, we set $\phi < 1/V$ in order to have topics defined by few words with high probability and the remaining with a low one. It is well evident that both Gibbs sampling and VI underestimate the true proportions of

[2]we are sampling from a symmetric Dirichlet distribution

topics within each document (the θ_d-s in Figure 3.3). However, VI tends to spread these proportions too evenly among documents.

In Figure 3.5ii we generate another corpus such that ($D = 8000, V = 500$) and we estimated a LDA model using VI and Gibbs sampling. We notice that, in this setting, VI tends to be very inaccurate.

However, some studies, not directly connected to LDA, have shown that despite providing approximate posteriors, point estimates can still have good properties [Yao et al., 2018]. Extensions like Black-Box VI [Ranganath et al., 2014] show that in relatively short computational times, variational algorithms can provide better predictive densities that Gibbs sampling.

In the forthcoming applications we will use Gibbs sampling since the size of the data that we are trying to model allows to have results in an affordable time window.

Algorithm 2: Gibbs sampling for LDA [Steyvers and Griffiths, 2007]

---

1. For each word in vocabulary
   (a) for each topic
      i. sample topic assigment using

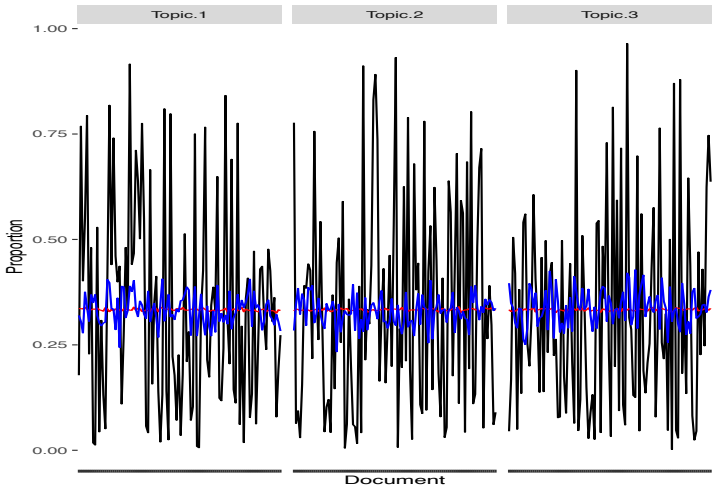
$$p(z_i = j|z_{-i}, w) \propto \frac{n_{-i,j}^{w_i} + \eta}{n_{-i,j}^{\cdot} + V\eta} \times \frac{n_{-i,j}^{d_i} + \alpha}{n_{-i,\cdot}^{d_i} + K\alpha} \tag{3.8}$$

   (b) compute word topic probabilities

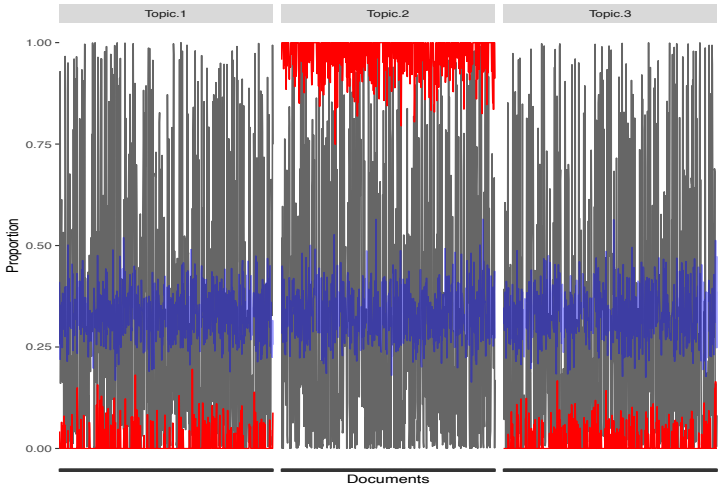
$$\hat{\beta}_j^w = \frac{n_j^w + \eta}{n_j^{(\cdot)} + V\eta} \tag{3.9}$$

   (c) compute document topic proportions

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_{\cdot}^{(d)} + K\alpha} \tag{3.10}$$



(i)



(ii)

Figure 3.5: Black line: True Proportions. Blue line: proportions with Gibbs sampling. Red line: proportions under VI.

3.3 Supervised Topic Modeling via Partial Least Squares

Both LSI and LDA can be regarded as *unsupervised* methodologies. Broadly speaking, this means that while one has data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i$ is the usual vector of features or covariates and y_i is an outcome, they use *only* the set $\boldsymbol{x}_i, i = 1...n$. Conversely, a methodology that makes use of response y to infer some traits about the x_i-s is regarded as *supervised*. Partial Least Squares (PLS) is a supervised dimensionality reduction technique that seeks a linear combination of the input variables $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_V\}$ to find directions in the data with larger variation that are most correlated with the response y. To briefly sketch the algorithm, suppose that covariates $\boldsymbol{x}_j, j = 1, .., V$ have been centered, the first PLS direction is $\boldsymbol{z}_1 = \sum_v \langle \boldsymbol{x}_v, \boldsymbol{y} \rangle \boldsymbol{x}_v$ (where $\langle . \rangle$ denotes inner product) . The response vector $\boldsymbol{y}$ is then regressed on $\boldsymbol{z}_1$ giving coefficient estimate $\hat{\beta}_1^{PLS}$. The remaining covariates are orthogonalized with respect to $\boldsymbol{z}_1$ and the process is repeated until $K \leq V$ directions are obtained. Typically, the number of PLS direction is chosen by cross-validation (see Section 3.8). This is described in more details in Algorithm 3 (Friedman et al. [2001]). The m-th PLS direction $\hat{\psi}_m$ solves the following optimization problem:

$$\begin{aligned} &\max_{\alpha} \text{Corr}^2(\boldsymbol{y}, \mathbf{X}\boldsymbol{\alpha}) \text{Var}(\mathbf{X}\boldsymbol{\alpha}) \\ &\text{subject to } \|\boldsymbol{\alpha}\| = 1, \quad \boldsymbol{\alpha}' \mathbf{S} \hat{\boldsymbol{\psi}}_l = 0, \quad l = 1, ...m-1 \end{aligned} \tag{3.11}$$

where $\mathbf{S}$ is the sample covariance matrix. Thus, PLS in performing a dimensionality reduction of the original data matrix driven by a response variable. Suppose that we have availability of a collection of documents and some quantitative variables associated with each.

In order to make explicit the connection of PLS, text data and topic modeling, suppose that some text has been processed so as to obtain its vector space representation in the DTM matrix and let $\{(p_d, \boldsymbol{w}_d)_{d=1}^D\}$ be the data to model. For example, in this study p_d is the price for a product sold in an e-commerce website and $\boldsymbol{w}_d$ is the vector space representation of the description that is provided by the vendor. If we applied PLS to this data then what we obtain is a topic model that uses the information contained in the response to drive the discovery of the topics. This is more evident if we established the connection existing between PLS, Principal components (PC) and LSI. If we omitted the Corr term in 3.11 then

the resulting minimization would be Principal Components. To establish the connection between PC and LSI, an alternative way to compute principal components is to use the truncated singular value decomposition of the data matrix after that it has been centered, then, principal components are then obtained by projection $\mathbf{X}\mathbf{V}_k = \mathbf{U}_k\mathbf{\Sigma}_k$.

Algorithm 3: Partial Least Squares

1. standardize each x_j such that $\sum_i x_{ij} = 0$ and $\|x_j\|_2 = 1$. Set $\hat{y}^0 = \bar{y}1$ and $x_j^0 = x_j, j = 1, ...p$
2. for $m = 1, 2, ...p$
   (a) $z_m = \sum_j \hat{\psi}_{mj} x_j^{(m-1)}$, where $\hat{\psi}_{mj} = \langle x_j^{(m-1)}, y\rangle$
   (b) $\hat{\theta}_m = \langle z_m, y\rangle / \langle z_m, z_m\rangle$
   (c) $\hat{y}^{(m)} = y^{(m-1)} + \hat{\theta}_m z_m$
   (d) Orthogonalize each $x_j^{(m-1)}$ w.r.t. z_m:
   $x_j^{(m)} = x_j^{(m-1)} - [\langle z_m, x_j^{(m-1)}\rangle / \langle z_m, z_m\rangle] z_m, j = 1, ..., p$
3. Output the sequence of fitted vectors $\{\hat{y}^{(m)}\}_1^p$. Since the $\{z_l\}_1^m$ are linear in the original x_j, so is $y^{(m)} = X\hat{\beta}^{\text{pls}}(m)$. These linear coefficients can be recovered from the sequence of PLS transformations.

3.4 Regression models

3.4.1 OLS

Let us consider the hedonic model introduced in 2.13, which we reformulate in matrix notation as $\boldsymbol{p} = \mathbf{W}\boldsymbol{\beta} + \boldsymbol{\epsilon}$. The first estimator that we may consider for this model is - of course - the OLS estimator

$$\hat{\boldsymbol{\beta}}^{ols} = \text{argmin}\|\boldsymbol{p} - \mathbf{W}\boldsymbol{\beta}\|_2^2.$$

Therefore, the hedonic value of the description - and thus the estimate of the price for the i-th item - is given by

$$\text{[OLS]: } \hat{p}_i = \boldsymbol{w}_i'\hat{\boldsymbol{\beta}}^{ols} \tag{3.12}$$

However, the dimensionality of the model depends heavily on the size of the vocabulary and usually a large number of features can be extracted from just few lines of text. Proper pre-processing can help to reduce the size of the vocabulary - and consequently the set of features that can be extracted - but even in those cases where $V < D$, OLS estimation can be undesired especially when the goal is to use the hedonic model for prediction [Monson, 2009]. In fact, if the model is too complex and overfits the data, predictions using OLS estimates are approximately unbiased but they will suffer from large variance. This is illustrated with a simple simulation in Figure 3.6 where we simulated data from the model $y = x^2 + \epsilon, \epsilon \sim N(0, 1)$. It is clear that the more variables we add to the model using higher order polynomials, the more the bias reduces at a price of an increasing variance. In some situations, the size of the vocabulary may exceed the available sample size, especially if one considers richer representation of documents (as n-grams) other than that provided by words. If $V > D$, then Ordinary Least Squares (OLS) would be unfeasible since the solution to the normal equations

$$(\mathbf{W}'\mathbf{W})\boldsymbol{\beta} = \mathbf{W}'\boldsymbol{y} \tag{3.13}$$

would not be unique. In fact, if $V > D$ then $rank(\mathbf{W}'\mathbf{W}) \leq \min(D, V) \leq D$, so there exists a vector in the nontrivial nullspace $\mathcal{N}$ of $\mathbf{W}'\mathbf{W}$, $\boldsymbol{v} \in \mathcal{N}(\mathbf{W}'\mathbf{W})$ - such that if $\boldsymbol{\beta}^*$ is a solution of the normal equations then $\boldsymbol{\beta}^* + \boldsymbol{v}$ would also be a solution.

3.5 Sparse models

Another issue in regression with text data that was raised in Section 2.6 is that the most common procedure for variable selection (like best subset selection) cannot be applied because the number of variables is almost always too large. Thus, we need to rely on statistical methodologies based on constrained estimators which are computationally less expensive and provide sparse estimates of the vector of regression coefficients. The advantage of using a constraint based approach is that these estimators will work also in high-dimensional scenarios - when $V > D$ - thus reducing the importance of pre-processing text.

