

Topics in Computational Advertising

by

Timothy C. Au

Department of Statistical Science
Duke University

Date: _____

Approved:

David L. Banks, Supervisor

James O. Berger

Li Ma

James W. Roberts

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2014

ABSTRACT

Topics in Computational Advertising

by

Timothy C. Au

Department of Statistical Science
Duke University

Date: _____

Approved:

David L. Banks, Supervisor

James O. Berger

Li Ma

James W. Roberts

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2014

Copyright © 2014 by Timothy C. Au
All rights reserved except the rights granted by the
Creative Commons Attribution-Noncommercial Licence

Abstract

Computational advertising is an emerging scientific discipline that incorporates tools and ideas from fields such as statistics, computer science, and economics. Although a consequence of the rapid growth of the Internet, computational advertising has since helped transform the online advertising business into a multi-billion dollar industry.

The fundamental goal of computational advertising is to determine the “best” online ad to display to any given user. This “best” ad, however, changes depending upon the specific context that is under consideration. This leads to a variety of different problems, three of which are discussed in this thesis.

Chapter 1 briefly introduces the topics of online advertising and computational advertising. Chapter 2 proposes a numerical method to approximate the pure strategy Nash equilibrium bidding functions in an independent private value first-price sealed-bid auction where bidders draw their types from continuous and atomless distributions—a setting in which solutions cannot generally be analytically derived, despite the fact that they are known to exist and to be unique. Chapter 3 proposes a cross-domain recommender system that is a multiple-domain extension of the Bayesian Probabilistic Matrix Factorization model. Chapter 4 discusses some of the tools and challenges of text mining by using the Trayvon Martin shooting incident as a case study in analyzing the lexical content and network connectivity structure of the political blogosphere. Finally, Chapter 5 presents some concluding remarks and briefly discusses other problems in computational advertising.

To my family

Contents

Abstract	iv
List of Tables	ix
List of Figures	x
List of Abbreviations and Symbols	xiv
Acknowledgements	xv
1 Introduction	1
2 Auctions: The Backwards Indifference Derivation (BID) Algorithm	5
2.1 Introduction	5
2.2 Standard Model: Theory and Notation	6
2.2.1 Symmetric Auctions	10
2.2.2 Asymmetric Auctions	11
2.3 Numerical Methods	13
2.4 Equilibrium Solution for Auctions with Finite Sets	18
2.4.1 Athey (2001)	19
2.5 Points of Indifference	23
2.6 Proposed Algorithm: The Backwards Indifference Derivation (BID) Algorithm	25
2.6.1 Core Algorithm: Constructing Finite-Action PSNE	28
2.6.2 Outer Algorithm: Estimating \bar{b}	34

2.6.3	Proof that the BID Algorithm Constructs a Finite-Action PSNE	38
2.7	Using Economic Theory to Evaluate Numerical Solutions	43
2.8	Extensions of the Standard Model	52
2.8.1	Different Supports for the Type Distributions	53
2.8.2	Different Utility Functions	57
2.9	Applications	62
2.10	Concluding Remarks	66
3	Cross-Domain Recommender Systems	68
3.1	Introduction	68
3.2	Related Work	69
3.2.1	Probabilistic Matrix Factorization (PMF)	71
3.2.2	Bayesian Probabilistic Matrix Factorization (BPMF)	75
3.2.3	Other Single-Domain PMF and BPMF Extensions	80
3.2.4	Multi-Domain Collaborative Filtering	81
3.3	Proposed Model	85
3.3.1	Posterior Inference	87
3.4	Experiments	91
3.4.1	Cross-Domain Recommendations	93
3.4.2	Semi-cold-start Recommendations	95
3.5	Conclusions	97
4	Text Mining: A Trayvon Martin and Political Blogs Case Study	98
4.1	Introduction	98
4.2	Trayvon Martin Shooting Incident	99
4.2.1	The Shooting	99
4.2.2	Media Coverage and Public Response	100

4.2.3	Court Case and Statutory Self-Defense Laws	102
4.3	Data	103
4.3.1	Data Collection	103
4.3.2	Data Fields	105
4.3.3	Data Quality and Data Cleaning	106
4.4	Data Analysis	108
4.4.1	Multi-word Expressions: <i>n</i> -grams	109
4.4.2	Sentiment Analysis and Word Usage	112
4.4.3	Topic Modeling and the Blogosphere	115
4.5	Conclusions	122
5	Concluding Remarks	124
	Bibliography	127
	Biography	132

List of Tables

3.1	Summary of the multiple-domain MovieLens data that we have constructed.	94
3.2	Breakdown of each method’s best RMSE scores and the feature dimensionality under which it occurred in the cross-domain recommendation situation.	95
3.3	Breakdown of each method’s best RMSE scores and the feature dimensionality that it occurred under in the semi-cold-start recommendation situation.	97
4.1	The most frequently used n -grams in the Trayvon Martin corpus that were identified by the turbo topics model (table reproduced from Soriano et al. (2013)).	112
4.2	The top 15 words in each topic found by LDA for our Trayvon Martin dataset (Table reproduced from Soriano et al. (2013)).	118

List of Figures

2.1	A stylistic example of a nondecreasing finite-action step function strategy α_n^M that specifies when bidder n “jumps” to a higher action. Notice that Definition 2 allows for some jump points to be equal. So, for this example $v_n^3 = v_n^4$ implies that the probability of action b^3 being used is 0. When we propose our own algorithm in Section 2.6, however, we impose an additional constraint that prevents this from happening.	21
2.2	The behavior of the core part of the BID algorithm when it was initialized at different guesses for the maximal bid. Earlier we showed that PSNE solution is given by $\beta(v) = \frac{v}{2}$ and that the true maximal bid is known to be $\bar{b} = \frac{1}{2}$. Initializing the algorithm at this true value produced results that agree with the known PSNE solution. When initialized too high at \bar{b}_H , however, the results approached the 45° line. When initialized too low at \bar{b}_L , results diverged to $-\infty$	36
2.3	Type distributions for Example 2. These distributions do not cross in the interior as F_2 stochastically dominates F_1	42
2.4	Finite-action PSNE constructed by the BID algorithm for Example 2 using various step sizes h to investigate its convergence properties. Notice that, despite the fact that the type distributions did not cross, the PSNE bidding functions appear to converge to continuous functions that cross once in the interior—a result which is consistent with Kirkegaard (2009). An accuracy of $\gamma = 10^{-8}$ was used in all three cases. Left: $h = .1$. Center: $h = .01$. Right: $h = .001$	43
2.5	Stylistic depiction of two “possible” paths that $R_{i,j}$ could take that would still be consistent with economic theory (figure reproduced from Kirkegaard (2009)). Although qualitatively there are many such possible paths, quantitatively only one will satisfy the conditions of the PSNE solution since it is unique.	45

2.6	Three examples of $F_{i,j}$ which exhibit the diminishing wave property (figure reproduced from Kirkegaard (2009)). Left: A stylistic example. Center: $F_i(v_i) = (\frac{v_i}{5})^2, v_i \in [0, 5]$ and $F_j(v_j)$ a normal distribution truncated on $[0, 5]$ with mean $\mu = 3$ and standard deviation $\sigma = 1$. Right: $F_i(v_i) = (v_i/10)^2, v_i \in [0, 10]$ while $F_j(v_j) = \frac{1}{3}(G_1(v_j)+G_2(v_j)+G_3(v_j))$ where G_i is a normal distribution truncated on $[0, 10]$ with mean $\mu_k = 3k$ and standard deviation $\sigma_1 = \sigma_2 = 1$ and $\sigma_3 = 0.25$	46
2.7	The BID algorithm applied to Example 2, where the type distributions exhibited first order stochastic dominance. Kirkegaard (2009) showed that this is a necessary, but not sufficient condition for the PSNE solutions not to cross. Left: The type distributions that do not cross. Center: The finite-action PSNE solution constructed by the BID algorithm that cross once (crossing indicated by the vertical dotted line). Right: The BID algorithm passed the necessary visual test proposed by Hubbard et al. (2013) (the vertical dotted line that appears here is the same as the one in the center panel).	48
2.8	The BID algorithm applied to Example 3, where $F_{1,2}$ exhibits the diminishing wave property. Left: The type distributions. Center: The finite-action PSNE solution constructed by the BID algorithm (crossings indicated by the vertical dotted line). Right: The BID algorithm passed the necessary visual test proposed by Hubbard et al. (2013) (the vertical dotted lines that appears here are the same as the ones in the center panel).	49
2.9	The BID algorithm applied to a three bidder situation. Left: The type distributions which cross twice in the interior (at $v_n = \sqrt{0.25}$ and $v_n = \sqrt{0.75}$). Right: The finite-action PSNE solution constructed by the BID algorithm.	50
2.10	The BID algorithm applied to a three bidder situation. Top Row: The finite-action PSNE constructed by the algorithm for each pair of bidders (crossings indicated by the vertical dotted lines). Bottom Row: The BID algorithm passed the necessary test proposed by Hubbard et al. (2013) for each pair of bidders (the vertical dotted lines that appears in each panel are the same as the ones in the panel directly above it).	52
2.11	The BID algorithm applied to Example 5, where the type distributions have different supports. These results agree with the known closed-form analytic solution derived in Kaplan and Zamir (2012).	57

2.12	The BID algorithm applied to a symmetric risk-averse bidders situation. Results agreed with the known closed-form analytic solution derived in Krishna (2002).	60
2.13	The BID algorithm applied to a situation where bidders are symmetric in their type distributions, but asymmetric in their degrees of risk aversion. Results, which show the more risk averse bidder (bidder 2) bidding more aggressively, are consistent with Maskin and Riley (1984).	61
2.14	Monte Carlo simulations for Example 8. Left: When there is no reserve price, the first-price auction generates more expected revenue than the second price auction; under the optimal reserve price of $r = .66$, however, the second-price auction actually generated more revenue. Right: The variances of the revenues. The second-price auction always had a higher variance.	65
3.1	Comparison of each model's overall test RMSE in the cross-domain recommendation situation for different levels of the latent feature dimensionality D . Notice that the performance of the Bayesian models continues to improve as D increases, but the MAP-trained MCF model begins to overfit. Finally, observe that the proposed BPFM-MD model outperforms every other model for all values of D	94
3.2	Comparison of each model's overall test RMSE in the semi-cold-start recommendation situation for different levels of the latent feature dimensionality D . Once again, the performance of the Bayesian models continues to improve as D increases, while the MAP-trained MCF model begins to overfit. And like the previous example, the proposed BPFM-MD model outperforms every other model for all values of D	96
4.1	Histogram of the estimated timestamps for the documents in the Trayvon Martin corpus, where the vertical dotted line corresponds to February 26, 2012 (the date the shooting actually occurred). Notice that two of the documents have been dated before the actual incident occurred, which is clearly an error committed by the sequential search method.	108
4.2	Histogram of the polarity scores of the documents in the Trayvon Martin corpus. Vertical dotted lines appear at $\pm .15$, and indicate the cutoff points used for classifying the documents as either positive, negative, or neutral.	114

4.3	Wordclouds contrasting the usage of tokens across documents in the Trayvon Martin corpus. Left Panel: Tokens appearing significantly more often in negative documents. Right Panel: Tokens appearing significantly more often in positive documents.	115
4.4	The plate diagram for the basic LDA generative process (figure reproduced from Soriano et al. (2013)).	117
4.5	Trayvon Martin Blogosphere on March 21, 2012 just as the incident started to become national news (figure reproduced from Soriano et al. (2013)). Early coverage of the story seems to be mostly focused on the actual details of the event, and the more authoritative blogs appear to be driving the discussion.	120
4.6	Trayvon Martin Blogosphere on April 11, 2012—the day that Zimmerman was finally charged and arrested (figure reproduced from Soriano et al. (2013)). The discussion seems to have shifted towards the racial and political topics.	121
4.7	Trayvon Martin Blogosphere on May 17, 2012 when prosecutors first publicly released evidence from the case (figure reproduced from Soriano et al. (2013)). Despite this fact, discussion still seems to be focused on the racial and political aspects of the incident rather than the legal one.	122

List of Abbreviations and Symbols

Abbreviations

BID	Backwards Indifference Derivation
BPMF	Bayesian Probabilistic Matrix Factorization
LDA	Latent Dirichlet allocation
MAP	Maximum a posteriori
MCF	Multi-Domain Collaborative Filtering
MCMC	Markov chain Monte Carlo
PMF	Probabilistic Matrix Factorization
PSNE	Pure strategy Nash equilibrium
RMSE	Root mean square error
RTB	Real-time bidding
SCC	Single crossing condition

Acknowledgements

First, I would like to thank my advisor, David Banks, for his constant support and encouragement. This thesis would not have been possible without his guidance, and I have learned a tremendous amount from him during our time together.

I would also like to thank Jim Berger, Li Ma, Jimmy Roberts, and Susie Bayarri for taking the time to serve on my committee and for providing valuable comments.

I am also extremely grateful to MaxPoint Interactive for their financial support of part of my graduate studies, and for providing me with my first experience in the Internet advertising industry. In particular, I would like to thank Michael Els, Mark Lowe, and Sean Murphy for their insights and encouragements.

I owe a huge thanks to Tsuyoshi Kuniyama and Daniel Heard for helping me to make it through these past four years. Thanks also to Fernando Bonassi, Mary Beth Broadbent, Andrew Cron, Monika Hu, Tommy Leininger, Thais Paiva, Shaan Qamar, Jacopo Soriano, Maria Terres, Doug VanDerwerken, Zoey Zhao, and many others for enriching my time in the department—whenever I was actually there, of course...