In sparse modeling it is assumed that the true coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^V$ is sparse, meaning that it contains $k < V$ non-zero coefficients with support denoted by

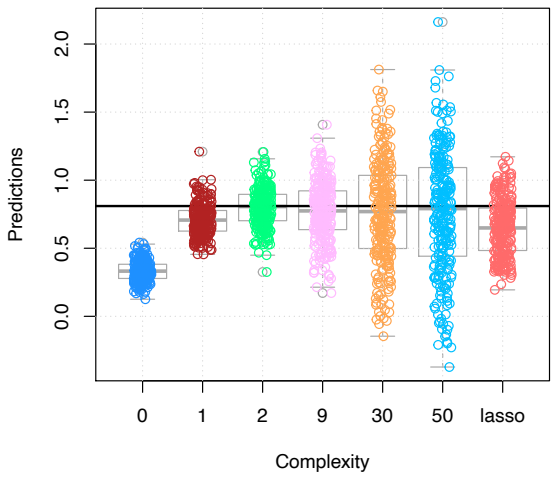


Figure 3.6: Prediction error at test point $x_0 = 0.9$.

$$\mathcal{S} = \text{support}(\boldsymbol{\beta}) \subset \{1, ..., V\} \tag{3.14}$$

Many statistical learning algorithms have been developed to obtain sparse coefficients estimates, see as example [Tibshirani, 1996, Zou and Hastie, 2005, Bogdan et al., 2015, Bondell and Reich, 2008]. These methods minimize the residual sum of squares plus a penalty term - also called the regularization term - which causes the solution to be sparse. The coefficient vector that is obtained is such that non-zero entries will hopefully correspond to words that resemble important attributes of prices. Of course, perfect variable selection is achievable under some conditions and assumptions - which we will briefly introduce - that are specific to each method. An additional advantage of penalty-based methods is that of improving predictive accuracy by reducing overfitting. This is accomplished by introducing a source of bias which is compensated by a reduction of the variance that leads to a lower mean squared prediction error. This is known as bias-variance tradeoff [Friedman et al., 2001] and was illustrated in Figure 3.6. In the figure, a model fit with 50 degree polynomial at test point $x_0 = 0.9$ yields MSE=0.2 which becomes 0.07 when a Lasso-type penalization is added. Remarkably, predictions made with the Lasso are clearly biased but show much less variance.

3.5.1 The Lasso

The Lasso [Tibshirani, 1996] is probably the most known shrinkage estimator that produces sparse coefficient estimates in regression problems. Let us consider the hedonic regression model in Equation 2.13. The Lasso solution is

$$\hat{\boldsymbol{\beta}}^{\text{Lasso}} \in \arg\min_{\beta} \sum_{i=1}^{D} \left(p_i - \beta_0 - \sum_{j=1}^{V} w_{ij}\beta_j \right)^2 \text{ subject to } \sum_{i=1}^{V} |\beta_j| \leq t \tag{3.15}$$

which can be rewritten in lagrangian or penalized form as

$$\hat{\boldsymbol{\beta}}^{\text{Lasso}} \in \arg\min_{\beta} \sum_{i=1}^{D} \left(p_i - \beta_0 - \sum_{j=1}^{V} w_{ij}\beta_j \right)^2 + \lambda \sum_{i=1}^{V} |\beta_j| \tag{3.16}$$

It can be shown that the two sets of solutions are equivalent for some specific values of λ and t. It can be noticed by looking at 3.15 that the Lasso is minimizing the same residual sum of squares as OLS subject to a specific constraint given by some ℓ_1 norm cone. This is why the Lasso can be seen as biased estimator. If the solution to the unconstrained problem is the same as is the constrained, meaning that the constraint is inactive at the optimal point, then both the Lasso and OLS will have the same solution. On the contrary, if the constraint is active then the solution would lie somewhere on the boundary of the constraint set. The shape of the constraint set, the ℓ_1 diamond, causes the solution to be sparse. Although there is no closed form solution for the Lasso in general, it can be reported as

$$\hat{\boldsymbol{\beta}}^{Lasso} = \boldsymbol{S}_{\lambda}(\mathbf{W}'(\mathbf{p} - \mathbf{W}\boldsymbol{\beta})) \tag{3.17}$$

Where $\boldsymbol{S}_{\lambda}$ is a soft-thresholding operator and λ is a tuning parameter that is chosen using cross-validation by picking the value that attains the lowest prediction error. The higher the value of the tuning parameter λ the higher will be the weight put on the penalization term in Equation 3.16 so that the solution will converge to 0. Conversely, for small lambdas, little effect is given to the penalization so that if $\lambda = 0$ the OLS fit is obtained. This means that as long as the value of λ varies we obtain different estimates of $\hat{\boldsymbol{\beta}}^{Lasso} \equiv \hat{\boldsymbol{\beta}}^{Lasso}(\lambda)$. This is called the Lasso path of solutions and several algorithms have been proposed to retrieve it. In what

follows we will use the Least Angle Regression (LARS) algorithm by Tibshirani et al. [2004] which is illustrated in Algorithm 4.

Algorithm 4: Least Angle Regression (Lasso Modification)

1. standardize predictiors such that $\sum_i w_{ij} = 0$ and $\|w_j\|_2 = 1$. Start with the residual $\mathbf{r} = \mathbf{p} - \bar{\mathbf{p}}$, $\beta_1, \beta_2, ..., \beta_p = 0$.
2. Find the predictor $\boldsymbol{w}_j$ most correlated with $\mathbf{r}$
3. move β_j from 0 towards its least square coefficient $\langle \boldsymbol{w}_j, \mathbf{r} \rangle$, until some other competitor $\boldsymbol{w}_k$ has much correlation with the current residual as does $\boldsymbol{w}_j$.
   (a) if a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction
4. move β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\boldsymbol{w}_j, \boldsymbol{w}_k)$, until some other competitor $\boldsymbol{w}_l$ has much correlation with the current residual
5. Continue in this way until all V predictors have been entered.

The fit for a Lasso model is given by

$$\text{[LASSO]: } \hat{p}_i = \boldsymbol{w}_i' \hat{\boldsymbol{\beta}}(\lambda_{cv}) \tag{3.18}$$

Where $\hat{\boldsymbol{\beta}}(\lambda_{cv})$ is a sparse vector tuned with cross-validation (see Section 3.8).

The variable selection properties of the Lasso are illustrated in [Tibshirani et al., 2015]. In brief, they rely on a *irrepresentability* condition that requires the variables belonging to the complement of the set $\mathcal{S}$ to be well separated[3] from the columns of the data matrix whose columns are in $\mathcal{S}$.

3.5.2 Sorted L-One Penalized Estimation

To increase the accuracy of the estimation of large signals and to eliminate some false discoveries, the adaptive and re-weighted

[3] Ideally, if these were orthogonal then this condition will be satisfied.

versions of the Lasso were introduced in Candes et al. [2008] and Zou [2006]. In these procedures the smoothing parameters are adjusted to the unknown signal magnitudes based on some previous estimates of coefficients. The idea is to take penalties inversely proportional to the estimated magnitudes so that large regression coefficients suffer from less shrinkage than small ones. In some situations, this selection of variables outperforms that provided by the Lasso [Zou, 2006].

The idea behind Sorted L-One Penalized Estimation (SLOPE) [Bogdan et al., 2015] is different in the sense that while in the adaptive Lasso the penalty tends to decrease, in SLOPE the exact opposite happens. This is because SLOPE tries to adapt to the signal sparsity by controlling the False Discovery Rate (FDR)[Benjamini and Hochberg, 1995] at some nominal level α. Indeed, the simulations carried out in Bogdan et al. [2015] show that the Lasso has no control over FDR. In fact, tuning the value of λ by cross-validation has revealed to select too many variables thus including many false discoveries along the Lasso path [Su et al., 2017]. For this reason, SLOPE is somehow preferable if one looks for a variable selection procedure that hinges more in favour of controlling false discoveries - as may be the case of hedonic modeling - other than on finding the best predictions.

SLOPE is the solution to the following convex optimization problem

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{p} - \mathbf{W}\boldsymbol{\beta}\|_2^2 + \lambda_1|\beta|_{(1)} + \lambda_2|\beta|_{(2)} + \ldots + \lambda_V|\beta|_{(V)} \tag{3.19}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_V$ and $\beta_{(1)} \geq |\beta|_{(2)} \geq \ldots \geq |\beta_{(V)}$. The sequence of λ values is selected according to different scenarios depending on whether there exists some correlations between regressors and whether the variance is known or unknown. In general, they are all based on some modification of the Benjamini-Hochberg sequence $\lambda_{BH}(i) := \Phi^{-1}(1 - q_i), q_i = iq/2V, q \in (0, 1)$ to control the false discovery rate.

The solution of the problem in 3.19 is obtained via proximal gradient methods with variations that allows for correlation in predictors and variance of the error term. A simple proximal algorithm is reported in Algorithm 5 where $J_\lambda(\boldsymbol{\beta}) = \lambda_1|\beta|_{(1)} + \lambda_2|\beta|_{(2)} + \ldots + \lambda_V|\beta|_{(V)}$ and the prox can be calculated either via standard Quadratic Programming or following the algorithm proposed by the authors. The variation of the algorithm for correlated predictors and unknown variance is sketched in Algorithm 6. For further details and

algorithms on the implementation of slope we refer the reader to Bogdan et al. [2015].

The fit that is obtained with SLOPE is

$$\text{[SLOPE]: } \hat{p}_i = \boldsymbol{w}_i' \hat{\boldsymbol{\beta}}^{slope} \tag{3.20}$$

where $\hat{\boldsymbol{\beta}}^{slope}$ is a sparse coefficient allowing for false discoveries.

Algorithm 5: Proximal gradient algorithm for SLOPE 3.19

1. Require $\boldsymbol{\beta}^0 \in \mathbb{R}^V$
   (a) for $k = 0, 1, ...$ do
   (b) $\boldsymbol{\beta}^{k+1} = \text{prox}_{J_\lambda}(\boldsymbol{\beta}^k - t_k \mathbf{W}'(\mathbf{W}\boldsymbol{\beta}^k - \boldsymbol{p}))$
   (c) end for

Algorithm 6: Iterative SLOPE fitting when σ is unknown

1. Input: $\mathbf{p}, \mathbf{W}$ and initial sequence λ^S.
2. Initialize: $S_+ = \emptyset$
3. Repeat:
   (a) $S = S_+$
   (b) compute the RSS obtained by regressing $\boldsymbol{p}$ onto variables in S
   (c) set $\hat{\sigma} = RSS/(D - |S| - 1)$
   (d) compute the solution $\hat{\boldsymbol{\beta}}$ to SLOPE with parameter sequence $\hat{\sigma}\lambda^S$
   (e) set $S_+ = \text{supp}(\hat{\beta})$
4. Until $S_+ = S$.

3.6 Aggregated Predictors

In a study on gene expression data, Park et al. [2007] prove that, under some conditions on the sample covariance structure of pre-

dictors, identifying and consolidating predictors into groups can yield a lower prediction error compared to an OLS model with no grouping. This results is given in the following theorem

Theorem 2 *Let $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m$ be columns of design matrix $\boldsymbol{X}$ with sample covariance structure $cor(\boldsymbol{x}_i, \boldsymbol{x}_j) = \rho > 0$ for $j \neq k$. Suppose that $y_i = \sum_m \beta_m x_{im} + \epsilon_i$ where ϵ_i are i.i.d with mean 0 and variance σ^2. Without loss of generality assume that covariates are standardized so that $\mathbf{X}'\mathbf{X}$ has 1 on the diagonal and ρ elsewhere. Let*

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{y}.$$

Let $\hat{\boldsymbol{\beta}}^A$ be the OLS coefficient when response y is regressed onto the sum of the m predictors. $\tilde{\boldsymbol{\beta}}$ denotes the corresponding vector of estimates for the original predictors

$$\tilde{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}^A, ..., \hat{\boldsymbol{\beta}}^A)',$$

where

$$\hat{\boldsymbol{\beta}}^A = \frac{\sum_i x_{.i} y_i}{\sum_i x_{.i}^2} \quad \textit{and} \quad x_{.i} = \sum_j x_{ji}$$

then

$$\mathbb{E}_{y|x}[\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2] < \mathbb{E}_{y|x}[\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2]$$

if and only if

$$\rho > 1 - \frac{\sigma^2}{\sum_m (\beta_m - \bar{\beta})^2/(m-1)}, \ \textit{where } \bar{\beta} = \sum_m \beta_m/m$$

The theorem claims that if the true coefficients of the predictors are similar, so that the ratio $\sum_m (\beta_m - \bar{\beta})^2/\sigma^2$ is small, then the range of ρ to improve the fit by averaging is large. This is shown in Figure 3.7.

Although $\tilde{\boldsymbol{\beta}}$ yields a larger bias than $\hat{\beta}$, the former is more accurate due to a lower variance. The authors provide a two step procedure to identify significant group of genes to predict a quantitative outcome and refer to this groups as *supergenes*. This procedure is reported in Algorithm 7,

The similarities existing between dealing with gene expressions and text data have been discussed for example in [Lee and Seung, 1999]. We begin by considering again the model in 2.13

$$p_i = \beta_0 + \beta_1 w_{i1} + \cdots + \beta_V w_{iV} + \epsilon_i.$$

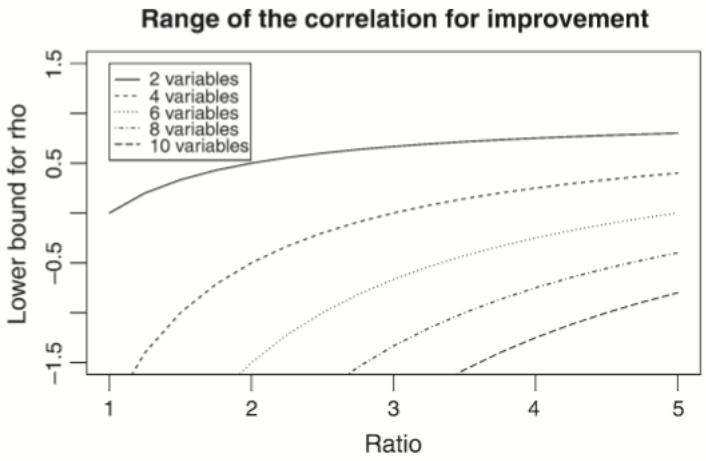


Figure 3.7: From [Park et al., 2007]. If the true coefficients of the predictors are similar,$\sum_m(\beta_m - \bar{\beta})^2/\sigma^2$ is small and the range of ρ to improve the fit is large. Improvements are expected in the upper left region

Suppose that there exists some words that have the same marginal effect on price. For example, we may suppose that the marginal effect of synthetic materials is approximately the same regardless of these being *elastan* or *polyester*. Let $(\mathcal{G}_i)_{i=1}^G$ be a partition of indices $\{1, 2, ..., V\}$, where there exists a one-to-one relation between this set and each word in the vocabulary $\mathcal{V}$. Then, the model can be re-formulated as

$$p_i = \beta_0 + \sum_{g=1}^{G} \beta_g \sum_{j=1}^{V} x_{ij} \mathbb{1}_{\{j \in \mathcal{G}_g\}} + \epsilon_i. \tag{3.21}$$

In this way, each covariate that belong to the same set $\mathcal{G}_g = \{j_1, ..., j_{|\mathcal{G}_g|}\}$ has the same coefficient β_g. This approach substitutes the original covariates with a new attribute set each being the sum of the weights of the words in the group. We refer to groups of words as *superwords*. The issue is how to select such groups. To this purpose we refer to same algorithm used in Park et al. [2007] but with a slight modification. We leverage the dual interpretation of the vector space of text, that is words in document space, to define the set of words to be clustered. This dual representation is given by the Term-Document Matrix. Also, we use cosine similarity instead of euclidean distance for clustering. Cosine similarity is widely used in text mining literature for measuring similarities between words for text clustering [Berry and Kogan, 2010],[Berry and Castellanos, 2004],[Aggarwal, 2018]. Cosine similarity between two word vectors

Algorithm 7: Hierarchical clustering and averaging for regression [Park et al., 2007]

1. Let $(y_i, \boldsymbol{x}_i)_{i=1}^n$ denote pairs of gene expression profiles ($\boldsymbol{x}_i \in \mathbb{R}^p$) and response variable
2. Apply hierarchical clustering of the genes to yield the nested correlation structure
3. At each level of hierarchy, create *supergenes* by averaging gene expressions inside each cluster. This gives p different sets of genes and supergenes that represent each level
4. For every set of predictors (genes and supergenes) fit Lasso using y as response variable
5. Using cross-validation (see 3.8) find the optimal degree of shrinkage and level of hierarchy

$\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^D$ is defined as $\boldsymbol{v}_1'\boldsymbol{v}_2/(\|\boldsymbol{v}_1\|\|\boldsymbol{v}_2\|)$. In this way we obtain a new design matrix $\mathbf{W}^G$ of dimension $D \times G$ whose columns are aggregated term weights. The vector $\boldsymbol{w}_i^G$ denotes to to the i-th row of this matrix. The fit for the aggregated predictors hedonic model is therefore given by

$$[\text{AP}_G]\text{: } \hat{p}_i = \boldsymbol{w}_i^G \hat{\boldsymbol{\beta}}_G^{ols} \tag{3.22}$$

Differently from [Park et al., 2007], given that we aim at identifying just few groups of words, we fit just simple OLS predictions.

3.7 Reduced regression

So far, the hedonic models that we discussed use the bag of words model to obtain the set of product attributes. As said, these models predict prices by estimating the hedonic value of the description that is provided to a product. Alternative models for text data like LSI and LDA represents documents as points in a lower dimensional latent space defined by the topics discovered in the collection. Using this lower dimensional representation, we can define a hedonic text regression models as

$$p_i = \beta_0 + \beta_1 z_{i1} + \beta_2 z_{i2} + \cdots + \beta_k z_{ik} + \epsilon_i \tag{3.23}$$

Algorithm 8: Aggregating predictors for grouped hedonic text regression

1. Let $(p_i, \boldsymbol{w}_i)_{i=1}^{D}$ denote pairs of (featurized) documents ($\boldsymbol{w}_i \in \mathbb{R}^V$) and prices
2. Use the dual representation of words in doc space given by the Term-document Matrix
3. Apply hierarchical clustering of terms and find the desired level of hierarchy to find words and group of words. This gives a partition $(\mathcal{G})_{i=1}^{G}$ with G groups.
4. Using this partition create *superwords* by summing word counts in the original DTM, to obtain a new DTM with G columns
5. Regress using p as response variable

where z_{ik} is the representation of a given description in its *latent* topic space that is estimated either by LSI or LDA. The hedonic values of the topic content of the descriptions - equivalently the predicted prices under 3.23 - by means of the methods illustrated in Sections 3.1-3.3 are the following

$$\text{[LSI]: } \hat{p}_i = \boldsymbol{u}_i' \hat{\boldsymbol{\beta}}^{ols} \tag{3.24}$$

$$\text{[LDA]: } \hat{p}_i = \boldsymbol{z}_i' \hat{\boldsymbol{\beta}}^{ols} \tag{3.25}$$

$$\text{[PLS]: } \hat{p}_i = \boldsymbol{w}_i' \hat{\boldsymbol{\beta}}^{pls} \tag{3.26}$$

There are two main reason to use such hedonic models. The first is practical: we want to predict the price of a given item based on the topic content of its description. Suppose that we wanted to predict the price of a t-shirt based on its description and that we estimated a $k = 3$ dimensional topic representation for that. Suppose also that we can interpret these topics as *quality of materials*, *appealing design* and *comfort*. To estimate the model in Equation 3.23, conditional on this estimated representation of documents, corresponds to estimate the marginal effect of each topic on prices. In this way, the model in 2.13 is simplified - the number of covariates is reduced from V to $k << V$ and its interpretation

may be improved. The second reason is technical, that is, to use a dense matrix of covariates instead of a sparse one.

In the case of LSI, we can think of the k-th topic $\boldsymbol{z}_k$ as a linear combination of the columns of $\mathbf{W}$ so that $\boldsymbol{z}_k = \sum_{i=1}^{V} \boldsymbol{w}_i v_{ik}$. This means that we can set $\mathbf{Z} = \mathbf{W}\mathbf{V}_k$, $\mathbf{Z} = \mathbf{W}\mathbf{V}_k\boldsymbol{\Sigma}_k$ or $\mathbf{Z} = \mathbf{U}_k$ [Foster et al., 2013] in 3.24. In LDA, a topic is not a linear combination of word columns but one sample from a V dimensional Dirichlet distribution. This means that each word is given higher or lower probability under the topic in the data generating process. To estimate an LDA model means to derive posterior probabilities of assignments for each topic given the observed DTM. Conditioning on a specific document, each word is assigned to a topic so that for each document i, we can collect into a matrix of counts $\mathbf{Z}$ of dimension $D \times k$ the number of words in documents that have been assigned to a specific topic. For what regards PLS instead the latent representation of supervised topics is learned naturally by Algorithm 3.

For all these models the number of latent topics k needs specification. For consistency with other methods, we treat it as a tuning parameter, therefore we look for the value of k in correspondence of the lowest cross-validation estimate of prediction error. This procedure is akin to that of principal components regression. Alternatively, we may choose a value of k after which the singular values in 3.5 become too small or some criterion like perplexity [Blei et al., 2003]. However, these criterions do not consider the information provided by prices but only that contained in the DTM. Given that the value of k may be large we add a Lasso penalty to the model in Equation 3.23 to allow for bias-variance tradeoff, so that we reformulate 3.24 and 3.25 to obtain the following fits

$$\text{[LSI-LASSO]: } \hat{p}_i = \boldsymbol{u}_i' \hat{\boldsymbol{\beta}}(\lambda_{cv}) \tag{3.27}$$

$$\text{[LDA-LASSO]: } \hat{p}_i = \boldsymbol{z}_d' \hat{\boldsymbol{\beta}}(\lambda_{cv}) \tag{3.28}$$

The rationale of this approach is as follows. Both LSI and LDA perform an unsupervised dimensionality reduction, so that not all the latent topics need to be significant predictors of the response, on the contrary they try to obtain the best approximation of the DTM. This is much more clear in LSI if we assume that only a subset of topics $\mathcal{K} \subset \{1, 2, ..., k\}$ is significant in predicting the response.

Consider the following minimization task

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{U}_k\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \tag{3.29}$$

The solution to 3.29 is unique since the objective is strictly convex. From first order optimality conditions we have

$$\boldsymbol{u}_j'(\boldsymbol{y} - \mathbf{U}_k\boldsymbol{\beta}) = \lambda s_j \quad j = 1, \ldots, K \tag{3.30}$$

Where s_j is a subgradient of the function $f(x) = \|x\|_1$ evaluated at β_j being equal to $\text{sign}(\beta_j)$ if $\beta_j \neq 0$ and some value in $[-1, 1]$ if $\beta_j = 0$. Given that $\mathbf{U}_k'\mathbf{U}_k = I$ we can obtain a closed form solution to solution to Equation 3.29 in terms of each coefficient so that

$$\hat{\beta}_j = \begin{cases} \boldsymbol{u}_j'\boldsymbol{y} + \lambda & \text{if } \boldsymbol{u}_j'\boldsymbol{y} < -\lambda \\ 0 & \text{if } |\boldsymbol{u}_j'\boldsymbol{y}| < \lambda \\ \boldsymbol{u}_j'\boldsymbol{y} - \lambda & \text{if } \boldsymbol{u}_j'\boldsymbol{y} > \lambda \end{cases} \tag{3.31}$$

That is, if the strength of linear dependence between $\boldsymbol{u}_j$ and $\boldsymbol{y}$ measured by its OLS coefficient $\hat{\boldsymbol{\beta}}^{OLS} = \boldsymbol{u}_j'\boldsymbol{y}$ exceeds the value of λ then the j-th topic is included in the model with shrunk coefficient. In order to select the model corresponding to best values of $\{\lambda, k\}$, we proceed via grid search by defining a grid of potential values of k, λ and applying for each the minimization as in Equation 3.29. Therefore, we obtain estimates as in Equation 3.31 and choose the model achieving the lowest estimate of prediction error. Thus:

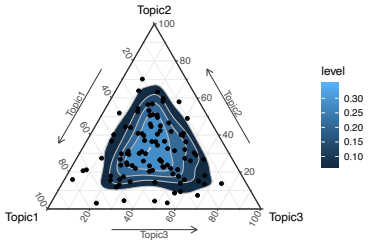
$$(k^*, \lambda^*) = \operatorname*{argmin}_{k} \operatorname*{argmin}_{\lambda} \text{CVErr}(\hat{f}_{\lambda,k}) \tag{3.32}$$