I would like to thank my family for their never-ending love, support, and encouragement throughout my entire life.

Finally, I would like to thank God for carrying me through these past four years. It has been quite the journey.

1

Introduction

In online advertising, advertisers use the Internet to distribute promotional marketing messages to consumers. Already a multi-billion dollar business, this industry continues to rapidly grow and evolve as new technologies emerge and develop.

Of course, this growth has only been possible because advertisers have found the Internet to be an extremely effective medium for delivering their ads. In particular, online ads have proven to be considerably cheaper and more targetable than their offline counterparts (e.g., print and television ads). Coupled with the global market coverage and speed at which online ads can be deployed, advertisers have been quick to adopt the Internet as a premiere marketing platform. Furthermore, due to the immediate feedback mechanisms that exist (e.g., clicks and conversions), online ads have also made it easier for advertisers to directly evaluate and improve their advertising efforts.

Although they lie at the other end of the industry, web publishers have also played a key role in the rise of online advertising. Most notably, publishers have become the content producers of the Internet, and they provide the web properties that allow advertisers to reach consumers. In return, publishers have been able to secure a

revenue stream for themselves that was unimaginable just over a decade ago.

Web users have also benefited enormously from the success of online advertising. Exposure to relevant Internet ads has made users more intelligent consumers who are more aware of their purchasing options. Furthermore, much of the web content that users enjoy is only available because potential ad revenue incentivizes publishers to produce it. Similarly, many of the free services that users regularly use (e.g., search engines, email, and social media platforms) are only possible because they are supported by advertising revenue.

Perhaps the largest stakeholders in this industry, however, are the companies such as Google, Yahoo!, and Facebook that are actively involved in all facets of the online advertising process. In addition to operating the major ad exchanges that facilitate the buying and selling of online ads, these companies also function as publishers who provide advertisers with highly unique advertising opportunities such as sponsored search ads and social media marketing. Furthermore, these companies often offer tools that either help advertisers increase the effectiveness of their advertising campaigns, or help publishers improve the performance of ads on their websites.

The development of these tools has helped lead to the emergence of computational advertising—a new scientific discipline that incorporates techniques and ideas from fields such as statistics, computer science, and economics. Consequently, the fundamental goal of computational advertising is to determine the “best” online ad to display to any given user. This “best” ad, however, changes depending upon the specific context that is under consideration. This leads to a variety of different problems, three of which are discussed in this thesis.

Chapter 2 covers the topic of auctions, which is the mechanism that ad exchanges use to buy and sell online advertisements. In this chapter, we propose the Backwards Indifference Derivation (BID) algorithm to numerically approximate the pure strategy Nash equilibrium bidding functions in an independent private value first-price

sealed-bid auction where bidders draw their types from continuous and atomless distributions—a setting in which solutions cannot generally be analytically derived, despite the fact that they are known to exist and to be unique. The BID algorithm is motivated in part by the results derived in Athey (2001), and attempts to construct a sequence of finite-action equilibria that converges to the continuum-action solution. Consequently, our approach differs from other numerical methods that directly consider a system of poorly behaved differential equations. We then evaluate the performance of the BID algorithm using numerical examples—including situations that other numerical algorithms have not yet considered or are unable to handle—and our results show that it produces solutions that are consistent with economic theory. Finally, we use the BID algorithm to investigate the area of auction design, which can provide insights into how ad auctions can be modified in order to generate more revenue.

Chapter 3 focuses on the area of recommender systems, which are a class of models that attempt to predict how a user will respond to an item. Consequently, recommender systems are particularly important in the field of computational advertising because they can help to identify the most cost-effective ad to display to the user. Traditionally, research in this space has focused on models that make recommendations within a single domain (e.g., movies or music). Oftentimes, however, advertisers would like to make recommendations that span multiple-domains. In these situations, recommendations in one domain can potentially be improved by leveraging knowledge from other domains—a concept that has motivated the recent development of cross-domain recommender systems. In this chapter we propose a cross-domain recommender system that is a multiple-domain extension of the Bayesian Probabilistic Matrix Factorization model proposed by Salakhutdinov and Mnih (2008b). We then provide experimental results showing that our proposed model outperforms other similar models in two different cross-domain recommendation situations.

In Chapter 4, we discuss some of the tools and challenges of text mining by using the Trayvon Martin shooting incident as a case study in analyzing the lexical content and network connectivity structure of the political blogosphere. Although the event and the set of websites that we investigate are not necessarily aligned with computational advertising, the tools that we use can be applied more broadly in order to better understand consumer preferences and track how online content evolves over time. Indeed, a growing front in computational advertising seeks to use text mining and network analysis to help select relevant online ads.

Finally, Chapter 5 presents some concluding remarks and briefly discusses other problems in computational advertising.

Auctions: The Backwards Indifference Derivation (BID) Algorithm

2.1 Introduction

Auctions have a long and rich history of being used as a market mechanism to buy and sell goods or services. In turn, the diverse range of circumstances in which auctions have been held has helped to shape the field of auction theory by introducing new auction formats into the literature. Even today, the field is undergoing a transformation: due to the rapid growth of the Internet, countless auctions are being held every second in a new online ad exchange environment where advertisers compete for the right to display their Internet ads to users.

Despite how far auction theory has progressed, however, one aspect of the independent private value first-price sealed-bid auction is still not well understood. Specifically, although it is known that the continuous pure strategy Nash equilibrium (PSNE) bidding functions for this auction format exist and are unique when bidders draw their valuations (types) from different continuous and atomless distributions, in general these strategies cannot be analytically derived. Consequently,

much research has been focused on developing numerical algorithms to approximate these PSNE solutions. To date, however, no method has yet been able to establish that it converges to the solution.

In this chapter we propose the Backwards Indifference Derivation (BID) algorithm, which is a backwards-shooting algorithm that is motivated in part by the results derived in Athey (2001). Specifically, the BID algorithm attempts to approximate the continuous PSNE solution through a sequence of finite-action PSNE that are constructed by finding where bidders are “indifferent” between two different actions. Consequently, our approach differs from other popular numerical methods that directly consider a system of poorly behaved differential equations.

Like all of the numerical methods that have been proposed to date, we evaluate the performance of the BID algorithm through the use of numerical examples. We first consider situations where the PSNE solution cannot be analytically derived, and show that our algorithm passes a necessary visual “test” that was recently proposed by Hubbard et al. (2013)—suggesting that our algorithm produces results that are consistent with economic theory. Afterwards, we investigate examples with known closed-form solutions and show that the BID algorithm is capable of handling other kinds of auction asymmetries that are not typically considered by other numerical methods—such as differences in the supports of the type distributions and asymmetries in the bidders’ utility functions. Finally, we use the BID algorithm analyze how ad auctions can be modified in order to generate more revenue.

2.2 Standard Model: Theory and Notation

Suppose that there are N bidders participating in an auction for a single object, where the bidders belong to the set $\mathcal{N} = \{1, 2, \dots, N\}$ and the letter n is used to index the members. We focus on independent private value first-price sealed-bid auctions, where the conditions imply the following:

- **Independent Private Value:** Each bidder only knows his own valuation for the object, and this valuation is not influenced by the other bidders.
- **First-Price:** The highest bidder wins the object and pays the amount that he bids. Losing bidders are no better or worse off.
- **Sealed-Bid:** Bids are submitted simultaneously so that no bidder knows the bid of any other participant.

The n^{th} bidder has a private valuation for the object being bid on, which is modeled as a random variable V_n that is independently drawn from his own type distribution $F_n(v_n)$. For all n , these distributions F_n are assumed to be continuous with strictly positive densities $f_n(v_n) = F'_n(v_n) > 0$ on their supports $\mathcal{T}_n \equiv [\underline{v}_n, \bar{v}_n] \subset \mathbb{R}$, where $\underline{v}_n \geq 0$. Although these distributions F_n are common knowledge, the actual realizations v_n for the valuations are private and known only to bidder n himself.

In the literature, a common and compact support for all of the bidders is usually assumed: $\mathcal{T}_n = [\underline{v}, \bar{v}]$ for all n . For the remainder of this section we adopt this assumption as we develop the standard model. However, we relax this assumption in later sections when we discuss results from Athey (2001) and introduce our own algorithm. Consequently, our algorithm is able to consider situations that have typically been ignored in the auction literature.

At this point, it is helpful to define some additional notation that will allow us to succinctly refer to certain groups of the bidders. We use bolded font without a subscript to denote the N -tuple collecting all N bidders: for example $\mathbf{F} = (F_1, F_2, \dots, F_N)$ to denote all N type distributions. Meanwhile, $(N-1)$ -tuples that collect all bidders except one are denoted with an additional subscript that indicates which bidder has been excluded: for example $\mathbf{F}_{-n} = (F_1, F_2, \dots, F_{n-1}, F_{n+1}, \dots, F_N)$ to denote everyone's type distribution except bidder n .

Using this notation, we now formally define a PSNE solution. The goal of the analysis is to determine each bidder's PSNE bidding function $\beta_n(v_n)$ that prescribes his best response bid b_n as a function of his type v_n . More specifically, the bidding function β_n is bidder n 's best response to β_{-n} if, for all his types $v_n \in \mathcal{T}_n$, β_n maximizes his expected utility given that the other bidders are using the bid functions β_{-n} and that their valuations are drawn according to F_{-n} . A PSNE is then defined as an N -tuple of bidding functions β such that for all n , β_n is a best response function to every other strategy in the collection β_{-n} .

These PSNE bidding functions are known to be nondecreasing in the sense that higher types place (nonstrictly) higher bids. We restrict our attention to situations where the range for the PSNE functions is identical for all bidders $[\underline{b}, \bar{b}]$.¹ If the auctioneer does not set a reserve price r , or if $r \leq \underline{v}$, then the lower bound $\underline{b} = \underline{v}$. In the presence of a reserve price r such that $\underline{v} < r < \bar{v}$, then the lower bound $\underline{b} = r$. A reserve price $r \geq \bar{v}$ leads to a trivial solution where no bidders participate in the auction.

Boundary conditions are also imposed on the PSNE bidding functions. The lower, left-boundary condition is given by $\beta_n(\underline{b}) = \underline{b}$ for all n . The upper, right-boundary condition is given by $\beta_n(\bar{v}) = \bar{b}$ for all n . Meanwhile, all types $v_n < \underline{b}$ will choose not to participate in the auction as they would receive a negative utility should they actually win. Notice that these boundary conditions highlight one other aspect of the problem. Regardless of the presence of a reserve price, the lower boundary \underline{b} is always known ahead of time. The upper boundary \bar{b} , on the other hand, is always *a priori* unknown and must be found.

We will also need to consider the *inverse* PSNE bidding functions for the bidders, which are functions that map bids back to the types that placed them. We will

¹ We continue to maintain this assumption when we move to the case of different supports for the type distributions.

denote these functions as $\phi_n(b_n)$. Observe that the boundary conditions imposed on the PSNE bidding functions β_n imply the following boundary conditions on these inverse functions: $\phi_n(\underline{b}) = \underline{b}$ and $\phi_n(\bar{b}) = \bar{v}$ for all n .

Another standard assumption in the literature is that all bidders are risk neutral.² Therefore, bidder n receives the following payoff when his type is v_n and he submits bid b_n :

$$U_n(v_n, \mathbf{b}) = \begin{cases} v_n - b_n & \text{if } b_n > b_m \text{ for all } n \neq m \\ 0 & \text{otherwise.} \end{cases}$$

That is, bidder n receives a utility of $v_n - b_n$ if he wins the auction by outbidding every other bidder with a bid of b_n , and he receives a utility of 0 if at least one other bidder outbids him.

Meanwhile, the probability that bidder n wins the auction with a bid of b_n is equal to the probability that every other bidder bids below b_n . This is given by:

$$\begin{aligned} Pr(b_n \text{ wins auction}) &= Pr(B_1 < b_n, B_2 < b_n, \dots, B_{n-1} < b_n, B_{n+1} < b_n, \dots, B_N < b_n) \\ &= \prod_{m \neq n} Pr(B_m < b_n) \\ &= \prod_{m \neq n} Pr(\beta_m(V_m) < b_n) \\ &= \prod_{m \neq n} Pr(V_m < \phi_m(b_n)) \\ &= \prod_{m \neq n} F_m(\phi_m(b_n)). \end{aligned} \tag{2.1}$$

Consequently, if bidder n 's type is v_n , then he wants to submit the bid b_n that maximizes his expected utility:

$$\max_{b_n} EU_n(v_n, b_n) = \max_{b_n} \left\{ (v_n - b_n) \prod_{m \neq n} F_m(\phi_m(b_n)) \right\}. \tag{2.2}$$

² Later on we will relax this assumption and consider asymmetries in the utility functions.

2.2.1 Symmetric Auctions

In the case of symmetric auctions, all bidders draw their types from the same distribution (i.e., $F_n(v_n) = F(v)$ for all n) and have identical PSNE functions (i.e., $\beta_n(v_n) = \beta(v)$ and $\phi_n(v_n) = \phi(v)$ for all n). Therefore, when considering symmetric auctions we can drop all of the subscripts in equation (2.2). Furthermore, equation (2.1), which gave the probability of bidder n winning the auction with a bid of b_n , is now equal to $[F(\phi(b))]^{N-1}$ in a symmetric auction. Consequently, the expected maximization problem reduces to:

$$\max_b EU(v, b) = \max_b \left\{ (v - b) [F(\phi(b))]^{N-1} \right\}.$$

Following Hubbard and Paarsch (2014), we obtain the following first order condition by differentiating with respect to b :

$$(v - b)(N - 1) [F(\phi(b))]^{N-2} f(\phi(b))\phi'(b) - [F(\phi(b))]^{N-1} = 0.$$

Next recall that ϕ and β are inverse functions of each other. Therefore, in equilibrium we know that $\phi(b) = v \iff \beta(v) = b$ and that $\phi'(b) = \frac{1}{\beta'(v)}$. Using these facts, we arrive at:

$$\beta'(v) + \beta(v) \frac{(N - 1)f(v)}{F(v)} = \frac{(N - 1)v f(v)}{F(v)}.$$

Notice that the above equation is a linear, first-order differential equation of the form $\frac{dy}{dx} + yp(x) = q(x)$ that can be solved analytically. Using the initial boundary condition $\beta(\underline{v}) = \underline{b}$, the unique symmetric PSNE solution can be derived as:

$$\beta(v) = v - \frac{\int_{\underline{b}}^v F(\tilde{v})^{N-1} d\tilde{v}}{F(v)^{N-1}}. \quad (2.3)$$

Consequently, we see that the symmetric auction problem has a closed-form PSNE solution. The following example illustrates a simple symmetric auction situation.

Example 1

Suppose that we have two symmetric bidders whose valuations are independent random variables drawn uniformly between 0 and 1. That is, $F(v) = v$ where $v \in [0, 1]$. Furthermore assume that there is no reserve price, which implies that the left boundary condition on the PSNE bidding functions is given by $\underline{b} = 0$.

Because the two bidders are symmetric, we know that they have identical PSNE bidding functions. In particular, using (2.3) we know that the PSNE solution is given by:

$$\begin{aligned}\beta(v) &= v - \frac{\int_0^v \tilde{v} d\tilde{v}}{v} \\ &= v - \frac{\frac{v^2}{2}}{v} \\ &= v - \frac{v}{2} \\ &= \frac{v}{2}.\end{aligned}$$

Therefore, the PSNE solution for each bidder is to bid exactly half of their valuation for the object. For example, suppose that bidder 1's type happened to be $\frac{1}{2}$ and bidder 2's type happened to be $\frac{4}{5}$. Then in a PSNE, bidder 1 will place a bid of $\frac{1}{4}$ while bidder 2 places a bid of $\frac{2}{5}$. Consequently, bidder 2 ends up winning the auction and receives a utility of $\frac{2}{5}$. Meanwhile, bidder 1 receives a utility of 0.

2.2.2 Asymmetric Auctions

In the case of asymmetric auctions, however, equation (2.2) cannot be reduced further. From Hubbard and Paarsch (2014), we see that differentiating it with respect to b_n results in the following set of first order conditions for bidders $n = 1, \dots, N$:

$$\phi'_n(b) = \frac{F_n(\phi_n(b))}{f_n(\phi_n(b))} \left\{ \left[\frac{1}{N-1} \sum_{m=1}^N \frac{1}{\phi_m(b) - b} \right] - \frac{1}{\phi_n(b) - b} \right\}, \quad (2.4)$$

with boundary conditions $\phi_n(\underline{b}) = \underline{b}$ and $\phi_n(\bar{b}) = \bar{b}$ for all n . Unfortunately, unlike the symmetric case, this system of differential equations cannot be simplified any further. Furthermore, observe that the boundary conditions themselves are problematic: the upper boundary \bar{b} is *a priori* unknown while the lower boundary $b = \underline{b}$, although known, does not satisfy a Lipschitz continuity condition—resulting in an extremely difficult problem to solve.

Intuitively, a Lipschitz continuous function $g : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is limited in how quickly it can change. Mathematically, we can state this as follows: for every pair of points \mathbf{x} and \mathbf{y} of this function g , there exists some real number $K > 0$ (which is called a “Lipschitz constant”) such that

$$\frac{\|g(\mathbf{y}) - g(\mathbf{x})\|}{\|\mathbf{y} - \mathbf{x}\|} \leq K,$$

where $\|\cdot\|$ is a given vector norm.

Returning to the case of asymmetric auctions, recall that the lower boundary condition is given by $\phi_n(\underline{b}) = \underline{b}$ for all n . Therefore, it can be observed that the system of differential equations given in (2.4) no longer satisfies a Lipschitz condition because some of the denominators vanish at $b_n = \underline{b}$. Consequently, much of the standard theory regarding systems of ordinary differential equations is no longer applicable.

Despite these mathematical difficulties, several important theoretical results have still been proven for asymmetric auctions—most notably that the PSNE solutions exist and are unique. As pointed out in Athey (2001), these proofs have generally followed one of two main approaches:

1. Establishing that a unique solution exists to the system of differential equations given in (2.4) via indirect proofs (Lebrun, 1999; Bajari, 2001).
2. Showing that an equilibrium exists when either types or actions are restricted

to finite sets, and then appealing to limiting arguments (Lebrun, 1996; Maskin and Riley, 2000; Athey, 2001; Jackson and Swinkels, 2005).

Even with these important theoretical results of existence and uniqueness, however, in general there is no closed-form solution to the asymmetric auction problem. Therefore, in parallel to these efforts, much research attention has been focused on developing numerical methods to approximate these asymmetric PSNE solutions.

As noted by Hubbard and Paarsch (2014), the development of these numerical algorithms is particularly important because they can help in the advancement and understanding of economic theory, empirical analysis, and policy evaluation. For example, asymmetries in the type distributions violates one of the conditions of the Revenue Equivalence Theorem, which holds that many types of auctions (e.g., first-price, second-price, all pay) generate the same expected revenue for the auctioneer. However, the actual consequences that may occur under these asymmetries are not yet completely well understood, and numerical investigation may help to provide some clues, intuition, or answers.

We discuss some of the most widely used numerical methods in more detail in the next section.

2.3 Numerical Methods

In this section, we briefly review some of the most popular numerical methods that have been used to investigate the asymmetric auction problem that we are considering. These algorithms typically follow one of three strategies: backwards-shooting algorithms, transformation methods, and projection methods. Despite their differences, as Hubbard and Paarsch (2014) recently noted, they all still have something in common: none of them have yet established that they converge to the truth. This may not be too surprising given that all of these numerical methods consider the

poorly behaved system of differential equations in (2.4). For a more thorough and comprehensive overview of numerical methods, see Hubbard and Paarsch (2014) or the actual papers themselves.

Marshall et al. (1994) pioneered research in this area by being the first to propose using numerical algorithms to approximate the PSNE functions for asymmetric auctions. In their paper, all bidders draw their types from the standard uniform distribution, with the asymmetries in the auction arising when bidders collude to form two coalitions of different sizes—a setup that can be shown to be equivalent to an asymmetric auction where two bidders draw their valuations from different power distributions. To try to solve the system of differential equations in (2.4) while avoiding the “nuisance” solutions resulting from the singularity at the lower boundary, the authors proposed a backwards-shooting method.

Recall that the system of differential equations for asymmetric auctions in equation 2.4 is a two-point boundary value problem where the lower boundary point \underline{b} is known, but the upper boundary point \bar{b} is *a priori* unknown. A backwards-shooting algorithm attempts to solve this type of problem by treating it as an initial value problem. Intuitively, this is done by first fixing the *a priori* unknown right boundary \bar{b} at some initial value (i.e., a guess). Using this fixed initial guess for the maximal bid, the differential equation in (2.4) is then solved backwards using some numerical method (e.g., Euler’s method or Runge-Kutta). The validity of the solution that is obtained is then checked by determining whether it satisfies the known lower boundary condition $\phi_n(\underline{b}) = \underline{b}$ for all n . Finally, depending on the results of this check, the initial guess for \bar{b} is adjusted so that successive approximations are more accurate. In a later section, we discuss how these adjustments to \bar{b} are made in the context of asymmetric auctions.

The specific numerical routine used for the backwards-shooting method in Marshall et al. (1994) is a recursive p^{th} order Taylor series expansion that runs backwards

along a set of equally spaced grid points in $[\underline{b}, \bar{b}]$, where \bar{b} has been fixed at some initial guess as was discussed in the previous paragraph.

Although the examples presented by Marshall et al. are rather simplistic by today's standards, their research has motivated many recent developments in the backwards-shooting literature. Indeed, backwards-shooting methods have been the most researched and most widely used numerical algorithms to date. For example, Bajari (2001) and Li and Riley (2007) showed how the behavior of a given backwards solution with respect to the known lower boundary \underline{b} can be used to adjust the initial guess of the *a priori* unknown upper boundary \bar{b} via a bisection method. Li and Riley also proposed using the Bulirsch-Stoer routine as a more accurate and efficient way of evaluating the system of differential equations. Meanwhile, Gayle and Richard (2008) extended backwards-shooting algorithms to N asymmetric bidders whose type distributions belong to one of four families (the two parameter Weibull, beta, normal, and lognormal), and they introduced a novel search algorithm to find the unknown upper boundary \bar{b} .

Despite the popularity of backwards-shooting methods, however, many authors have noted a numerical instability in their algorithms existing near the left boundary condition $\beta_n(\underline{y}_n) = \underline{b}$ for all n . These observations were recently formalized by Fibich and Gavish (2011), who showed that all backwards-shooting algorithms are inherently unstable for solving asymmetric auctions problems.

Specifically, Fibich and Gavish noted that the upper boundary \bar{b} —even if known or computed exactly (e.g, in the case of symmetric auctions)—can only be numerically approximated to some order of accuracy due to machine round-off error. They denote this computer representation of the maximal bid as $\bar{b}_\epsilon = \bar{b} + \epsilon$ for some error $\epsilon \neq 0$. The authors then showed that using \bar{b}_ϵ as the initial right boundary condition for backwards-shooting instead of the true \bar{b} leads to inherently unstable solutions. Furthermore, this instability worsens as the number of bidders increases. Therefore,

the instability that many authors had observed was not simply a “technical issue” of any single backwards-shooting method, but instead an “inherent analytic property of backwards solutions” that “cannot be eliminated by changing the numerical methodology for backwards integration.”

Consequently, Fibich and Gavish proposed a new approach that transforms the system of differential equations in (2.4) to use one of the bidder’s types v_n as the independent variable instead of b . This transformed system then lies on a fixed known domain (i.e., the support for one of the type distributions), and it has slightly different boundary conditions. More importantly, this transformation allowed Fibich and Gavish to avoid the instability that inherently characterizes backwards-shooting algorithms. The authors then used fixed-point iterations and Newton’s iterations to solve this transformed system.

Nevertheless, Fibich and Gavish noted that their proposed method is not without its own shortcomings: more “research is needed to eliminate the ad-hoc choice of the independent variable,” as the “wrong choice of the independent variable may lead to divergence.” This behavior can even occur in the relatively simple examples that backwards-shooting methods appear to routinely handle.³

Projection methods are the final numerical technique commonly used to approximate asymmetric auction PSNE solutions. This approach attempts to use a finite linear combination of simpler, known basis functions to approximate a more complex, unknown function. Bajari (2001) proposed an algorithm using polynomials as these basis functions. Meanwhile, Hubbard and Paarsch (2009) modified Bajari’s algorithm to instead use Chebyshev polynomials. Furthermore, they recast the problem within the Mathematical Programs with Equilibrium Constraints (MPEC) approach advocated by Su and Judd (2012) and then imposed a monotonicity constraint on

³ The actual example in Fibich and Gavish (2011) where the authors noted this behavior is $F_1(v_1) = v_1$ and $F_2(v_2) = v_2^2$ for $v_n \in [0, 1]$. Notice that this example is equivalent to the coalition situation that was first considered by the backwards-shooting method of Marshall et al. (1994).

candidate solutions. Because projection methods do not need to solve the system of differential equations iteratively, they tend to be faster and more efficient than backwards-shooting algorithms. Furthermore, they avoid the inherent instability of backwards-shooting algorithms. On the other hand, they provide the user with less control over the solutions obtained, and as Hubbard et al. (2013) showed, applications of these methods may not necessarily be flexible enough to approximate some of the more complex PSNE bidding functions—particularly those that may intersect each other.⁴

Despite the large variety of numerical algorithms that have been proposed, as Hubbard and Paarsch (2014) recently noted, they all have something in common:

One weakness of all research in this literature is that all evidence concerning the performance of the proposed approach is purely numerical and done via example: no one has considered analytically the efficiency and convergence properties of the proposed solutions...a shortcoming of this field is that no one has proved that an approach converges to the truth.

Furthermore, observe that these methods also share something else in common: they all consider the poorly behaved system of differential equations given in (2.4). Consequently, the literature in numerical methods has largely paralleled the first approach used to answer the theoretical questions of existence and uniqueness that we mentioned at the end of Section 2.2.

However, as we also pointed out at the end of Section 2.2, there has been another successful approach to answering these theoretical questions—one that avoids having to directly analyze the complicated differential system. Instead, this approach attempts to find PSNE to games with successively finer action sets and appeals to

⁴ That is, PSNE functions that do not exhibit first-order stochastic dominance—a recent area of research in the literature that we will discuss in a later section.

limiting arguments to show that a sequence of these finite-action PSNE converges to the continuum-action PSNE solution. More importantly, and as noted by Athey (2001), this motivates a computational algorithm with a convergence result. Consequently it is rather surprising that, to our knowledge, there is no numerical method in the literature to date that is based on this approach.