The same intuition holds also for LDA where the number of topics needs to be tuned accordingly to the values of hyperparameters $\{\alpha, \phi\}$. This can be done using the same supervised selection strategy used for LSI. Again, the goal is to look for some form of supervision to maximize predictive accuracy. Let $\mathcal{A}, \mathcal{B}$ denote respectively a set of values for LDA (hyper) parameters α, ϕ (assuming symmetric Dirichlet distributions). For each value of k, let $\mathcal{H} = \mathcal{A} \times \mathcal{B}$, then we look for

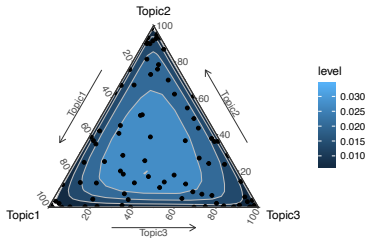
$$(\alpha^*, \phi^*, \lambda^*) = \operatorname*{argmin}_{h \in \mathcal{H}} \operatorname*{argmin}_{\lambda} \text{CVErr}(\hat{f}_{\lambda,\alpha,\phi}) \tag{3.33}$$

where the search for the best $h^* = (\alpha^*, \phi^*) \in \mathcal{H}$ is carried out via grid search. Tuning for different values of α and ϕ allows us to encourage different doc-topic and word-topic distributions. That is, we can encourage topics to be uniformly distributed across documents or we can encourage sparsity. In the same way we can

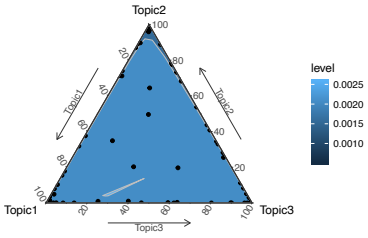
encourage a topic to be a sparse distribution over words, uniformly distributed or putting its mass around its expected value. Figure 3.8 explains this arguments by showing different samples from three symmetric Dirichlet distributions according to different values of parameter α for a $K = 3$ topic model.



(i) Dir(3)



(ii) Dir(1/3)



(iii) Dir(0.1)

Figure 3.8: Samples from different symmetric Dirichlet Distribution

3.8 Performance Evaluation

This section provides details on prediction error and its estimation. Given that the response variable for hedonic regression is continuous, the results reported are specific for continuous outcomes. Traditional hedonic regression modeling relies on goodness of fit to select the best functional form of the model [Cassel and Mendelsohn, 1985]. Results from statistical learning literature [Friedman et al., 2001] show that this measure is not the most appropriate for model selection when the number of covariates is large, as is the case of the model in 2.13. Therefore, in this work model selection

is performed by selecting the model that achieves the lowest prediction error for the price of new, unseen, observations based on a given set of covariates.

The generic setting is as follows. Suppose that the goal is to predict a continuous outcome y given some inputs $\boldsymbol{x}$. Suppose that, we estimated a predictive model $\hat{f}(\boldsymbol{x})$ based on some *training* data $\mathcal{F}$. Define a squared error loss function $L()$ as the loss that one has to pay each time y is predicted using $\hat{f}(\boldsymbol{x})$. Some widely used loss functions are the following [Friedman et al., 2001]:

$$L(y, \hat{f}(\boldsymbol{x})) = \begin{cases} (y - \hat{f}(\boldsymbol{x}))^2 & \text{squared error loss} \\ |y - \hat{f}(\boldsymbol{x})| & \text{absolute error} \end{cases} \tag{3.34}$$

Of course many other loss functions have been proposed in the literature. Some of these are more used in the field of engineering and computer science (see [Boyd et al., 2004] for more details). Given a loss function we can define the following quantities:

Definition 2 (Test error) *Test error is defined as the expected prediction error over an independent test sample* $(y, \boldsymbol{x})$ *drawn from some joint probability distribution* $p(y, \boldsymbol{x})$

$$Err_{\mathcal{F}} = \mathbb{E}[L(y, \hat{f}(x))|\mathcal{F}]$$

This leads to the definition of Prediction Error which is Test Error averaged over all possible training sets that can be drawn from $p(y, \boldsymbol{x})$

Definition 3 (Prediction Error)

$$Err = \mathbb{E}[L(y, \hat{f}(\boldsymbol{x}))] = \mathbb{E}[\mathbb{E}_{\mathcal{F}}]$$

Bias-Variance decomposition Assume the following model to hold for the population

$$y = f(\boldsymbol{x}) + \epsilon, \quad \mathbb{E}[\epsilon] = 0, \quad \text{Var}(\epsilon) = \sigma_\epsilon^2$$

Suppose that we want to assess predictive fit at input $\boldsymbol{x}_0$. Then

$$\begin{aligned} Err(x_0) &= \mathbb{E}_{\mathcal{F}}\mathbb{E}_{y|x_0}[L(y, \hat{f}(\boldsymbol{x}_0))] \\ &= \mathbb{E}_{y|x_0}[(y - f(\boldsymbol{x}_0))^2] + \mathbb{E}_{\mathcal{F}}[(f(\boldsymbol{x}_0) - \hat{f}(\boldsymbol{x}_0))^2] \\ &= \sigma_\epsilon^2 + \text{Bias}^2[\hat{f}(\boldsymbol{x}_0)] + \text{Var}[\hat{f}(\boldsymbol{x}_0)] \end{aligned} \tag{3.35}$$

This decomposition suggests two distinct facts. The first is that perfect prediction is not possible, even if one correctly specify the true model there always be a source of error σ_ϵ^2 which is *irreducible*. Second, biased estimators can be preferred to unbiased if their variance compensate for this loss. This is known as bias variance trade-off and it was illustrated in Figure 3.6.

Cross-Validation In this section we discuss the most common way to estimate exptected prediction error. A first guess would to estimate it using training data

$$\mathbb{E}_n[L(y, \hat{f}(\boldsymbol{x}))] = \mathbb{E}_n[(y - \hat{f}(\boldsymbol{x}))^2] \tag{3.36}$$

where $\mathbb{E}_n$ denotes expectation with respect to the empirical measure. Broadly speaking, this estimate can be regarded as being too optimistic since $\hat{f}$ is chosen so as to fit well to the data. Cross-validation is a widely used procedure that provides good estimates of the expected test error. There are two main versions of cross-validation: exhaustive cross-validation and non-exhaustive cross-validation. What is common to both approaches is the subdivision of the available data into a training and a test set. The training set is always used to learn/estimate a certain model while the test set is used to have an estimate of the test error. We first consider non-exhaustive cross-validation. Among the many variants of this methods, the most diffused is K-fold cross-validation. Here, The original sample is partitioned into almost equal sized K subsamples. Of these subsamples $K-1$ are retained to form the training set and the remaining to form the test set. This procedure is repeated K times with each of the K subsamples used once as test set. The estimate of prediction error is obtained by averaging the estimates obtained using each test set. Algorithm 9 illustrates how it works. Popular choices for K are $K = 5, 10, N$. The latter is called *leave one out cross-validation* because prediction is done just on one observation. The choice is usually arbitrary but some insights can be derived. For a small sample size, the choices $K = 5, 10$ yield estimates of prediction error being bit further - on average - from the true value. On the contrary the variance is reduced since these estimates $\hat{f}^{-k}$ are obtained using less overlapping data. This is because the variance of the sum of highly correlated quantities is larger than that of mildly correlated quantities. On the contrary, leave one out cross-validation usually provides better point estimates. Other variants of K-fold cross-validation including the *holdout* method or *monte-carlo* cross-validation and are discussed

in [Dubitzky et al., 2007, Arlot et al., 2010]. Exhaustive cross-validation [Celisse et al., 2014] involves using M observations as test set and the remaining as the training set. This is repeated on all ways to cut the original sample on a test set of M observations and a training set. This procedure is not particularly adopted as it is non-exhaustive cross-validation since it becomes computationally infeasible for moderate sample size. Remarkably for $M = 1$ this procedure and leave-one-out cross validation coincide.

Algorithm 9: K-Fold Cross-validation

1. Divide training set $\mathcal{F}$ into K folds $\{\mathcal{F}_1, ..., \mathcal{F}_k\}$ such that $\cup_k \mathcal{F}_k = \mathcal{F}$ and $\mathcal{F}_i \cap_{i \neq j} \mathcal{F}_j = \emptyset$
2. For each $k = 1, ..., K$
   (a) Estimate $\hat{f}^{-k}$ using $\mathcal{F} \backslash \mathcal{F}_k$
   (b) Compute
   $$\mathrm{CV}_k(\hat{f}^{-k}) = n_k^{-1} \sum_{i \in \mathcal{F}_k} (y_i - \hat{f}^{-k}(\boldsymbol{x}_i))^2$$
3. Obtain estimate
   $$\mathrm{CVErr}(\hat{f}) = K^{-1} \sum_{k=1}^{K} \mathrm{CV}_k(\hat{f}^{-k}) \tag{3.37}$$

Cross-validation is also an important tool for model selection. In fact, suppose that we estimated a certain number of different models. To select the best model - in terms of predictive accuracy - it is sufficient to apply cross-validation and select the one with the lowest estimate of prediction error. This holds also for selecting a model that depends on some tuning parameter as those that shall be proposed afterwards. Let $\{\lambda\}_{m=1}^{M}$ be a finite sequence of tuning parameters and let $\{\hat{f}_{\lambda_1}, ..., f_{\lambda_M}\}$ a sequence of models. Then we have

$$\lambda^* = \underset{\lambda \in \{\lambda_1, ..., \lambda_M\}}{\operatorname{argmin}} \mathrm{CVErr}(\hat{f}_\lambda) \tag{3.38}$$

and the selected model will be $\hat{f}^*_\lambda$. Alternatively, a *one standard error rule* [Friedman et al., 2001] is often used with cross-validation,

in which we choose the most parsimonious model whose error is no more than one standard error above the error of the best model.

CHAPTER

4

EMPIRICAL ANALYSIS

This chapter offers an application of the hedonic pricing models proposed in the previous chapter to fashion products. Before delving into the empirical analysis, it is appropriate to provide a brief description of the fashion industry that we consider for this application. Cachon and Swinney [2011] define four different systems of firms that operate in the fashion market, namely *traditional*, *enhanced design*, *quick response* and *fast-fashion*. The distinction is straightforward; traditional firms are characterized by long production lead times and standard product design abilities. This system closely resemble a newsvendor model. An Enhanced Design (ED) system employs enhanced design capabilities - thus a greater consumers' willingness to pay - but maintains long production lead times. Enhanced Design focuses on product design while avoiding the kind of radical supply chain overall necessary to achieve lead time reduction. A quick response system does not employ enhanced design capabilities, but does yield significantly reduced production lead times. Eventually, Fast-Fashion systems employ both quick response and enhanced design capabilities. For this application, we collected public available data from the Italian websites of five fashion brands, namely Zara (Z), H&M, Pinko (P), Patrizia Pepe (PP)

and Elisabetta Franchi (EF)[1]. For each retailer, we developed an apposite crawler to scrape the whole apparel section of the website. The crawling routine started from October to January 2016 on a weekly basis. This choice has been made for two practical reasons. The first was to avoid the discount period when due to special offers the effect of the attributes on prices is not the real one. In fact, our interest is on the level of prices during the regular winter-autumn season. The second one is that after this period new collections come out (spring, summer..) so that data would not be coherent for hedonic pricing as the set of attributes that determines prices completely changes. We decided to work on two sub-categories: trousers and dresses collections for women.

4.1 Preliminary data processing

In Chapter 2, we outlined that pre-processing is usually useful to have a proper vector space representation of text data. Here, we applied some of the procedures described therein. Preliminarily, we tidied-up and cleaned the data. As first step, we merged the data from each website to form a whole data-set. In addition, we enriched the product descriptions by attaching the name of the item - like *white t-shirt* - in order to use as much text data that we had at disposal. Afterwards, we proceeded by removing duplicated records. As we saw in Chapter 3, hedonic models are usually specified in terms of a dummy time variable. While we collected data on a weekly basis we observed that prices in websites remained unchanged and that items followed a sort of lifecycle. Sometimes items were dropped and other came out but the prices remained constant for the whole season (excluding the sales period). Since each crawling routine makes a copy of the whole website, the resulting data set contains severe overlaps of data. Thus, we discarded records that had the same price and description, keeping just one record for each.