2.4 Equilibrium Solution for Auctions with Finite Sets

The existence of an equilibrium solution to the general asymmetric auction problem with N bidders was first established by Lebrun (1996), who proved this result by approximating the original asymmetric auction game with a sequence of finite-action games. Existence of at least one Nash equilibrium (possibly in mixed strategies) in each of these finite-action games is guaranteed by Nash's Existence Theorem. Lebrun then extended this existence result to the continuum-action situation by showing that the limit of Nash equilibria for these approximating games is itself a Nash equilibrium of the original asymmetric auction game.

Meanwhile, Maskin and Riley (2000) used a slightly different approach in establishing their own proof of existence of an equilibrium for asymmetric auction games. Rather than a finite-action space, the authors instead approximated the original game with a sequence of games having finite, discrete types and a continuum-action space. Because Nash's Existence Theorem no longer applies under this situation, an equilibrium solution no longer necessarily exists in each of these games when the standard tie-breaking rule of selecting a winner at random is enforced. However, Maskin and Riley were able to prove the existence of a monotonic equilibria when a second-round Vickrey auction is instead used as the tie-breaking rule. Afterwards, they extended their existence result under this modified tie-breaking rule to the case of a continuous type space by considering the limit of this sequence of games. Finally, they established the existence of a continuous equilibrium under the standard

auction rules and assumptions by arguing that ties will never occur in the continuous case anyway. As a result, the bidding strategies that are obtained in the limit remain best response strategies even when the second-round Vickrey tie-breaking rule is changed. A similar argument is also adopted several years later by Jackson and Swinkels (2005), who proved existence of equilibria for a wide class of private value auctions under even more general tie-breaking rules—such as allowing the auctioneer to break ties by using information that he would not normally have (e.g., the true values of the bidders).

The final paper adopting the auction with finite sets approach that we consider in this chapter is Athey (2001). As this is also the paper that our proposed algorithm will most closely follow, we devote the following subsection to its discussion.

2.4.1 Athey (2001)

Unlike the papers that we have discussed so far, Athey (2001) allowed the utility functions and the supports \mathcal{T}_n for the type distributions F_n to be different across bidders. As we shall later see, this allows us to generalize our proposed algorithm to cases that other numerical methods are unable to handle or have not yet considered.

Furthermore, rather than directly address the question of the existence of a PSNE, Athey instead considered the simpler question of whether the single crossing condition (SCC) holds. The SCC was first introduced by Karlin (1968), and later had its definition generalized by Milgrom and Shannon (1994) as follows:

Definition 1 (Milgrom and Shannon (1994)). *The function $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ satisfies the single crossing condition in (x, θ) if for all $x' > x$ and $\theta' > \theta$:*

$$h(x', \theta) > h(x, \theta) \Rightarrow h(x', \theta') > h(x, \theta')$$

and

$$h(x', \theta) \geq h(x, \theta) \Rightarrow h(x', \theta) \geq h(x, \theta').$$

For our purposes, however, it is enough to describe the underlying intuition behind the SCC as follows: whenever each opponent uses a nondecreasing strategy (i.e., higher types choose nonstrictly higher actions), a player’s best response is to also use a nondecreasing strategy. From our discussion in Section 2.2, it is clear that the auction problem that we are considering satisfies the SCC.

Athey then noted that any nondecreasing strategy is a step function whenever the set of available actions is finite. Therefore, any nondecreasing strategy for bidder n can be described by simply enumerating the types at which he “jumps” from one action to the next higher action.

More specifically, consider the following representation. Let $\mathcal{B}^M = \{b^0, b^1, \dots, b^M\}$ denote the set of actions (bids) in ascending order that are available to all of the bidders, where $M + 1$ is the number of possible actions. Notice that here we have assumed a common action set for all bidders.⁵ Furthermore, because participation in a finite-action auction is assumed to be voluntary, the action b^0 denotes the choice to not participate in the auction—a decision which always provides a fixed, certain utility of zero.

Next Athey defined

$$\Sigma_n^M \equiv \{ \mathbf{v}_n \in \mathcal{T}_n^{M+2} \mid v_n^0 = \underline{v}_n \leq v_n^1 \leq \dots \leq v_n^M \leq v_n^{M+1} = \bar{v}_n \}, \quad (2.5)$$

and she let $\Sigma^M = \Sigma_1^M \times \dots \times \Sigma_N^M$. The relationship between a vector $\mathbf{v}_n \in \Sigma_n^M$ and *any* nondecreasing strategy for bidder n , $\alpha_n^M : \mathcal{T}_n \rightarrow \mathcal{B}^M$, can then be described as follows:

⁵ Athey (2001) also initially made this assumption for notational simplicity. Eventually this condition is relaxed and she proves existence for the more general case of bidder’s with different action sets.

Definition 2 (Athey (2001)). A nondecreasing strategy for bidder n , $\alpha_n^M : \mathcal{T}_n \rightarrow \mathcal{B}^M$ is represented by a vector $\mathbf{v}_n \in \Sigma_n^M$ if:

1. $v_n^m = \inf \{v_n | \alpha_n^M(v_n) \geq b^m\}$ whenever there is some $k \geq m$ such that $\alpha_n^M(v_n) = b^k$ on an open interval of \mathcal{T}_n , and $v_n^m = \bar{v}_n$ otherwise; and
2. $\alpha_n^M(v_n) = b^{\max\{m | v_n^m < v_n\}}$ for all $v_n \in \mathcal{T}_n^M \setminus \{\mathbf{v}_n\}$.

Therefore, each component of the vector $\mathbf{v}_n \in \Sigma_n^M$ can be regarded as a “jump point” of the step function described by a bidding strategy α_n^M for bidder n . This can be visualized in Fig. 2.1. Notice, however, that \mathbf{v}_n can be consistent with more than one nondecreasing strategy because it does not specify the exact behavior for types $v_n \in \{\mathbf{v}_n\}$. Because the type distributions are assumed to be atomless, however, we can ignore this technicality as these points have measure zero and do not affect the bidding behavior of the other players.

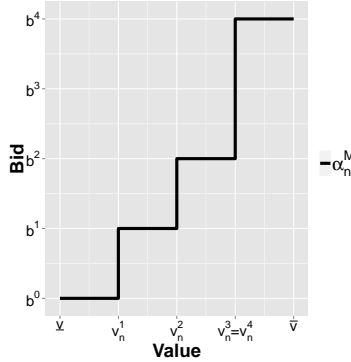


FIGURE 2.1: A stylistic example of a nondecreasing finite-action step function strategy α_n^M that specifies when bidder n “jumps” to a higher action. Notice that Definition 2 allows for some jump points to be equal. So, for this example $v_n^3 = v_n^4$ implies that the probability of action b^3 being used is 0. When we propose our own algorithm in Section 2.6, however, we impose an additional constraint that prevents this from happening.

Under the standard auction assumptions, Athey was able to show that if the SCC holds in a game where the action space is finite for all bidders, then there exists a

fixed point in Σ^M . Consequently, this implies that the game has a PSNE where each player's equilibrium strategy is nondecreasing. We denote these PSNE bidding functions as β_n^M for all n to distinguish them from any other nondecreasing strategy α_n^M that may not necessarily be a PSNE bidding function.

Furthermore, by carefully accounting for discontinuities that arise in the limit as the finite bidding units become increasingly small (i.e., as $M \rightarrow \infty$), Athey was able to establish the existence of a PSNE β^* to the continuum-action auction game as the limit of a sequence of PSNE to finite-action games $\{\beta^M\}$. More formally, she accomplished this by first proving that $\{\beta^M\}$ converges uniformly to β^* except on a set of arbitrarily small measure. This result, coupled with the fact that each β^M is a PSNE, allowed Athey to argue that no mass points arise in the limit. Hence β^* is indeed a PSNE of the continuum auction game.

In addition, recall that Athey was able to establish her existence result without requiring many of the assumptions commonly imposed by others. Consequently, her results extend to more general asymmetric auction games that had largely been ignored—such as situations where bidders have different supports for their type distributions or where bidders have heterogeneous utility functions.

More importantly, and as Athey went on to note, her convergence result was particularly significant because it motivated a computational algorithm: compute PSNE to games with successively finer action sets. Notice that this approach differs greatly from the strategies of the numerical methods discussed in Section 2.3 that consider the poorly behaved system of differential equations given in (2.4). Furthermore, this algorithm would also obtain the convergence property that has been so elusive in the literature.

Of course, creating the algorithm that is suggested by Athey's results is easier said than done. Recall that the PSNE for any finite-action auction game is a fixed point

solution to a nonlinear system of equations. As pointed out by Athey, however, finding this fixed point is rather difficult since there is no global “contraction mapping” theorem. Therefore, the simplest fixed point iteration methods are not guaranteed to converge (and in practice they tend to perform rather poorly). Consequently, another approach is needed.

In the next section we introduce some of the intuition behind our own algorithm that is motivated in part by the results derived in Athey (2001). We do this by showing the connection between Athey’s jump points and the bidders being exactly indifferent between two different actions.

2.5 Points of Indifference

Recall that Athey (2001) showed that every finite-action auction has a PSNE in nondecreasing strategies where, in order to describe a nondecreasing strategy, we need only specify a step function whose jump points correspond to the types at which a player switches from one action to the next higher action. In this section, we investigate this behavior from the actual decision process that a bidder faces. In particular, we consider points where a bidder is exactly indifferent between two actions. This perspective helps to motivate the algorithm that we propose in the next section.

We begin by considering the following scenario. Suppose that bidder n is trying to decide which of two bids he would prefer: b^{m-1} or b^m , where $b^{m-1} < b^m$. No matter what his actual type v_n is, as a rational agent he will make his decision by evaluating the expected utility that each bid provides. In particular, the relationship between the preferences of a bidder with type v_n and his expected utilities will behave as

follows:

$$\begin{aligned}
b^{m-1} > b^m &\iff (v_n - b^{m-1}) \cdot Pr(b^{m-1} \text{ wins}) > (v_n - b^m) \cdot Pr(b^m \text{ wins}), \\
b^{m-1} < b^m &\iff (v_n - b^{m-1}) \cdot Pr(b^{m-1} \text{ wins}) < (v_n - b^m) \cdot Pr(b^m \text{ wins}), \quad (2.6) \\
b^{m-1} \sim b^m &\iff (v_n - b^{m-1}) \cdot Pr(b^{m-1} \text{ wins}) = (v_n - b^m) \cdot Pr(b^m \text{ wins}).
\end{aligned}$$

Next, let us suppose that there exists a type $v_n^m \in \mathcal{T}_n$ where bidder n is exactly indifferent between actions b^{m-1} and b^m . Therefore, using (2.6) we know that the following equality holds at this point:

$$(v_n^m - b^{m-1}) \cdot Pr(b^{m-1} \text{ wins}) = (v_n^m - b^m) \cdot Pr(b^m \text{ wins}). \quad (2.7)$$

But notice that (2.7) can only hold if $Pr(b^{m-1} \text{ wins}) < Pr(b^m \text{ wins})$ is true. Therefore, for all $\epsilon > 0$, the following inequality must also be true:

$$\epsilon \cdot Pr(b^{m-1} \text{ wins}) < \epsilon \cdot Pr(b^m \text{ wins}). \quad (2.8)$$

Adding (2.7) to (2.8) gives:

$$(v_n^m + \epsilon - b^{m-1}) \cdot Pr(b^{m-1} \text{ wins}) < (v_n^m + \epsilon - b^m) \cdot Pr(b^m \text{ wins}), \quad (2.9)$$

which implies that all of bidder n 's types $v_n > v_n^m$ strictly prefer bidding b^m to b^{m-1} .

Similarly, it is easy to show that for all $\epsilon > 0$:

$$(v_n^m - \epsilon - b^{m-1}) \cdot Pr(b^{m-1} \text{ wins}) > (v_n^m - \epsilon - b^m) \cdot Pr(b^m \text{ wins}). \quad (2.10)$$

So, all of bidder n 's types $v_n < v_n^m$ strictly prefer bidding b^{m-1} to b^m .

Therefore, in addition to being a point of indifference, v_n^m is also the exact point where bidder n jumps from action b^{m-1} to b^m . Furthermore, v_n^m is the only point for bidder n where this behavior can occur.

And although the fact that bidder n is indifferent between b^{m-1} and b^m at v_n^m implies that we may not necessarily be able specify his optimal action at this point,

we do not need to. Like the jump points in Athey (2001), his behavior at this point of indifference has measure zero and will not affect the best responses of other bidders.

In the next section, we introduce our algorithm that is largely motivated by this idea. Specifically, it attempts to construct a finite-action PSNE by iteratively solving a nonlinear system of equations to find these points of indifference.

2.6 Proposed Algorithm: The Backwards Indifference Derivation (BID) Algorithm

In this section, we propose the Backwards Indifference Derivation (BID) algorithm, which is a backwards-shooting algorithm that is motivated in part by the results of Athey (2001) and the intuition described in Section 2.5. Specifically, the goal of the BID algorithm is to construct a finite-action PSNE by finding where bidders are indifferent between two different actions.

Similar to Athey, given the type distributions \mathbf{F} and the reserve price r , the finite-action PSNE that is constructed by the BID algorithm is one that is characterized by:

- A finite set of potential actions that are commonly available to all bidders:

$$\mathcal{B}^M = \{b^0, b^1, \dots, b^M\}, \quad (2.11)$$

where the actions are indexed in ascending order.

- A vector $\mathbf{v}_n \in \hat{\Sigma}_n^M$ for each bidder n specifying the types at which he jumps from one action to the next higher action, where

$$\hat{\Sigma}_n^M \equiv \{\mathbf{v}_n \in \mathcal{T}_n^{M+2} | v_n^0 = \underline{v}_n < v_n^1 < \dots < v_n^M < v_n^{M+1} = \bar{v}_n\}. \quad (2.12)$$

- A bidding function β_n^M for each bidder n that corresponds to his vector of jump

points v_n as follows:

$$\beta_n^M(v_n) = b^0 \mathbf{1}_{[v_n^0, v_n^1]}(v_n) + \sum_{m=2}^{M+1} b^{m-1} \mathbf{1}_{(v_n^{m-1}, v_n^m]}(v_n), \quad (2.13)$$

where $v_n \in \mathcal{T}_n$, and $\mathbf{1}_A(v_n)$ is the indicator function that is equal to 1 if $v_n \in A$ and equal to 0 otherwise.

Notice that, unlike Athey, we are requiring the vector of jump points to contain distinct components—which is why we have defined the new set of vectors in $\hat{\Sigma}_n^M$ using strict inequalities. Therefore, in the finite-action PSNE constructed by the BID algorithm, each action in \mathcal{B}^M will be used by at least some types v_n for all n .⁶

Our algorithm attempts to numerically approximate nontrivial continuous PSNE bidding functions where the codomain $[\underline{b}, \bar{b}]$ is the same for all bidders who actually choose to participate in the auction.⁷ This includes the standard model that we described in Section 2.2 where the support for the type distributions is identical, but also encompasses some situations that are less well understood and have largely been ignored in the literature. In particular, like Athey, we are allowing the support \mathcal{T}_n for the type distributions F_n to be different across the bidders.⁸

Consequently, it is necessary to impose boundary constraints on the PSNE solutions that are more general than the ones we considered in Section 2.2. Like the standard model, we assume that the lower bound \underline{b} is always *a priori* known, but that the upper bound \bar{b} may be potentially unknown. We also assume that $\bar{v}_n > \underline{b}$ for all n , as bidders who never participate in the auction can be excluded from the analysis.

⁶ This is a consequence of how the BID algorithm constructs the PSNE. Note that Definition 2, which is based on Athey (2001), allows for there to be actions which are entirely ignored by a bidder and never used.

⁷ An example of a bidder who would always choose not to participate in the auction would be someone whose maximal type is less than the reserve price (i.e., $\bar{v}_n < r$).

⁸ This situation leads to a slightly different system of differential equations than the one described in (2.4), which may explain why other numerical methods ignore this situation. However, Athey (2001) and Lebrun (2006) show that a PSNE still exists in these cases.

Meanwhile, the boundary conditions imposed on the PSNE functions are as follows. The right-boundary condition is $\beta_n(\bar{v}_n) = \bar{b}$ for all n . The left-boundary condition for bidders with $v_n < \underline{b}$ is $\beta_n(\underline{b}) = \underline{b}$, while bidders with $v_n \geq \underline{b}$ have $\beta_n(v_n) = \underline{b}$ as their relevant boundary condition. Like always, all types $v_n < \underline{b}$ choose not to participate in the auction. Notice that these constraints reduce to those described in Section 2.2 when the standard model with identical supports is being assumed.

We also need some method of accounting for ties. The BID algorithm considers a modified auction rule where all of the highest bidder(s), and only the highest bidder(s), receive the object (or some other substitute offering the same exact payoff). This modified rule still preserves the SCC. Therefore, by our earlier discussion of Athey (2001), we know that a PSNE still exists for any finite-action auction operating under this modified auction rule, and we eventually prove that our algorithm does in fact construct a finite-action PSNE under these assumptions.

Admittedly, the tie breaking rule that we are considering is quite nonstandard in the literature. One can perhaps argue that, when we pass to the limit to the continuous case, the probability of a tie occurring has measure zero—which would imply that the tie-breaking rule was irrelevant to begin with. Indeed, as we have already mentioned in Section 2.4, this approach has previously been used by others in this space to establish properties of asymmetric auctions as the limit of finite-set auctions being conducted under other highly unusual tie-breaking rules (Maskin and Riley, 2000; Jackson and Swinkels, 2005). We do not, however, actively nor rigorously try and argue this point here. Instead, like all prior research in this area, we offer what we believe to be compelling numerical evidence regarding the effectiveness of the BID algorithm. This evidence includes passing a recently proposed necessary visual “test” that other methods have failed, and arriving at results that agree with known closed-form solutions to auction asymmetries that other methods have been unable to consider.

It is helpful to think about the BID algorithm in two parts: a core inner part whose purpose is to construct the finite-action PSNE and an outer shell part that improves the accuracy of the algorithm by adjusting the guess for the true maximal bid \bar{b} . Consequently, we introduce the BID algorithm by first discussing the details of the core algorithm in the next subsection. Afterwards, we describe how the outer algorithm can be used to search for the unknown high bid. Finally, we prove that the results of the BID algorithm do indeed characterize a finite-action PSNE under the auction rules that we are considering.

2.6.1 Core Algorithm: Constructing Finite-Action PSNE

Recall that we are trying to find the solution to the asymmetric auction problem where the codomain $[b, \bar{b}]$ for the PSNE bidding functions β_n is the same for all n and where, in practice, the true maximal bid \bar{b} is *a priori* unknown. Fortunately, this problem has been extensively discussed in the backwards-shooting literature—a class that the BID algorithm falls into. Therefore, for the purposes of discussing the BID algorithm we first treat \bar{b} as being known in the core algorithm. Later we show how the outer algorithm can, in theory, be used to improve the accuracy of our guess for \bar{b} to within some accuracy $\gamma > 0$ of the truth.

The core algorithm begins by first designating one of the bidders as the “reference bidder” in the sense that it uses a partitioned set of “seed points” along the support of his type distribution to help guide the process along. Although the actual finite-action PSNE that is constructed by the BID algorithm will vary somewhat depending on which bidder is used as the reference, the process of the core algorithm and the fact that it produces a finite-action PSNE will remain unchanged. For these reasons, along with the fact that we are more interested in the limiting behavior of these finite-action PSNE, we proceed by using bidder 1 as the reference bidder.

Similarly, in general the seed points used to partition the reference bidder’s sup-

port do not necessarily have to be equally spaced. For the purposes of discussing the core algorithm, however, it is convenient to treat them as being equally spaced with a step size of h :

$$\{\underline{v}_1, \underline{v}_1 + h, \underline{v}_1 + 2h, \dots, \bar{v}_1 - h, \bar{v}_1\}. \quad (2.14)$$

After the reference bidder and his seed points have been set, the core algorithm proceeds by setting $b^M = \bar{b}$ and $v_n^{M+1} = \bar{v}_n$ for all n . This step simply initializes the strategies to ensure that all bidders will satisfy the desired upper boundary condition: $\beta_n^M(\bar{v}_n) = \bar{b}$ for all n .

Next assume that, for some integer index m , we know the values b^m and v_n^{m+1} for all n . At this point the goal of the algorithm is to try to find a new, smaller action b^{m-1} and a new, smaller set of jump points $v_1^m, v_2^m, \dots, v_N^m$ that maintains and extends the PSNE structure that has already been constructed up to this point.

This can only be accomplished if, for all n , these new set of points v_n^m truly are points where the bidders will switch from bidding this new action b^{m-1} to the higher action b^m that we are assuming as known. By our discussion in Section 2.5, we treat this process as trying to find the types v_n^m where each bidder is exactly indifferent between bidding b^{m-1} and b^m . This can be mathematically expressed by setting the expected utilities that the bidders receive from these actions equal to each other:

$$\begin{cases} (v_1^m - b^{m-1}) \cdot Pr(1 \text{ wins with } b^{m-1}) & = (v_1^m - b^m) \cdot Pr(1 \text{ wins with } b^m) \\ (v_2^m - b^{m-1}) \cdot Pr(2 \text{ wins with } b^{m-1}) & = (v_2^m - b^m) \cdot Pr(2 \text{ wins with } b^m) \\ & \vdots \\ (v_N^m - b^{m-1}) \cdot Pr(N \text{ wins with } b^{m-1}) & = (v_N^m - b^m) \cdot Pr(N \text{ wins with } b^m), \end{cases} \quad (2.15)$$

where we are assuming that b^m and v_n^{m+1} are known for all n . This implies that we can find the exact probability of each bidder n winning the auction with a bid of b^m .

To see why, recall that we are attempting to construct a finite-action PSNE such that v_n^{m+1} is the point where all bidders n switch from bidding b^m to b^{m+1} . But

since strategies must be nondecreasing, this then implies that all types $v_n < v_n^{m+1}$ for all bidders n must bid b^m or lower. Hence, $F_n(v_n^{m+1})$ gives the probability of any particular bidder n placing a bid of b^m or lower. So, under the auction rules that we laid out earlier, bidder 1 wins the auction with a bid of b^m if everyone else bids b^m or lower. Therefore, $Pr(1 \text{ wins with } b^m) = \prod_{n \neq 1} F_n(v_n^{m+1})$.⁹ Similar arguments can be used to find the probability of any of the other bidders winning with a bid of b^m .

Next observe that, because we want the values v_n^m to be the points where bidders jump from b^{m-1} to b^m , we can also apply this same reasoning to infer what the probability of any bidder n winning with b^{m-1} will be. For example, the relevant probability for bidder 1 would be $Pr(1 \text{ wins with } b^{m-1}) = \prod_{n \neq 1} F_n(v_n^m)$.

Substituting all of these probabilities into (2.15) gives the following nonlinear system of expected utility equations:

$$\left\{ \begin{array}{l} (v_1^m - b^{m-1}) \cdot \left[\prod_{n \neq 1} F_n(v_n^m) \right] = (v_1^m - b^m) \cdot \left[\prod_{n \neq 1} F_n(v_n^{m+1}) \right] \\ (v_2^m - b^{m-1}) \cdot \left[\prod_{n \neq 2} F_n(v_n^m) \right] = (v_2^m - b^m) \cdot \left[\prod_{n \neq 2} F_n(v_n^{m+1}) \right] \\ \vdots \\ (v_N^m - b^{m-1}) \cdot \left[\prod_{n \neq N} F_n(v_n^m) \right] = (v_N^m - b^m) \cdot \left[\prod_{n \neq N} F_n(v_n^{m+1}) \right], \end{array} \right. \quad (2.16)$$

where we are treating the action b^m and the N points v_n^m as the unknowns to be solved for. Therefore, we have N equations and $N + 1$ unknowns.

Before solving the system in (2.16), however, we have to impose the following constraints in order to ensure that the solution we arrive at exhibits the behavior

⁹ Note that this is not quite the exact argument when $m = M$, as the points $v_n^{M+1} = \bar{v}_n$ simply denote the upper bound on the support for their type distributions and no longer represent a point where the bidders jump to a higher action. In this case the expression still holds, however, because under the auction rules we are considering, bidder 1 can guarantee that he wins the auction by bidding $b^M = \bar{b}$. This agrees with the probability of winning the auction with a bid of $b^M = \bar{b}$ given in the expression: $Pr(1 \text{ wins with } \bar{b}) = \prod_{n \neq 1} F_n(v_n^{M+1}) = \prod_{n \neq 1} F_n(\bar{v}_n) = 1$.

that is required of a PSNE:

$$\begin{aligned}
 b^{m-1} &< b^m, \\
 \underline{v}_n &< v_n^m < v_n^{m+1} \text{ for all } n, \\
 b^m &\leq v_n^m \text{ for all } n.
 \end{aligned} \tag{2.17}$$

The first constraint, $b^{m-1} < b^m$, simply highlights the fact that we are only interested in solutions that yield a smaller action. Similarly, $\underline{v}_n < v_n^m < v_n^{m+1}$ for all n , is imposed to make sure that the new jump point that we find for each bidder n is smaller than the last jump point v_n^{m+1} that was found for him, while still lying in the support for his type distribution.¹⁰ The third set of constraints, $b^m \leq v_n^m$ for all n , ensures that v_n^m is only considered a valid jump point solution from action b^{m-1} to b^m if the bidders receive a nonnegative payoff (because otherwise they would be better off not participating in the auction).

Finally, we impose an additional ‘‘pseudo constraint’’ on bidder 1 (the reference bidder) to help guide the algorithm. Let i be a positive indexing integer where we first let $i = 1$. Then before solving the nonlinear system in (2.16), the core algorithm provisionally sets $v_1^m = v_1^{m+1} - i \cdot h$. Intuitively, we can think of this as follows. Suppose that, for reasons other than expected utility, all bidders highly value the added flexibility of having two equivalent actions to choose from if they are able to do so. In particular, bidder 1 enjoys this freedom along his set of seed points given in (2.14). Therefore, knowing that v_1^{m+1} was the last jump point for bidder 1 that was found, the BID algorithm looks to accommodate his request by using another one of his seed points (i.e., $v_1^m = v_1^{m+1} - i \cdot h$, for some i) to solve the nonlinear system given in (2.16). Here standard numerical routines (e.g., Newton-Raphson, secant method, etc.) can be used to try to find numerical solutions to this system.

¹⁰ Notice that this condition also means that the interval (v_n^m, v_n^{m+1}) will be nonempty for each bidder, guaranteeing that there exist at least some types for each bidder whose best response is to actually bid b^m .

If we can facilitate his request by finding a solution to (2.16), then we have determined a new action b^m and a new set of types v_n^m for all n such that everyone is simultaneously indifferent between choosing b^{m-1} and b^m . In this case, we have successfully found what we have been looking for. Therefore, we keep these values and move on to the next iteration of the algorithm, where we search for additional indifference points using the next largest seed point that bidder 1 has available (i.e., by setting $i = 1$).