The pre-processing steps that we adopted are as follows. First, we removed punctuation and substituted accented letters with non-accented. Also, we removed special characters and numbers. We removed the latter since we found out that they refer to details about the models (i.e. *...the model is 160cm height...*, or *...the model is wearing size 40...* etc..). The second step was to remove stopwords.

[1]These brands are at the same time both producers and retailers. Therefore, the terms brand, retailer and producers can be considered synonyms.

To this purpose we use the italian list of stopwords provided inside the rpackage **tm** [Meyer et al., 2008]. We added to this list words like *model*, *size*, *cm*, *wears* referring again to the model and not providing any detail on the products. The final step was stemming, which is nothing else but reducing each word to its root. This step need not to produce meaningful words in general. For this step we again used the **tm** R package which builds over the Snowball stemmer.

Table 4.1 reports the number of observations and the number of features from text after we went through the pre-processing step.

Table 4.1: Data after pre-processing

Category	Sample size	Number of features
Trousers	692	302
Dresses	814	309

We are not left with much sample size. While this is somehow obvious since we are bounded by the effective stock of items showed by the websites for a given collection, to keep on scraping data would not have been a good choice for the reasons illustrated above; sales period approaching and new collections about to come. However, the methods that will be using are in some sense robust to limited sample size. As example, the Lasso and SLOPE are designed to work also when the number of covariates is much greater than the sample size so that the ratio of number of variables over sample size never constitute a serious issue.

4.2 Results

The results for the Lasso and SLOPE are obtained using R packages **glmnet** [Hastie and Qian, 2014] and **slope** [Bogdan et al., 2015], respectively. To run LDA we used the R-package **text2vec**. The presentation of the results is separated for each of the two categories that we consider.

4.2.1 Trousers

Descriptive statistics It is well evident from Figure 4.1 that two couples of fashion brands, namely, (ZZ and HM) and (P,PP) have similar, nearly overlapping, price distributions. On the contrary

EF looks like a stand alone reality. In fact, the mean price for ZZ and HM are respectively 33.92 and 34.41 (median respectively 29.95 and 34.99). The mean price for P and PP are respectively 188.81 and 182.06 (median 185 and 182 respectively) while the mean price for EF is 250.7. Further details on the distribution of prices is reported in Table 4.2. Kernel density estimates in Figure 4.1 show that both ZZ and HM have peaked distributions. This is because their pricing strategy is to concentrate prices in classes rather than spreading over a certain range.

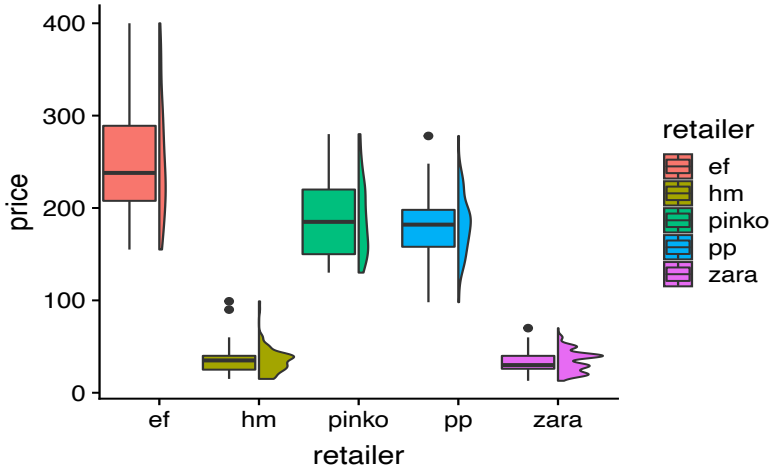


Figure 4.1: Distribution of prices of trousers.

Table 4.2: Trousers. price distribution descriptive statistics

Brand	n	Min	1st Q.	Mean	Med.	3rd Q.	Max
EF	58	155	207.75	250.65	238	289	400
HM	114	14.99	24.99	34.41	34.99	39.99	99
P	76	130	150	188.81	185	220	280
PP	65	98	158	182.06	182	198	278
ZZ	379	12.95	25.95	33.92	29.95	39.95	69.95

Figure 4.2 provides a graphical view of the Document-Term matrix obtained from the descriptions of trousers. It is well evident that counts are sparse and most features are present just once across documents. The *wordcloud* in Figure 4.3 is a graphical representation that reports the most frequent words in the trousers' descriptions. The size of the font is directly proportional to the frequency

of the word. The most frequent words are associated with the fit, like tight (*aderent*), low waist (*bass*) and the design: classic and bell-shaped (*campan*). For each retailer, Figure 4.4 displays the distribution of word counts that are used to describe each item. We used regular expressions to split each string in correspondence of white-spaces between words. Few differences outstand between brands: for E, H and PP the distributions are very similar while Z is the one that shows greater variability. On average approximately 146 words are used to form a description of an item.

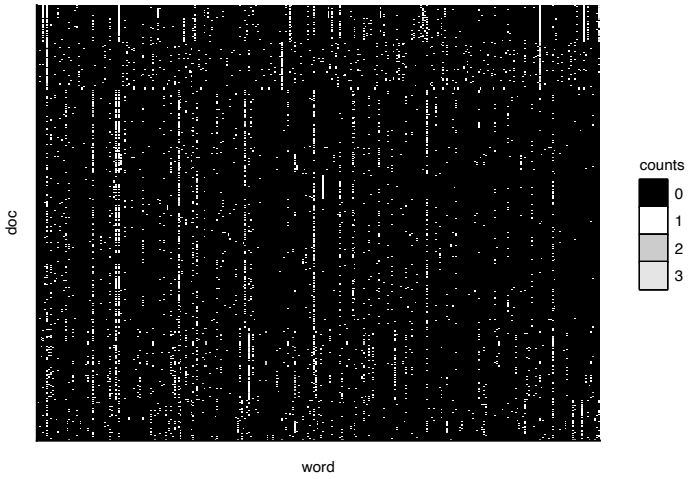


Figure 4.2: Document Term Matrix for descriptions of trousers

Figure 4.3: Worcloud of the most common words in the collection of trousers' descriptions

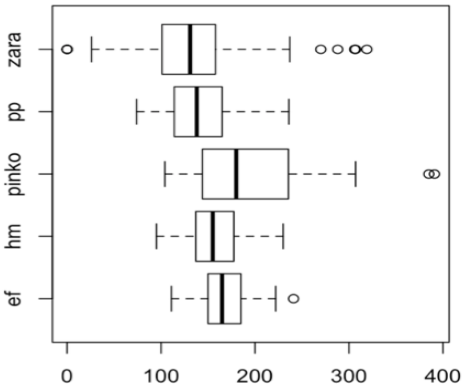


Figure 4.4: Conditional word count distribution

Figure 4.5 reports the pairwise correlations between word frequencies in the DTM. It can be noticed that while there are some groups of words that exhibit high correlations, these are very low in general.

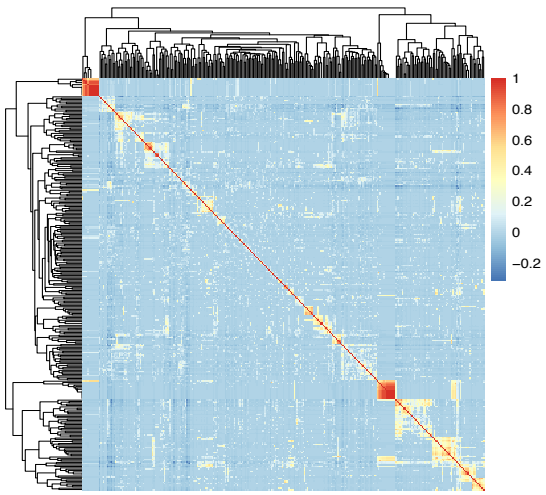


Figure 4.5: Correlation matrix of terms in trousers DTM

Hedonic Text Regression Modeling While most variables come with a natural measurement scale, the "measure" of a text feature depends on the weighting scheme that has been used to obtain the vector space representation. The most common ways to define weights w_{ij}-s were introduced in Chapter 2. To the purpose of this analysis, we believe term presence and term frequency to be the

most appropriate schemes to weight text features in this context. For example, if we used term presence, the coefficients would be interpreted as the shift in the average level of price associated with the presence of the word in the description. In the example given in Figure 2.1i, the value of the coefficient related to the word *silk* would give the average price difference of apparels containing silk, all else equal. If we used term frequency, each coefficient would give the marginal effect on prices for an additional count of the word in the description. This could emphatize the importance of the attribute given by the producers. The more the counts the more the importance of the word-attribute in the description. In this application - and in general - when the documents are short, there is little difference between the two as most words are used just once. On the contrary, the marginal effect on prices due to variation in tf-idf has not a straightforward interpretation. Therefore, if we used this scheme parameter estimates would not correspond to shadow prices.

Table 4.3 reports our results for in-sample goodness of fit and predictive accuracy for each of the model that were introduced in Chapter 3. The values of adjusted R^2 and RSE are defined respectively as $1-[(1-R^2)(n-1)/(n-k-1)]$ and $\sum_i (y_i - \hat{y}_i)^2 / \sum_i (y_i - \bar{y})^2$. We will refer to RMSE_{cv} as the estimate of Prediction Error obtained under a squared loss function using cross-validation. Additionally, we will refer to MAE (Mean Absolute Error) as the estimate of prediction error based on a loss function measuring the absolute value of deviation. In the first part of the table, the *base* fit has to be intended as $\hat{p}_i = \bar{p} = n^{-1} \sum_i p_i$ while it is the least squares fit using brand fixed effects for the brand case - taking EF as a baseline - in the second part.

Clearly, the brand is an important determinant of prices. Cross-Validation shows that this model provides also a good out-of-sample predictive accuracy with $\text{RMSE}_{cv} = 26.707$. The hedonic text regression model yields adjusted-$R^2 = 0.947$. It is interesting to observe that OLS attains the highest value of adjusted R^2 while providing the worst predictive accuracy among all the proposed fits. This result is likely due to the fact that we are fitting a model with a high number of covariates and that we are overfitting the data. It can be noticed from first part of Table 4.3 that while text features enhances predictive power from BASE to OLS they deteriorates it if combined with the brand. A F-test ($F = 7.567 > F^{0.05}(301, 386)$ rejects the hypothesis that the effect of the words is insignificant at level $\alpha = 0.05$.

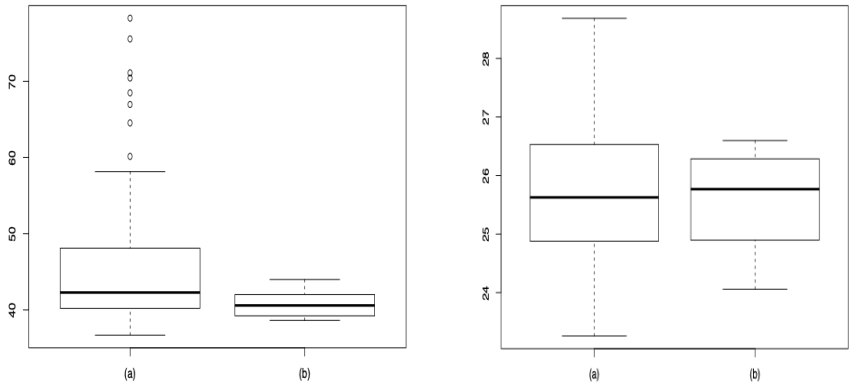
Table 4.3: Results of hedonic text pricing models for trousers category

	No brand				
Fit	nvar*	adj-R^2	$RMSE_{cv}$	RSE	MAE_{cv}
BASE	1	-	83.339	1	70.069
Text					
OLS	302	0.947	70.056	0.701	30.502
Sparse					
LASSO	82	0.901	35.428	0.181	21.813
SLOPE	85	0.913	34.193	0.168	-
Dim.red					
LSI-LASSO	96	0.884	32.167	0.149	22.139
LDA-LASSO	58	0.855	33.529	0.162	22.160
PLS	4	0.902	27.629	0.110	19.127
Aggregated					
AP_{10}	10	0.698	46.753	0.315	30.544
AP_{20}	20	0.793	39.212	0.221	25.721
	Brand				
Fit	nvar*	adj-R^2	$RMSE_{cv}$	RSE	MAE_{cv}
BASE	5	0.897	26.707	0.103	70.069
Text					
OLS	306	0.947	60.205	0.552	34.970
Sparse					
LASSO	122	0.928	20.948	0.063	24.447
SLOPE	59	0.906	22.055	0.070	-
Dim.red					
LSI-LASSO	138	0.961	21.221	0.065	14.255
LDA-LASSO	84	0.936	23.002	0.076	14.946
PLS	8	0.902	15.653	0.035	19.127
Aggregated					
AP_{10}	10	0.919	25.919	0.097	16.700
AP_{20}	20	0.922	24.535	0.087	15.546

* refers to the number of selected variables in the model.

As expected, regularization with the Lasso reduces the out-of-sample prediction error. In the no-brand effect scenario, the CV estimate of Prediction Error drops from 70.056 to 35.428 (approximately 50% lower) with 82 non zero coefficients. With brand effects, the estimated RMSE drops from 60.205 to 20.948 (66% lower - with 122 non zero coefficients). The fit obtained using SLOPE provides very similar results compared to that of the Lasso in the no-brand scenario. The estimate of prediction error is slightly better and the two models select approximately the same number of variables. The situation is different when we consider also the brands. The Lasso retains a greater number of variables, on the contrary the selection made by SLOPE is more strict. The reason may rely on the fact that SLOPE has a higher control of false discoveries, specially if we add a strong signal like the brand, while the Lasso tends to select too many irrelevant variables, since the tuning parameter is selected to minimize prediction error. In fact, the estimated prediction error is a bit lower if compared to SLOPE.

The results obtained using dimension reduction show good predictive performances. In the case of LSI with brand effects, cross-validation suggests to select a model with 200 left singular vectors. Among these, the Lasso selects 138 left singular vectors and brand dummies yielding adj-$R^2 = 0.96$ and $RMSE_{cv} = 21.221$. This value is selected in correspondence to a low value of the tuning parameter λ equal to 0.019. Thus, prediction error should be close to what would be obtained without adding the penalization. The same tuning procedure omitting the brand effect yields adj-$R^2 = 0.884$ and $RMSE_{cv} = 32.167$ which improves the Lasso prediction by 10%. Regression on lower dimensional representation of documents provided by LDA and brand effects yields adj-$R^2 = 0.936$, $RMSE_{cv} = 23.002$ ($k = 80, \alpha = 0.009, \phi = 0.019$). The same procedure excluding brand effect yields adj-$R^2 = 0.855$, $RMSE_{cv} = 33.529$, ($k = 80, \alpha = 0.01, \phi = 0.02$) which is approximately 1% lower than what we obtained applying the Lasso on the OLS model plus brand. Comparing our model selection procedure with that proposed in [Taddy, 2013] we noticed better predictive performances. For example, Taddy [2013] sets $\beta = 1/K, \alpha = 1/V$ to run a regression using LDA. With this setting, the lowest estimate of out-of-sample prediction error that we achieve is 38.62 for the model with no brand effects and 24.06 with brand effects. Thus, tuning over β and α can effectively achieve better predictive results. (see Figure 4.6).



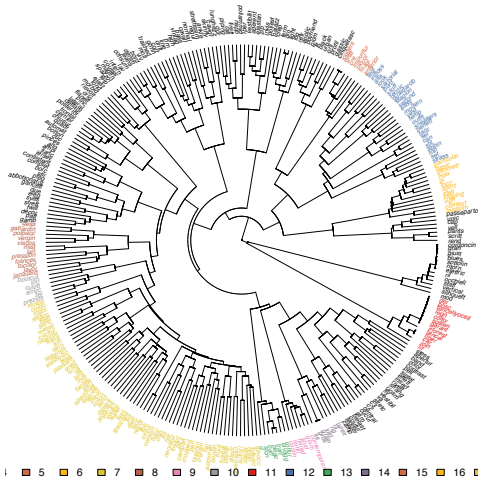
(i) *Brand* **(ii)** *No brand*

Figure 4.6: a) Cross-validated Root MSE using the grid search optimization. b) Cross-validated Root MSE using $\alpha = 1/V, \beta = 1/K$.

In terms of predictive accuracy, we obtained the best results using PLS. For a regression without brand dummies we get $\text{RMSE}_{cv} = 27,629$. This estimate is 56% lower than what we obtained with OLS and 21% lower than that obtained with the Lasso. Plugging these directions into an OLS model with brand dummies we get $\text{RMSE}_{cv} = 15,653$ which improves OLS and Lasso CV estimate of Root-MSE respectively by 72% and 37%.

The results of hedonic pricing using aggregated predictors obtained with Algorithm 8 show that prediction is worse if compared to other methods. Still, our results confirms empirically the results in Park et al. [2007] that aggregating predictors can provide some benefit in terms of predictive accuracy with respect to ordinary least squares. In particular, the prediction error is, on average, approximately 40% lower than OLS in the specification with no brand effects and almost 60% lower including brand effects. Figure 4.7 illustrates the hierarchical hidden structure of words. Words corresponding to the same group are given the same color. The values $g = 10, 20$ have been selected arbitrarily to avoid a clustering structure too dependent on brand effects that may have prevailed for values close the effective number of brands available.

(i) $g = 10$ *groups*



(ii) $g = 20$ *groups*

Figure 4.7: Hierarchical Clustering of word vectors for trousers DTM

Regarding the interpretability of our results, the variable selection procedures that we adopted have screened out many irrelevant terms. Plus, the interpretation is straightforward as each variable

corresponds to a single word. Unfortunately, the approach of dimension reduction led to a too large number of topics to be correctly interpreted, meaning that our pricing algorithm is nearly a black-box. We suspect this behaviour to be related to a low variance in the word frequencies. While one could expect that the reason is that description are short, we notice by looking at the Gibbs update in Algorithm 2 that the length of the description it is not a determinant of the topic assignments. Indeed, topic assignments depend on the number of times that a given words is assigned to a specific topic, thus, if descriptions were to spend words each on a given topic then a properly specified topic model should be able to find good topics. On the contrary, if words were evenly spread along documents then we would not be able to estimate a good topic model from data, because latent topics will not emerge from text. Eventually, we notice that some superwords have a nice practical interpretation. For example, in Figure 4.7i, Group 1 gathers the effect of tencel (*tencel, tencelyoce, fiber*), Group 2 gather words like combine (*abbin*), outfit, wardrobe (*guardarob*) and look which value of the item when paired with others. Group 3 contains features related to maintenance (*rottur, lavaggio, trattamento*). Group 8 collects words regarding trousers with skinny fit.

4.2.2 Dresses

descriptive statistics The disposable data consists of 814 observations and 309 textual features from descriptions (Table 4.1). The Document Term Matrix obtained by preprocessing the descriptions is displayed in Figure 4.8. The main difference that emerges from that observed for dresses (Figure 4.2) is that few words have larger frequencies in the collection.

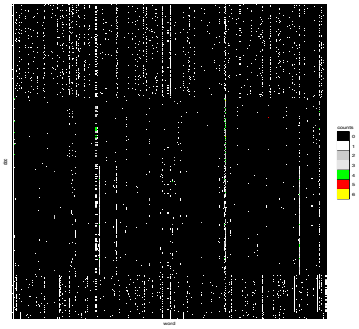


Figure 4.8: Document Term Matrix for descriptions of dresses.

The wordcloud in Figure 4.9 shows the most frequent words in the collection. Interestingly, in this category the most common words regard materials: poliester (*poliestere*), viscosa, elastan, cotton, (*cotone*), wool (*lana*).

Figure 4.9: Wordcloud of most common words in dresses descriptions

Figure 4.10 shows the distribution of the length of the descriptions for each brand. As observed for trousers, Z and HM show similar distributions as well as PP and EF. Retailer P is the one that spends more words in describing its products.

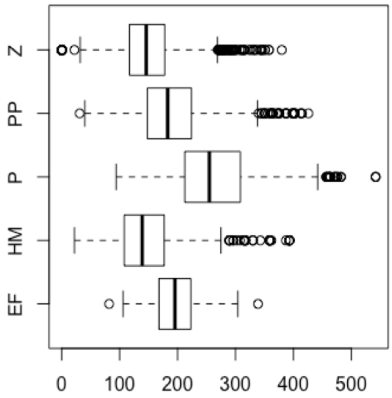


Figure 4.10: Conditional word count distribution.

Figure 4.11 suggests that few groups of words show high correlations albeit in general words are not highly correlated.

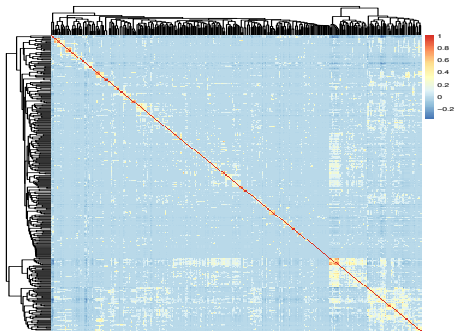


Figure 4.11: Correlation matrix of terms in dresses DTM.

Figure 4.12 shows the price distribution of each brand. It is clear that prices for fast fashion retailers like HM and Z have much lower average and median prices with respect to the other competitors.

Hedonic Text Regression Modeling Results in Table 4.4 suggest similar concerns as those noticed for trousers, the OLS models are probably overfitting the data. In fact, in front of high values of R^2, they provide very inaccurate predictions. For OLS plus brand effects, the estimate of prediction error is 199.851, which is higher compared to 150.270 of the OLS model with only brand dummies. Opposed to what observed for trousers, we notice that in this category the effect of the brands explains much less proportion of the total variability - even though it is still a high percentage.

The effect of regularization is to reduce dramatically the out of sample prediction error, regardless of the specification of brand ef-

Table 4.4: Results of hedonic text pricing models for dresses category

	No brand				
Fit	nvar*	adjR^2	RMSE$_{cv}$	RSE	MAE$_{cv}$
BASE	1	-	219.709	1	-
Text					
OLS	309	0.923	236.437	1.158	108.846
Sparse					
LASSO	99	0.9	113.256	0.266	63.109
SLOPE	87	0.897	108.935	0.246	-
Dim.red					
LSI-LASSO	154	0.863	119.996	0.293	72.367
LDA-LASSO	9	0.554	148.203	0.455	86.018
PLS	14	0.785	101.994	0.216	68.329
Aggregated					
AP$_{10}$	10	0.433	166.792	0.576	97.279
AP$_{20}$	20	0.548	156.697	0.509	95.807
	Brand				
Fit	nvar*	adjR^2	RMSE$_{cv}$	RSE	MAE$_{cv}$
BASE	4	0.499	150.27	0.468	78.398
Text					
OLS	313	0.928	199.851	0.827	113.591
Sparse					
LASSO	117	0.905	112.094	0.26	57.52
SLOPE	68	0.904	102.127	0.216	-
Dim.red					
LSI-LASSO	121	0.829	112.094	0.26	71.664
LDA-LASSO	32	0.652	132.069	0.361	71.283
PLS	18	0.933	53.367	0.059	36.125
Aggregated					
AP$_{10}$	10	0.521	154.009	0.491	78.932
AP$_{20}$	20	0.630	141.643	0.416	81.137

* refers to the number of variables in the model.