Of course there may be times when we cannot satisfy his wishes for this temporarily fixed seed point v_1^m , which is a situation that occurs when a numerical solution to (2.16) cannot be found. In this case, the algorithm attempts to do the next best thing: no new values are kept, the index i is incremented up by one, and we try to fulfill bidder 1's request by temporarily fixing $v_1^m = v_1^{m+1} - i \cdot h$ at the next largest seed point that he has available.

This iterative process of trying to find the points of indifference continues as long as $v_1^{m+1} - i \cdot h > \underline{v}_1$. This stopping point for the core algorithm can be attributed to the fact that, by Athey (2001), we know that for all n and all $\delta > 0$ there exists a finite-action PSNE where each bidder type on $v_n \in [\underline{v}_n, \underline{v}_n + \delta]$ is required to use action b^0 and opt not participate in the auction.

When the core algorithm concludes, we are left with a finite increasing set of actions $\{b^1, b^2, \dots, b^M = \bar{b}\}$ and an increasing set of jump points $\{v_n^2, v_n^3, \dots, v_n^{M+1} = \bar{v}_n\}$ for all n that solve a sequence of the nonlinear systems given in (2.16). Notice that, although we have briefly mentioned it in the previous paragraph, the choice b^0 to not participate in the auction has not yet been formally included in the action set. Furthermore, the set of values v_n^0 and v_n^1 for each bidder have also not been added to the set of jump points. These values are accounted for in the outer algorithm that we will be discussing in the next subsection.

Finally, observe that the cardinality of the set of seed points is always at least

as large as the resulting set of jump points that the core algorithm has constructed for each bidder n . This is a consequence of what occurs when the core algorithm fails to keep any new values when it cannot find a solution to the nonlinear system given in (2.16). Therefore, the actual value of M is unknown until the algorithm is finished running. This same reasoning also applies to the index m , whose actual value is unknown during each iteration. These reasons may partially excuse the slightly clumsy use of our notation for these indices.

In summary, the core part of the BID algorithm proceeds as follows

Core Algorithm

1. Initialize $b^M = \bar{b}$ and $v_n^{M+1} = \bar{v}_n$ for all n .
2. Initialize the indexing integer $i = 1$.

WHILE $v_1^{m+1} - i \cdot h > \underline{v}_1$, where m always indexes the most recently kept value for b^m :

 - (a) Provisionally set $v_1^m = v_1^{m+1} - i \cdot h$.
 - (b) Attempt to solve the nonlinear system of equations given in (2.16) subject to the constraints given in (2.17).
 - (c) IF a numerical solution is found:

Set $i = 1$.

Keep the solved values for b^{m-1} and v_n^m for all n .

ELSE:

Set $i = i + 1$.

In the next subsection we describe the outer part of the BID algorithm that allows us to approximate the true unknown maximal bid \bar{b} .

2.6.2 Outer Algorithm: Estimating \bar{b}

Recall that the core algorithm assumed that the maximal bid \bar{b} for the right boundary condition was known. In reality, however, this assumption will rarely hold—particularly in the case of asymmetric auctions—and typically one must guess the true value of \bar{b} before implementing a numerical algorithm. Fortunately, much research has been devoted to improving the accuracy of this guess. This section describes the one that is implemented in the BID algorithm, allowing it to find an interval of length γ containing the true highest bid \bar{b} .

Recall that a backwards-shooting algorithm attempts to solve a two-point boundary value problem by approaching it as an initial value problem. In the case of the asymmetric auction problem, for all n the *a priori* unknown upper boundary $\beta_n(\bar{v}_n) = \bar{b}$ is treated as the initial condition while the *a priori* known lower boundary condition $\beta_n(\underline{v}_n) = \underline{b}$ is treated as the target condition. By fixing \bar{b} at some initial guess, solving the system backwards, and then checking whether the target condition is satisfied, a backwards-shooting algorithm makes adjustments so that subsequent guesses for \bar{b} are closer to the truth—provided that there is some way of knowing what adjustments actually need to be made when the target condition is not satisfied. Luckily, recent developments in the area of asymmetric auctions that characterize the behavior of a solution when it has an incorrectly initialized \bar{b} allow us to systematically make these adjustments.

Observe that any guess for the true \bar{b} can be off in exactly one of two ways: it can either be too high or too low. When the guess is too high, Hubbard and Paarsch (2014) noted that solutions to the system of differential equations (2.4) never reach the known target condition $\beta_n(\underline{v}_n) = \underline{b}$. Instead, the system approaches the 45° line and the bidders' optimal bids start to approach their valuations: $\beta_n(v_n) \rightarrow v_n$. If we look at this behavior from the perspective of the inverse bidding functions,

$\phi_n(b_n) \rightarrow b_n$, we can see why the solution never reaches the target condition: some of the denominators in (2.4) vanish and the system approaches a singularity.

Consider what happens in the core part of the BID algorithm when the guess for \bar{b} is too high. If bids begin to approach their valuations, then for some iteration of step 2 we will eventually be attempting to solve (2.16) by treating $b^m \approx v_n^{m+1}$ as the known values for the bidders. In particular, our reference bidder 1 will have $b^m \approx v_1^{m+1}$. The core algorithm will then be attempting to find solutions to (2.16) while temporarily fixing $v_1^m = v_1^{m+1} - i \cdot h < b^m$ for $i = 1$. Recall, however, that one of the constraints placed on the system is $b^m \leq v_1^m$, which ensures nonnegative utilities. Hence not only will no solution to the system be found for this attempt, but subsequent attempts are also guaranteed to fail because the algorithm will always be temporarily fixing a seed point $v_1^m < b^m$. Consequently, at the end of the core algorithm we are left with the smallest action that we have constructed being larger than the known left boundary condition \underline{b} . Therefore, if $b^1 > \underline{b}$, then we know that we have guessed \bar{b} to be too high.

However, if our guess for \bar{b} is too low, then Fibich and Gavish (2011) showed that the solution to (2.4) diverges in the sense that $\beta_n(v_n) \rightarrow -\infty$ for all n . In this situation, the core part of the BID algorithm will conclude with the smallest constructed action being smaller than the known left boundary condition.¹¹ Therefore, if $b^1 < \underline{b}$, then we know that we have guessed \bar{b} to be too low.

The following simple example illustrates how this behavior is expressed in our core algorithm.

Example 1 (Continued)

We return to the symmetric two bidder example with $F(v) = v$, where $v \in [0, 1]$. Earlier we saw that the PSNE solution had a closed-form that was given by $\beta(v) = \frac{v}{2}$.

¹¹ This is the reason why we did not include the additional constraint that $b^m > \underline{b}$ to (2.17).

Therefore, the true maximal bid is known to be $\bar{b} = \frac{1}{2}$. Meanwhile, the minimal bid (i.e., the known target condition) is $\underline{b} = 0$.

Fig. 2.2 demonstrates the behavior of the core algorithm when it was initialized at different guesses for the maximal bid, and when a step size of $h = 10^{-4}$ was used. When the algorithm was initialized at the true value of \bar{b} , it produced a result that agrees with the known PSNE solution. When the guess for the maximal bid was too high, as was the case with \bar{b}_H , the result approached the 45° line and the algorithm never reached the known target condition $\underline{b} = 0$. When the guess for the maximal bid was too low, which occurred with \bar{b}_L , then $\beta(v) \rightarrow -\infty$. This behavior agrees with the theoretical results and allows us improve our guess of \bar{b} after each run of the core algorithm.

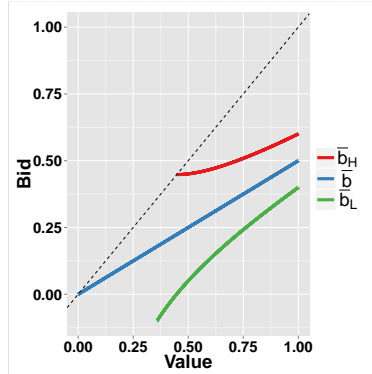


FIGURE 2.2: The behavior of the core part of the BID algorithm when it was initialized at different guesses for the maximal bid. Earlier we showed that PSNE solution is given by $\beta(v) = \frac{v}{2}$ and that the true maximal bid is known to be $\bar{b} = \frac{1}{2}$. Initializing the algorithm at this true value produced results that agree with the known PSNE solution. When initialized too high at \bar{b}_H , however, the results approached the 45° line. When initialized too low at \bar{b}_L , results diverged to $-\infty$.

In particular, as Li and Riley (2007) and Bajari (2001) showed, for a given accuracy of $\gamma > 0$ it is possible to use this behavior to create a binary search method to determine an interval $[\bar{b}_L, \bar{b}_U]$ of length less than γ that will (at least theoretically) contain the true maximal bid \bar{b} . Specifically, first let $\bar{b}_L = \underline{b}$ and $\bar{b}_U = \max_n \{\bar{v}_n\}$ be

the initial lower and upper bounds on our guess for the true \bar{b} , respectively.¹² The complete BID algorithm then proceeds as follows:

BID Algorithm

1. WHILE $\bar{b}_U - \bar{b}_L > \gamma$ OR $b^1 < \underline{b}$
 - (a) Set $\bar{b} = \frac{\bar{b}_L + \bar{b}_U}{2}$.
 - (b) Run the **Core Algorithm**, which gives $\{b^1, b^2, \dots, b^M\}$.
 - (c) IF $b^1 > \underline{b}$:
 - Set $\bar{b}_U = \bar{b}$.
 ELSE:
 - Set $\bar{b}_L = \bar{b}$.
2. Set $v_n^0 = \underline{v}_n$ and $v_n^1 = b^1$ for all n .
3. Add the choice b^0 to the set of available actions.

At the end of the BID algorithm, we have constructed a finite set of actions \mathcal{B}^M , a vector of jump points $\mathbf{v}_n \in \hat{\Sigma}_n^M$ for each bidder n , and a bidding function β_n^M for each bidder n that corresponds to his vector of jump points \mathbf{v}_n as defined in (2.13).

Here there are several things worth mentioning. First, computational time for each full run of the core algorithm increases as the step size decreases. Consequently, the computational time can be improved if these earlier iterations of the core algorithm are run with larger step sizes to give a general sense of where the true \bar{b} is. However, the reader should be aware that the interval $[\bar{b}_L, \bar{b}_U]$ only provides us with a theoretical error bound of γ . As noted by Li and Riley (2007), there are other

¹² Observe that the true \bar{b} must lie in this interval due to the boundary constraints being imposed on the PSNE solution.

sources of error that can increase γ —such as computational inaccuracies that arise when evaluating functions (e.g., the type distribution) or solving the nonlinear system in (2.16). Therefore, the interval constructed for one step size may not necessarily overlap with the interval constructed for another step size. Consequently, caution should be used if the reader decides to gradually decrease the step size for subsequent runs of the core algorithm.

Next, observe that in step 1 of the BID algorithm we have included $b^1 < \underline{b}$ as one of the conditions for the WHILE loop. This simply ensures that we have $b^1 > \underline{b}$ when the entire BID algorithm finishes running (otherwise the PSNE solution would not make sense).

Finally, recall that the choice b^0 to not participate in the auction and the jump points v_n^0 and v_n^1 for all n were conspicuously absent at the conclusion of the core algorithm. Steps 2 and 3 of the complete algorithm rectify this problem. Furthermore, notice that the jump point $v_n^1 = b^1$ is indeed a point where all bidders are indifferent between actions b^0 and b^1 as both actions provide these types with an expected utility of 0.

In the next subsection we prove that the BID algorithm constructs a finite-action PSNE under the auction rules that we discussed previously in Section 2.6.

2.6.3 Proof that the BID Algorithm Constructs a Finite-Action PSNE

Given the type distributions \mathbf{F} and the reserve price r , the BID algorithm produces a finite-action set \mathcal{B}^M , a vector of jump points $\mathbf{v}_n \in \hat{\Sigma}_n^M$ for each bidder n , and a bidding function β_n^M for each bidder n that corresponds to his vector of jump points \mathbf{v}_n as first defined in (2.13). In this section we prove the following proposition:

Proposition 1. *Given the type distributions \mathbf{F} and the auction rules laid out in Section 2.6, the finite-action set \mathcal{B}^M and the bidding functions β^M constructed by the BID algorithm characterize a finite-action PSNE.*

Proof. To prove this proposition we have to show that any bidder n 's best response is to use the strategy β_n^M , given that:

- Everyone else's type distributions are \mathbf{F}_{-n} .
- The set of available actions is \mathcal{B}^M .
- Everyone else is following the strategies prescribed by β_{-n}^M .

For expository clarity, we only prove this result for bidder 1 as the proof for any other bidder is symmetric.

We would like to show that bidder 1's best response is given by the following strategy:

$$\beta_1^M(v_1) = b^0 \mathbf{1}_{[v_1^0, v_1^1]}(v_1) + \sum_{m=2}^{M+1} b^{m-1} \mathbf{1}_{(v_1^{m-1}, v_1^m]}(v_1),$$

where $v_1 \in \mathcal{T}_1$.