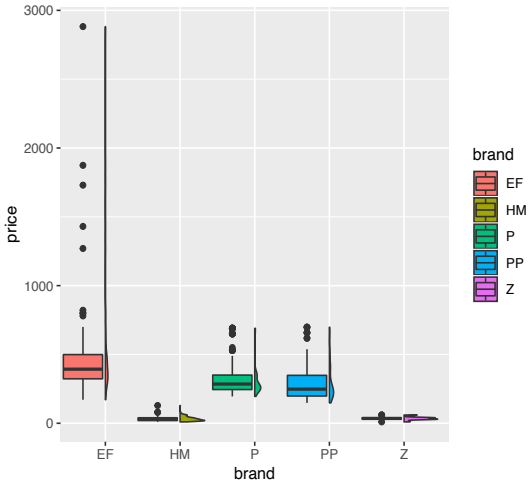


Figure 4.12: Price distributions for dresses

fects. Adding a Lasso regularization to the hedonic text regression model with no brand effects yields an estimate of RMSE approximately 52% lower. The same regularization applied to the model with brand effects yields a CV estimate of RMSE 44% lower. This lower reduction with respect to what we observed for trousers is likely due to the fact that within this category the brand has much lower explanatory power.

Opposed of what observed for trousers, dimensionality reduction via LSI or LDA led to higher prediction errors than sparse modeling. In particular, the prediction error that we attained with dimensionality reduction using LSI is approximately 5% higher than that obtained using the Lasso and 9% higher compared to SLOPE without allowing for brand effects. Allowing for brand effects improved the overall predictive accuracy, but did not change the relative order of the methodologies. The effect of additional regularization provided by the Lasso is more evident here than observed in trousers, since the values of λ are not close to zero. The prediction error that is attained using the lower dimensional representation provided by LDA is higher if compared to sparse modeling, LSI and PLS. In particular LDA yields the lowest value of adj-R^2. This is somehow surprising given the good performance that we observed for trousers. Figure 4.13 compares the optimal prediction error for LDA obtained via grid search with the setting $\alpha = 1/k, \phi = 1/V$. We notice that with this strategy we are able to explore a wider range of models in which to select the one with the best predictive accuracy. As observed for the category of trousers, we reach the

best results using PLS hedonic text-regression.

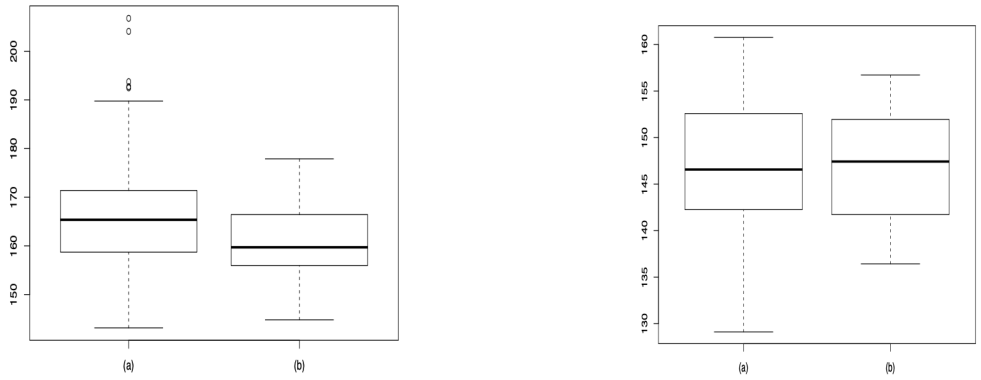
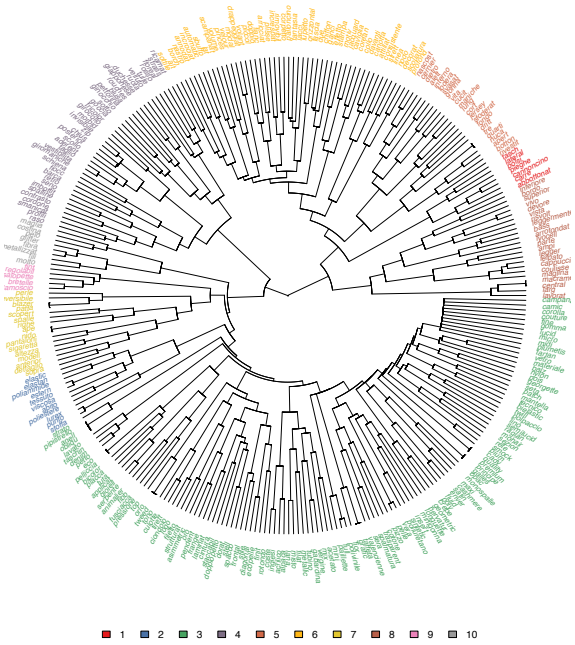


Figure 4.13: a) Cross-validated Root MSE using the grid search optimization. b) Cross-validated Root MSE using $\alpha = 1/V, \beta = 1/K$.

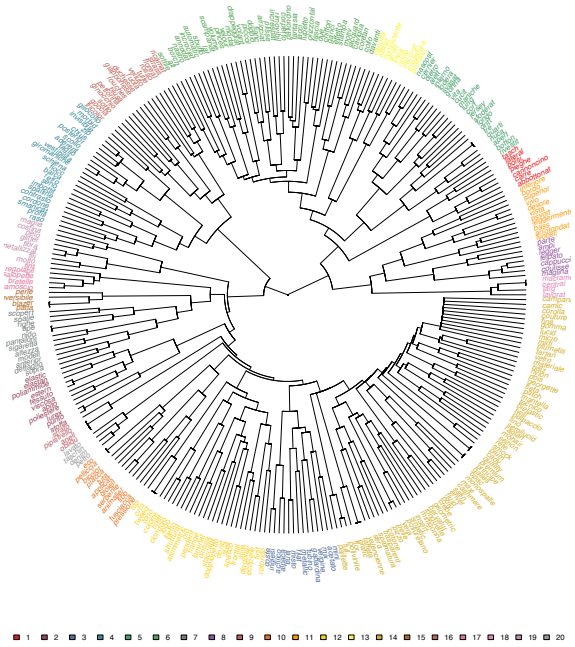
Again, hedonic regression based on aggregated predictors provides the worst fit. The fit with brand specification, attains adjR^2 = 0.521 and $RMSE_{cv}$ = 154.008 using 10 superwords, while adjR^2 = 0.630 and $RMSE_{cv}$ = 141.543 using 20 superwords. The fit with no brand specification instead attains adjR^2 = 0.433 and $RMSE_{cv}$ = 166.792 using 10 superwords, while adjR^2 = 0.548 and $RMSE_{cv}$ = 156.697 using 20 superwords. However, the predictive accuracy of this fit is better than that obtained with ordinary least squares even if the model explains less variance in data. This is again consistent with the theoretical results provided by Park et al. [2007].

Regarding the interpretability of the results, our findings are the same as those observed in the first category. The variable selection procedures are more interpretable, on the contrary dimension reduction does not give clear enough results to be interpreted. The interpretation of groups is made simpler with aggregating predictors. For example, in Figure 4.14i, we see that group 2 collects words that are associated with synthetic materials (*elastic, elastan, poliammide, viscosa*). This means that each word that belongs to this group will have the same marginal effect on price. Group 9 gathers words that remind to salopettes so that these attributes of this subcategory will contribute to prices in the same amount. In Figure 4.14ii, group 20 denotes a superword collecting attributes concerning fur coats (*pellicce*) and and leather finishes

(*leather, snake*). Group 14 contains clearly high quality attributes denoted by the words *mohair, premium, cashmere, galles*.



(i) $g = 10$ *groups*



(ii) $g = 20$ *groups*

Figure 4.14: Hierarchical Clustering of word vectors for dresses DTM

4.2.3 The role of the brand

In the models that we have considered so far, the choice of including brand effect seemed reasonable as clothes' prices are known to be heavily influenced by brands. However, given that we are modelling web data, we should also ask what would happen if items from a new brand became available and we wanted to estimate their prices based on their descriptions. A model that is not including brand effects should be more able to generalize to new data from a previously unseen brand. Suppose that new data from one *new* fashion brand become available and we wanted to model this data using a previously estimated hedonic-text regression model. If the model contained brand effects, then the price for these new observations would be predicted by a model fitted on a training set that does not know the marginal effect for this new brand. On the contrary, if the model is specified without brand effects it will use only the marginal prices of the features from text. However, if brand effects were important determinant of prices then these coefficients will be biased due to omitted variables. The extent such that it affects prediction accuracy depends on bias-variance decomposition of prediction error. We conducted a test using the available data. We estimated a series of OLS text-regression models excluding each time a given brand - the test brand - and used the estimates obtained to predict prices of the test brand. Estimates prediction errors are reported in Table 4.5.

Table 4.5: Predicting a new brand.

Test Brand	Brand		No brand	
	Trousers	Dresses	Trousers	Dresses
ef	292	1760	360	1496
hm	267	31665	241	38178
pinko	129	3081	136	2913
pp	341	371	550	572
zara	8154	314	5025	262

These results suggest that it is not clear which strategy is the best. In the category of trousers, allowing for brand fixed effects turns out to perform better for predicting three out of five brands (EF, P and PP). On average though, no brand pricing is significantly better, specially when Z is used as test retailer. This could

be due to the fact that Z's items constitute a wide portion of data available. For dresses, the opposite holds. Predicting using brand effect yields better results only in two out of five scenarios. On the contrary the mean prediction error is 7438 when using brand and 8684 when not used.

Therefore, one possible strategy would be to use brand fixed effects, which turned out to have better prediction properties in terms of CV estimates of prediction error (see Tables 4.4-4.3), and re-estimate the model in case new massive data from a previously unobserved brand become available. However, this approach may be not particularly useful if data from multiple brands would become available at different time stamps, as we should re-estimate the model each time. For this reason, we believe that hedonic model that do not incorporate fixed effect may be easier to scale to new products.

CHAPTER

5

DISCUSSION

In this study we built hedonic pricing models of italian fashion products using the internet as a source of data. To this purpose, we web-scraped publicly available data from the websites of five famous fashion brands. In order to obtain a set of attributes, we text mined the descriptions of the items.

In fact, descriptions contain information on attributes, such as the composition of materials, that can be turned into a set of features via text-mining procedures. A study by Archak et al. [2011] claims that working with descriptions shall not be recommended since the text that is contained is too static and gives little emphasis on characteristics of the goods. On the contrary, our findings provided empirical evidence that this is not necessarily true. In fact, compared to the baseline models, the set of attributes that we obtained from the descriptions improved dramatically the predictive performance of the model. This is an outstanding result since clothes are well known to be heavily influenced by brand names - which was also confirmed empirically by our analysis.

A potential drawback of working with text data is that descriptions can contain lots of noisy variables but, interestingly, the variable selection procedures that we proposed in our analysis gave coefficients with expected sign. As example, in the category of trousers, the presence of synthetic materials (*sintet*, *poliest*) leads

on average to lower prices while high-quality materials like virgin wool (*vergin*) leads to higher. Also, words relating to details and to particular trousers' shapes lead to higher prices, as is the case of *palazzo* style trousers and attributes denoting *minimal* design. Among dresses, the red color, furs, leather, georgette, lightweight and embroideries are all attributes that reflect into higher prices, as well as evening dresses, while low quality materials like elastan reflect to lower. Remarkably, some of these effects would not have been considered if we did not text-mined the descriptions.

The attempt to use text mining techniques for dimensionality reduction and topic modelling provided good predictive performances (especially the usage of PLS) but unfortunately did not provide interpretable topics. The advantage of aggregating predictors over dimensionality reduction is that one can easily understand the marginal price of each word in a given group. This is because each word is assigned just to one group while in topic modelling each word can contribute to different topics with different weights. Indeed, the majority of the groups of words that we identified aggregating predictors have a meaningful interpretation as well as expected sign.

Throughout this study, while setting up different pricing models, we also explored a variety of solutions to estimate high-dimensional linear models applied to text data. This analysis may be of practical use in many real world problems even beyond the fashion industry. As example, the ultimate interest for a selling company like Mercari is to have a model that suggests the right price regardless of interpretability and consistency of results. For this reason, among the proposed models, Mercari would probably opt for a topic model based on PLS. However, this framework may also be of much practical use for brands themselves. Consider for example the launch of a new product. The brand management could just type in a description of what he wants to produce and the model would predict the price, or, he could predict the price for his competitors. This is certainly a great advantage for pricing strategies.

Eventually, a model that hinges in favour of interpretability can be of much interest from a consumers' perspective. In fact, while providing a fair price to pay, it would give also the fair contribution of each attribute. For example, if used in online shops, consumers can be made aware of the marginal markups for a given brand, colour, or design, all else equal. In this use case, probably a sparse hedonic model estimated with SLOPE may be preferable since the selection of variable is made as much as possible to control false

discoveries. From an economic point of view, this would reduce the asymmetry in information available to sellers and consumers and balance the amount of total surplus by reducing the seller's in favour of the consumer's.

As outline of future analysis we plan to experiment with non-parametric techniques like random forest or ensemble learning. Also, an interesting approach would be to create word features using word-embeddings like word2vec [Mikolov et al., 2013b]. These methods have become very popular in text mining over the last years. In addition, to further check the consistency of our results, we plan to apply the proposed models to newer datasets in the fashion industry and, eventually, we plan to experiment with other industries.

BIBLIOGRAPHY

C. C. Aggarwal. *Machine learning for text*. Springer, 2018.

C. C. Aggarwal and C. Zhai. *Mining text data*. Springer Science & Business Media, 2012.

W. Antweiler and M. Z. Frank. Is all that talk just noise? the information content of internet stock message boards. *The Journal of finance*, 59(3):1259–1294, 2004.

N. Archak, A. Ghose, and P. G. Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 56–65. ACM, 2007.

N. Archak, A. Ghose, and P. G. Ipeirotis. Deriving the pricing power of product features by mining consumer reviews. *Management science*, 57(8):1485–1509, 2011.

S. Arlot, A. Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

S. R. Baker, N. Bloom, and S. J. Davis. Measuring economic policy uncertainty. *The quarterly journal of economics*, 131(4):1593–1636, 2016.

G. Baltas and C. Saridakis. Measuring brand equity in the car market: a hedonic price analysis. *Journal of the Operational Research Society*, 61(2):284–293, 2010.

O. Bandiera, A. Prat, S. Hansen, and R. Sadun. Ceo behavior and firm performance. *Journal of Political Economy*, 128(4):1325–1369, 2020.

A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1): 289–300, 1995.

M. Berry and J. Kogan. Text mining: Applications and theory. john wiley & sons. 2010.

M. W. Berry and M. Castellanos. Survey of text mining. *Computing Reviews*, 45(9):548, 2004.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

M. Bogdan, E. Van Den Berg, C. Sabatti, W. Su, and E. J. Candes. Slope—adaptive variable selection via convex optimization. *The annals of applied statistics*, 9(3):1103, 2015.

J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008.

B. Born, E. Michael, and F. Marcel. Central bank communication on financial stability. *Economic Journal*, 124(577):701–34, 2014.

S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

J. Büschken and G. M. Allenby. Sentence-based text analysis for customer reviews. *Marketing Science*, 35(6):953–975, 2016.

G. P. Cachon and R. Swinney. The value of fast fashion: Quick response, enhanced design, and strategic consumer behavior. *Management science*, 57(4):778–795, 2011.

E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.

E. Cassel and R. Mendelsohn. The choice of functional forms for hedonic price equations: comment. *Journal of Urban Economics*, 18(2):135–142, 1985.

A. Cavallo. Are online and offline prices similar? evidence from large multi-channel retailers. *American Economic Review*, 107 (1):283–303, 2017.

A. Cavallo. Scraped data and sticky prices. *Review of Economics and Statistics*, 100(1):105–119, 2018.

A. Celisse et al. Optimal cross-validation in density estimation with the l2-loss. *The Annals of Statistics*, 42(5):1879–1910, 2014.

J. A. Chevalier and D. Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of marketing research*, 43(3): 345–354, 2006.

A. Court. *Hedonic price indexes with automotive examples*. 1939.

A. Cowles 3rd. Can stock market forecasters forecast? *Econometrica: Journal of the Econometric Society*, pages 309–324, 1933.

S. P. Crain, K. Zhou, S.-H. Yang, and H. Zha. Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond. In *Mining text data*, pages 129–161. Springer, 2012.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

I. Dhillon, J. Kogan, and C. Nicholas. Feature selection and document clustering. In *Survey of Text Mining*, pages 73–100. Springer, 2004.

W. Dubitzky, M. Granzow, and D. P. Berrar. *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media, 2007.

L. Einav and J. Levin. Economics in the age of big data. *Science*, 346(6210):1243089, 2014.

L. Einav, T. Kuchler, J. D. Levin, and N. Sundaresan. Learning from seller experiments in online markets. Technical report, National Bureau of Economic Research, 2011.

L. Einav, C. Farronato, J. D. Levin, and N. Sundaresan. Sales mechanisms in online markets: What happened to internet auctions? Technical report, National Bureau of Economic Research, 2013.

J. E. Engelberg and C. A. Parsons. The causal impact of media in financial markets. *The Journal of Finance*, 66(1):67–97, 2011.

N. Evangelopoulos, X. Zhang, and V. R. Prybutok. Latent semantic analysis: five methodological recommendations. *European Journal of Information Systems*, 21(1):70–86, 2012.

R. C. Feenstra and M. D. Shapiro. *Scanner data and price indexes*, volume 64. University of Chicago Press, 2007.

L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531. IEEE, 2005.

D. P. Foster, M. Liberman, and R. A. Stine. Featurizing text: Converting text into predictors for regression analysis. *The Wharton School of the University of Pennsylvania, Philadelphia, PA*, 2013.

A. M. Freeman III. Benefits of environmental improvement: theory and practice. 1979.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

M. Gentzkow and J. M. Shapiro. What drives media slant? evidence from us daily newspapers. *Econometrica*, 78(1):35–71, 2010.

M. Gentzkow, B. Kelly, and M. Taddy. Text as data. *Journal of Economic Literature*, 57(3):535–74, 2019.

G. George, E. C. Osinga, D. Lavie, and B. A. Scott. Big data and data science methods for management research, 2016.

A. Ghose, P. Ipeirotis, and A. Sundararajan. Opinion mining using econometrics: A case study on reputation systems. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 416–423, 2007.

J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2009.

T. Groseclose and J. Milyo. A measure of media bias. *The Quarterly Journal of Economics*, 120(4):1191–1237, 2005.

M. Gupta and J. F. George. Toward the development of a big data analytics capability. *Information & Management*, 53(8):1049–1064, 2016.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

R. Halvorsen and H. O. Pollakowski. Choice of functional form for hedonic price equations. *Journal of urban economics*, 10(1): 37–49, 1981.

S. Hansen, M. McMahon, and A. Prat. Transparency and deliberation within the fomc: a computational linguistics approach. *The Quarterly Journal of Economics*, 133(2):801–870, 2018.

T. Hastie and J. Qian. Glmnet vignette. *Retrieve from http://www. web. stanford. edu/hastie/Papers/Glmnet/ignette. pdf. Accessed September*, 20:2016, 2014.

G. Hoberg and G. Phillips. Text-based network industries and endogenous product differentiation. *Journal of Political Economy*, 124(5):1423–1465, 2016.

N. Jegadeesh and D. Wu. Word power: A new approach for content analysis. *Journal of Financial Economics*, 110(3):712–729, 2013.

H. A. Johnson, M. M. Wagner, W. R. Hogan, W. W. Chapman, R. T. Olszewski, J. N. Dowling, G. Barnas, et al. Analysis of web access logs for surveillance of influenza. In *Medinfo*, pages 1202–1206, 2004.

M. Joshi, D. Das, K. Gimpel, and N. A. Smith. Movie reviews and revenues: An experiment in text regression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 293–296. Association for Computational Linguistics, 2010.

D. Jurafsky and J. H. Martin. *Speech and language processing*, volume 3. Pearson London, 2014.

B. Kelly, D. Papanikolaou, A. Seru, and M. Taddy. Measuring technological innovation over the long run. Technical report, National Bureau of Economic Research, 2018.

S. Kogan, D. Levin, B. R. Routledge, J. S. Sagi, and N. A. Smith. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280. Association for Computational Linguistics, 2009.

T. Kwartler. *Text mining in practice with R.* John Wiley & Sons, 2017.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

D. Lenz, P. Winker, et al. Measuring the diffusion of innovations with paragraph vector topic models. Technical report, Philipps-Universitat Marburg, Faculty of Business Administration, 2018.

B. Liu. *Sentiment analysis: Mining opinions, sentiments, and emotions.* Cambridge University Press, 2015.

L. Liu, J. Kang, J. Yu, and Z. Wang. A comparative study on unsupervised feature selection methods for text clustering. In *2005 International Conference on Natural Language Processing and Knowledge Engineering*, pages 597–601. IEEE, 2005.

T. Loughran and B. McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

D. O. Lucca and F. Trebbi. Measuring central bank communication: an automated approach with application to fomc statements. Technical report, National Bureau of Economic Research, 2009.

A. Manela and A. Moreira. News implied volatility and disaster concerns. *Journal of Financial Economics*, 123(1):137–162, 2017.

C. Manning, P. Raghavan, and H. Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

J. D. Mcauliffe and D. M. Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

D. Meyer, K. Hornik, and I. Feinerer. Text mining infrastructure in r. *Journal of statistical software*, 25(5):1–54, 2008.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

M. Monson. Valuation using hedonic pricing models. *Cornell Real Estate Review*, 7(1):10, 2009.

A. Nowak and P. Smith. Textual analysis in real estate. *Journal of Applied Econometrics*, 32(4):896–918, 2017.

M. Y. Park, T. Hastie, and R. Tibshirani. Averaged gene expressions for regression. *Biostatistics*, 8(2):212–227, 2007.

J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

M. F. Porter. Snowball: A language for stemming algorithms, 2001.

M. Rabinovich and D. Blei. The inverse regression topic model. In *International Conference on Machine Learning*, pages 199–207, 2014.

R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, 2014.

S. Rosen. Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of political economy*, 82(1): 34–55, 1974.

A. Saiz and U. Simonsohn. Proxying for unobservable variables with internet document-frequency. *Journal of the European Economic Association*, 11(1):137–165, 2013.

G. Salton and M. J. McGill. Introduction to modern information retrieval. 1986.

S. L. Scott and H. R. Varian. Bayesian variable selection for nowcasting economic time series. Technical report, National Bureau of Economic Research, 2013.

S. L. Scott and H. R. Varian. Predicting the present with bayesian structural time series. *International journal of Mathematical Modeling and Numerical Optimization*, 5((1-2)):4–23, 2014.

S. Stephens-Davidowitz. The cost of racial animus on a black candidate: Evidence using google search data. *Journal of Public Economics*, 118:26–40, 2014.

M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

W. Su, M. Bogdan, E. Candes, et al. False discoveries occur early on the lasso path. *The Annals of statistics*, 45(5):2133–2150, 2017.

M. Taddy. Multinomial inverse regression for text analysis. *Journal of the American Statistical Association*, 108(503):755–770, 2013.

P. C. Tetlock, M. Saar-Tsechansky, and S. Macskassy. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467, 2008.

L. A. Thorsrud. Words are the new numbers: A newsy coincident index of the business cycle. *Journal of Business & Economic Statistics*, pages 1–17, 2018.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

R. Tibshirani, I. Johnstone, T. Hastie, and B. Efron. Least angle regression. *The Annals of Statistics*, 32(2):407–499, Apr 2004. ISSN 0090-5364. doi: 10.1214/009053604000000067. URL http://dx.doi.org/10.1214/009053604000000067.

R. Tibshirani, M. Wainwright, and T. Hastie. *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015.

C. J. Van Rijsbergen, S. E. Robertson, and M. F. Porter. *New models in probabilistic information retrieval*. British Library Research and Development Department London, 1980.

M. J. Wainwright, M. I. Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

S. M. Weiss, N. Indurkhya, and T. Zhang. *Fundamentals of predictive text mining*. Springer, 2015.

T. P. Wisniewski and B. Lambe. The role of media in the credit crunch: The case of the banking sector. *Journal of Economic Behavior & Organization*, 85:163–175, 2013.

Y. Yao, A. Vehtari, D. Simpson, and A. Gelman. Yes, but did it work?: Evaluating variational inference. *arXiv preprint arXiv:1802.02538*, 2018.

X. Zeng and M. Wagner. Modeling the effects of epidemics on routinely collected data. *Journal of the American Medical Informatics Association*, 9(Supplement_6):S17–S22, 2002.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.