First, we show that bidder 1 will bid b^{m-1} if $v_1 \in (v_1^{m-1}, v_1^m)$ for all integers $1 \leq m \leq M+1$. Recall that, by the BID algorithm's construction, for integers $2 \leq m \leq M$ the set of jump points $v_1^m \in \{\mathbf{v}_1\}$ give the exact points where bidder 1 is indifferent between bidding b^{m-1} and b^m . In particular, these points satisfy the following equation:

$$(v_1^m - b^{m-1}) \cdot \left[\prod_{n \neq 1} F_n(v_n^m) \right] = (v_1^m - b^m) \cdot \left[\prod_{n \neq 1} F_n(v_n^{m+1}) \right]. \quad (2.18)$$

Next, observe that for all $\epsilon > 0$:

$$\epsilon \cdot \left[\prod_{n \neq 1} F_n(v_n^m) \right] < \epsilon \cdot \left[\prod_{n \neq 1} F_n(v_n^{m+1}) \right]. \quad (2.19)$$

By adding (2.18) to (2.19), we see that:

$$(v_1^m + \epsilon - b^{m-1}) \cdot \left[\prod_{n \neq 1} F_n(v_n^m) \right] < (v_1^m + \epsilon - b^m) \cdot \left[\prod_{n \neq 1} F_n(v_n^{m+1}) \right], \quad (2.20)$$

which implies that all types $v_1 > v_1^m$ strictly prefer bidding b^m to b^{m-1}

Similarly, it is easy to show that for all $\epsilon > 0$:

$$(v_1^m - \epsilon - b^{m-1}) \cdot \left[\prod_{n \neq 1} F_n(v_n^m) \right] > (v_1^m - \epsilon - b^m) \cdot \left[\prod_{n \neq 1} F_n(v_n^{m+1}) \right], \quad (2.21)$$

which implies that all types $v_1 < v_1^m$ strictly prefer bidding b^{m-1} to b^m .

Therefore, for all integers $2 \leq m \leq M$, we can represent the relationship between bidder 1's types and his preferences as follows:

$$\begin{aligned} v_1 > v_1^m &\Rightarrow b^{m-1} < b^m, \\ v_1 < v_1^m &\Rightarrow b^{m-1} > b^m, \\ v_1 = v_1^m &\Rightarrow b^{m-1} \sim b^m. \end{aligned} \quad (2.22)$$

So for all integers $2 \leq m \leq M$, the set of jump points divide bidder 1's support into regions that allow us to infer his preferences.

Next consider $m = 1$ and suppose that $v_1 < v_1^1$. But recall that the BID algorithm sets $v_1^1 = b^1$. Therefore, the best response for types $v_1 < v_1^1$ is to select action b^0 and not participate in the auction as any other choice provides a negative expected utility. Similar reasoning shows that types $v_1 > v_1^1$ will strictly prefer b^1 to b^0 because b^1 provides a positive expected utility while b^0 gives a utility of 0. Finally, the type $v_1 = v_1^1$ is exactly indifferent between bidding b^0 and b^1 as he receives an expected utility of 0 from both. Therefore, the relationship in (2.22) also holds for $m = 1$.

Since (2.22) holds for all integers $1 \leq m \leq M$, we can now use the transitive property to determine his optimal bid for any $v_1 \in \mathcal{T}_1 \setminus \{\mathbf{v}_n\}$. Specifically, suppose that $v_1 \in (v_1^{m-1}, v_1^m)$ for some integer $1 \leq m \leq M + 1$. Then the following must be true:

$$\begin{aligned} v_1 > v_1^{m-1} > v_1^{m-2} > \dots > v_1^0 &\Rightarrow b^{m-1} > b^{m-2} > \dots > b^0, \\ v_1 < v_1^m < v_1^{m+1} < \dots < v_1^M &\Rightarrow b^{m-1} > b^m > \dots > b^M, \end{aligned} \quad (2.23)$$

and his best response is to choose action b^{m-1} . Therefore, for all integers $1 \leq m \leq M + 1$, we have shown that bidder 1 will bid b^{m-1} if $v_1 \in (v_1^{m-1}, v_1^m)$.

Next we have to prove that β_1^M gives the best response for bidder 1's types $v_1 \in \{v_1\}$. First consider $m = 0$. Then $v_1 = \underline{v}_1$ and he will choose action b^0 as any other action gives him a negative expected utility. Meanwhile, if $m = M + 1$, then $v_1 = \bar{v} > v_1^M$ and once again using (2.23) we see that his optimal action is to bid b^M . Finally in the case of integers $1 \leq m \leq M$, we can once again use the relationships given in (2.22) and the transitive property to show that:

$$\begin{aligned}
v_1 > v_1^{m-1} > v_1^{m-2} > \dots > v_1^0 &\Rightarrow & b^{m-1} > b^{m-2} > \dots > b^0, \\
v_1 < v_1^{m+1} < v_1^{m+2} < \dots < v_1^M &\Rightarrow & b^m > b^{m+1} > \dots > b^M, \\
v_1 = v_1^m &\Rightarrow & b^{m-1} \sim b^m,
\end{aligned} \tag{2.24}$$

and that in these situations he is indifferent between bidding b^{m-1} and b^m . Consequently, for his types $v_1 \in \{v_1\}$, bidder 1 has no incentive to deviate from the strategy prescribed by β_1^M .

Therefore, we have shown that β_1^M is indeed the best response strategy for bidder 1 and that the BID algorithm constructs a finite-action PSNE. \square

Although this proof is not formally a convergence result for the BID algorithm, it does show that the algorithm produces a finite-action PSNE when given a set of candidate jump points along one of the bidder's type space (i.e., the reference bidder's seed points). Consequently, as this result closely resembles the convergent numerical method suggested by Athey (2001), the next aspect to consider is the algorithm's performance as the step size $h \rightarrow 0$ (i.e., when given more and more seed points along the reference bidder's type space). This is investigated further in the next example.

Example 2

Consider the following example from Fibich and Gavious (2003). Suppose that we have two bidders who draw their types from the following distributions:

$$\begin{aligned} F_1(v_1) &= v_1 + .4v_1^2(1 - v_1^2), & v_1 &\in [0, 1], \\ F_2(v_2) &= v_2 - .4v_2^2(1 - v_2^2), & v_2 &\in [0, 1]. \end{aligned}$$

Fig. 2.3 depicts these distributions. Notice that they do not cross in the interior as F_2 stochastically dominates F_1 .

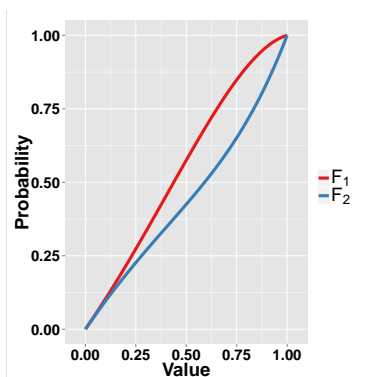


FIGURE 2.3: Type distributions for Example 2. These distributions do not cross in the interior as F_2 stochastically dominates F_1 .

Although this problem was first introduced by Fibich and Gavious, Kirkegaard (2009) noted that they “do not plot bidding strategies or comment on whether they intersect.” Kirkegaard also did not attempt to plot the strategies, but he did go on to show that the equilibrium functions must intersect at least once in the interior—despite the fact that the type distributions exhibit first order stochastic dominance. In fact, Kirkegaard proved that first order stochastic dominance is a necessary, but not sufficient condition for the PSNE solutions not to cross. We applied the BID algorithm to this example with varying step sizes h . Results appear in Fig. 2.4.

So we see that the solutions appear to converge to a continuous solution as the step size h decreases. Furthermore, the algorithm finds that the PSNE solutions

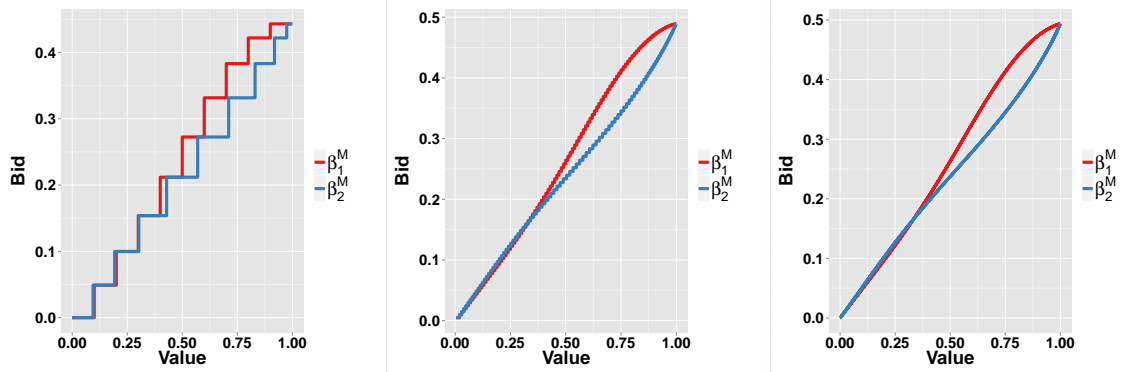


FIGURE 2.4: Finite-action PSNE constructed by the BID algorithm for Example 2 using various step sizes h to investigate its convergence properties. Notice that, despite the fact that the type distributions did not cross, the PSNE bidding functions appear to converge to continuous functions that cross once in the interior—a result which is consistent with Kirkegaard (2009). An accuracy of $\gamma = 10^{-8}$ was used in all three cases. Left: $h = .1$. Center: $h = .01$. Right: $h = .001$.

cross once in the interior—a result that is consistent with Kirkegaard’s findings.

Of course, without a known closed-form solution to Example 2, it is difficult to be completely sure that this is in fact the continuous PSNE solution. Unfortunately, as there is still no convergent algorithm, this also happens to be the main weakness of all numerical methods to date. There have been recent developments, however, which allow us to test whether an algorithm’s solutions are consistent with economic theory. We discuss this test and the theory behind it in the next section.

2.7 Using Economic Theory to Evaluate Numerical Solutions

Because no existing numerical method has yet been able to prove convergence, it has been extremely difficult to evaluate any algorithm’s performance—particularly in situations where there is no closed-form solution. Recent theoretical developments derived in Kirkegaard (2009), however, provide a way forward that sidesteps the need to examine the PSNE bidding functions directly. Hubbard et al. (2013) then showed how these results can be used to evaluate the performance of any numerical method

through a necessary visual “test” that must be passed.

Kirkegaard (2009) noted that, prior to his work, “all existing classes of numerical examples of asymmetry in first price auctions share a common feature...bidding strategies will be found to cross at most once,” despite the fact that it was “straightforward to construct examples in which bidding strategies cross several times.” Consequently, under the standard model assumptions discussed in Section 2.2, he proceeded to characterize the general behavior of these PSNE solutions that may cross.¹³

Specifically, he derived theoretical results that allow us to make qualitative predictions about how the PSNE bidding functions should behave by considering two ratios: the ratio of the type distributions and the ratio of PSNE expected utilities. For any two bidders i and j in the N bidder case, these ratios are defined as follows:

$$F_{i,j}(v) = \frac{F_j(v)}{F_i(v)}, \quad (2.25)$$

$$R_{i,j}(v) = \frac{EU_i(v, \beta_i(v))}{EU_j(v, \beta_j(v))}, \quad (2.26)$$

where $v \in (\underline{v}, \bar{v}]$. Notice that the first ratio is exogenously given (i.e., it is *a priori* known since it is determined by the given type distributions) while the second ratio is endogenously given (i.e., it is *a priori* unknown since it depends on the unknown PSNE bidding functions β_i and β_j).

Kirkegaard showed that these two ratios must exhibit the following qualitative behavior when the bidders are following their PSNE bidding functions:

$$R_{i,j}(v) \underset{<}{\overset{\geq}{\cong}} F_{i,j}(v) \iff \beta_i(v) \underset{<}{\overset{\geq}{\cong}} \beta_j(v) \quad (2.27)$$

$$R_{i,j}(v) \underset{<}{\overset{\geq}{\cong}} F_{i,j}(v) \iff R'_{i,j}(v) \underset{<}{\overset{\geq}{\cong}} 0. \quad (2.28)$$

Here equation (2.27) determines which bidder is more aggressive in certain regions.¹⁴

¹³ Observe that we are once again assuming a common support $\mathcal{T}_n = [\underline{v}, \bar{v}]$ for the type distributions F_n . An earlier version of Kirkegaard (2009) also has results that extend to the different support case when there are exactly two bidders.

¹⁴ Given the same type, the more aggressive bidder is the one who chooses to place the higher bid.

Notice that (2.27) also implies that, whenever the PSNE bidding functions cross at some type $v \in (\underline{v}, \bar{v}]$, the two ratios must cross at that same type as well. Meanwhile, equation (2.28) limits the behavior of the paths that $R_{i,j}$ can take: $R_{i,j}$ is increasing whenever it is above $F_{i,j}$, $R_{i,j}$ is decreasing whenever it is below $F_{i,j}$, and $R_{i,j}$ is stationary whenever it crosses $F_{i,j}$. Some “possible” paths that would be consistent with this (2.27) and (2.28) appear in Fig. 2.5.

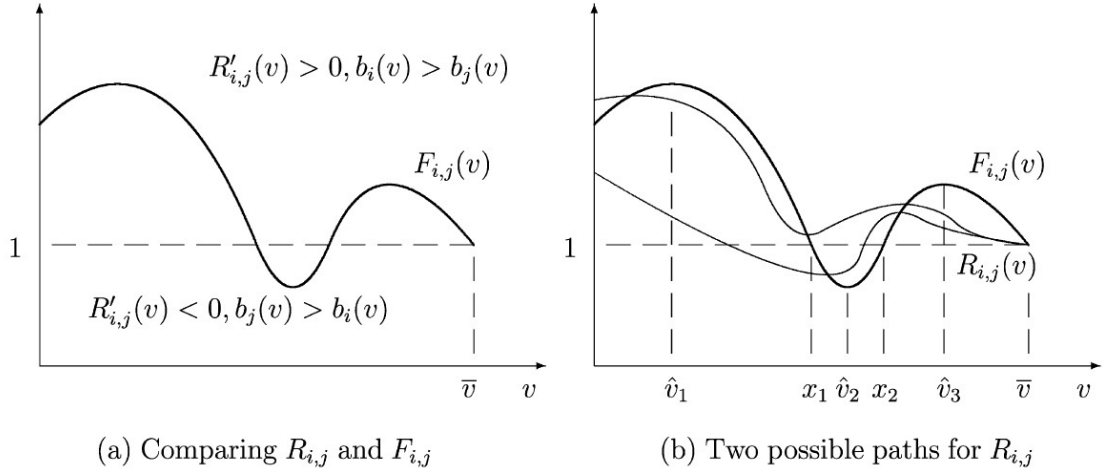


FIGURE 2.5: Stylistic depiction of two “possible” paths that $R_{i,j}$ could take that would still be consistent with economic theory (figure reproduced from Kirkegaard (2009)). Although qualitatively there are many such possible paths, quantitatively only one will satisfy the conditions of the PSNE solution since it is unique.

Kirkegaard also derived results that show how $F_{i,j}$ can be used to determine an upper bound for the number of times that the PSNE bidding functions can cross in $v \in (\underline{v}, \bar{v})$. Let $\hat{v}_1 < \hat{v}_2 < \dots < \hat{v}_K$ be the interior types where $F_{i,j}$ is locally maximized or minimized (i.e., “peaks”) and let $\hat{v}_0 = \underline{v}$ and $\hat{v}_{K+1} = \bar{v}$. Kirkegaard then showed that $R_{i,j}$ and $F_{i,j}$ can cross at most once in any of the intervals in $\{(\hat{v}_k, \hat{v}_{k+1}] : k = 0, \dots, K - 1\}$, and that cannot not cross in the interval $(\hat{v}_K, \hat{v}_{K+1}]$. Therefore, the number of times the PSNE bidding functions can cross is bounded above by the number of interior peaks K that $F_{i,j}$ has.

Furthermore, Kirkegaard proved that there are certain situations where the number of times the PSNE functions cross must be equal to the number of interior peaks K of $F_{i,j}$. These situations occur when $F_{i,j}$ exhibits the “diminishing wave” property. Intuitively, $F_{i,j}$ exhibits the diminishing wave property when the following three conditions are satisfied:

1. The interior peaks of $F_{i,j}$ alternate between being above and below 1;
2. As v increases:
 - (a) The interior peaks of $F_{i,j}$ that are above 1 get closer to 1;
 - (b) The interior peaks of $F_{i,j}$ that are below 1 get closer to 1;
3. $F_{i,j}$ is maximized or minimized as $v \rightarrow \underline{v}$.

Fig. 2.6 depicts several examples of $F_{i,j}$ that exhibit the diminishing wave property. For the formal definition of the diminishing wave property, see Kirkegaard (2009).

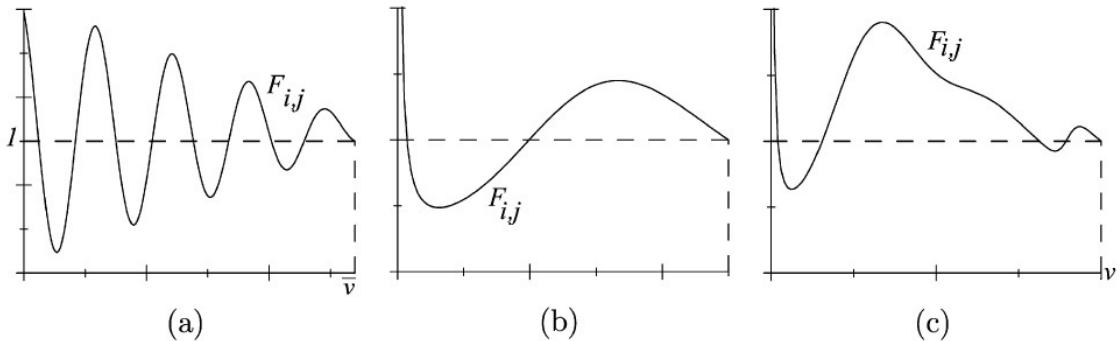


FIGURE 2.6: Three examples of $F_{i,j}$ which exhibit the diminishing wave property (figure reproduced from Kirkegaard (2009)). Left: A stylistic example. Center: $F_i(v_i) = \left(\frac{v_i}{5}\right)^2$, $v_i \in [0, 5]$ and $F_j(v_j)$ a normal distribution truncated on $[0, 5]$ with mean $\mu = 3$ and standard deviation $\sigma = 1$. Right: $F_i(v_i) = (v_i/10)^2$, $v_i \in [0, 10]$ while $F_j(v_j) = \frac{1}{3}(G_1(v_j) + G_2(v_j) + G_3(v_j))$ where G_i is a normal distribution truncated on $[0, 10]$ with mean $\mu_k = 3k$ and standard deviation $\sigma_1 = \sigma_2 = 1$ and $\sigma_3 = 0.25$.

Kirkegaard proved that if $F_{i,j}$ satisfies the diminishing wave property, then $R_{i,j}$ and $F_{i,j}$ must cross exactly once in all of the intervals in $\{(\hat{v}^k, \hat{v}^{k+1}] : k = 0, \dots, K-1\}$, and they must not cross in the interval $(\hat{v}_K, \hat{v}_{K+1}]$. Therefore, by (2.27), in this situation the number of times the PSNE bidding functions cross must be equal to the number of interior peaks K that $F_{i,j}$ has.

Consequently, even though the ratio of expected utility payoffs is endogenous, Hubbard et al. (2013) noted that Kirkegaard’s results are suggestive of a necessary visual “test” that can be used to help evaluate whether a numerical method’s proposed solution is consistent with economic theory. Specifically, they proposed estimating the true unknown ratio $R_{i,j}(v)$ with $\hat{R}_{i,j}(v)$, which is the ratio obtained by using a numerical method’s approximated PSNE bidding functions. Plotting $F_{i,j}$ and $\hat{R}_{i,j}$ then gives a visual test for the numerical method’s proposed solutions. In particular, according to Hubbard et al., one should pay specific attention to:

1. **Slope:** Whenever $F_{i,j}$ and $\hat{R}_{i,j}$ cross (i.e., whenever the proposed PSNE bidding functions cross), $\hat{R}_{i,j}$ should be flat. Meanwhile, whenever $\hat{R}_{i,j}$ is above $F_{i,j}$, $\hat{R}_{i,j}$ should be increasing; whenever $\hat{R}_{i,j}$ is below $F_{i,j}$, $\hat{R}_{i,j}$ should be decreasing.
2. **Location:** $F_{i,j}$ and $\hat{R}_{i,j}$ should cross at most once in any interval in $\{(\hat{v}_k, \hat{v}_{k+1}] : k = 0, \dots, K-1\}$ (and exactly once when the diminishing wave property is exhibited), and never in the interval $(\hat{v}_K, \hat{v}_{K+1}]$.

Regardless of the actual method used by a numerical algorithm, this test can be used to evaluate the validity of any solution that it produces. Therefore, any proposed algorithm should yield solutions that, at the very minimum, pass this necessary test.

With this test in mind, we evaluated the performance of the BID algorithm using various numerical examples. In all of the examples, a step size of $h = 10^{-4}$ and an accuracy of $\gamma = 10^{-8}$ were used.

Example 2 (Continued)

We applied the visual test to the PSNE solution that was obtained in Example 2. Fig. 2.7 presents a summary of our results, where the right panel shows that the BID algorithm’s solution passes the visual test.

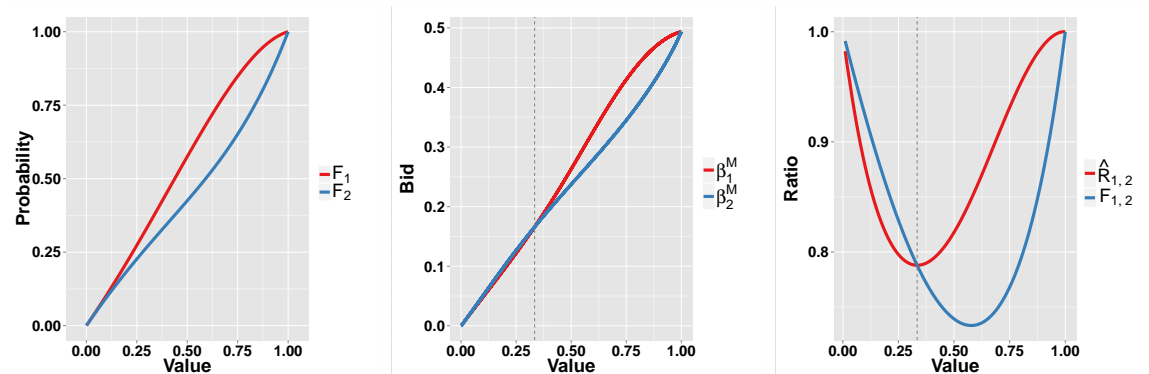


FIGURE 2.7: The BID algorithm applied to Example 2, where the type distributions exhibited first order stochastic dominance. Kirkegaard (2009) showed that this is a necessary, but not sufficient condition for the PSNE solutions not to cross. Left: The type distributions that do not cross. Center: The finite-action PSNE solution constructed by the BID algorithm that cross once (crossing indicated by the vertical dotted line). Right: The BID algorithm passed the necessary visual test proposed by Hubbard et al. (2013) (the vertical dotted line that appears here is the same as the one in the center panel).

Furthermore, observe that this example highlights the fact that the number of times the type distributions cross is not necessarily indicative of the number of times the PSNE bidding functions will cross—a result that was proved by Kirkegaard (2009). The next example, however, investigates a situation where we can determine the exact number of crossings ahead of time.

Example 3

Consider the following unsolved example that was first introduced by Kirkegaard (2009), which we have normalized to the unit interval:

$$F_1(v_1) = v_1^2, \quad v_1 \in [0, 1],$$

$$F_2(v_2) = \frac{1}{3} (G_1(v_2) + G_2(v_2) + G_3(v_2)), \quad v_2 \in [0, 1],$$

where $G_i(v_n)$ is the cdf of a normal distribution truncated on $[0, 1]$ with mean $\mu_i = \frac{3i}{10}$ and standard deviation $\sigma_1 = \sigma_2 = \frac{1}{10}$ and $\sigma_3 = \frac{1}{40}$. Fig. 2.8 summarizes our results.

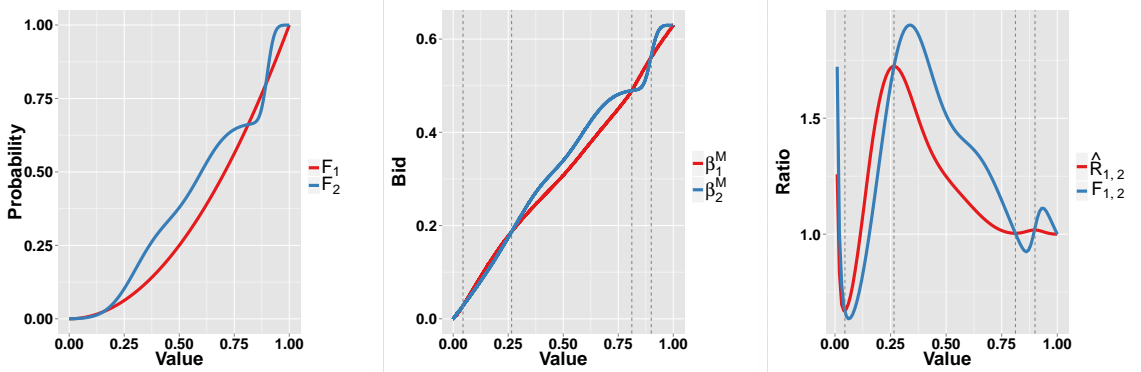


FIGURE 2.8: The BID algorithm applied to Example 3, where $F_{1,2}$ exhibits the diminishing wave property. Left: The type distributions. Center: The finite-action PSNE solution constructed by the BID algorithm (crossings indicated by the vertical dotted line). Right: The BID algorithm passed the necessary visual test proposed by Hubbard et al. (2013) (the vertical dotted lines that appears here are the same as the ones in the center panel).

The first thing to notice from the right panel in Fig. 2.8 is that $F_{1,2}$ exhibits the diminishing wave property with four peaks. Therefore, from Kirkegaard (2009), the PSNE bidding functions must cross exactly four times. The BID algorithm not only produced PSNE solutions that were consistent with this result (center panel), the solutions also passed the necessary visual test (right panel).

Example 4

The final example we consider in this section is a three bidder situation that was first introduced by Fibich and Gavish (2011). The type distributions are given by:

$$\begin{aligned}
 F_1(v_1) &= v_1, & v_1 &\in [0, 1], \\
 F_2(v_2) &= v_2 + 2v_2^2(1 - v_2^2)(0.25 - v_2^2)(0.75 - v_2^2), & v_2 &\in [0, 1], \\
 F_3(v_3) &= v_3 - 3v_3^2(1 - v_3^2)(0.25 - v_3^2)(0.75 - v_3^2), & v_3 &\in [0, 1].
 \end{aligned}$$

Fig. 2.9 portrays these distributions as well as the PSNE functions obtained from the BID algorithm.

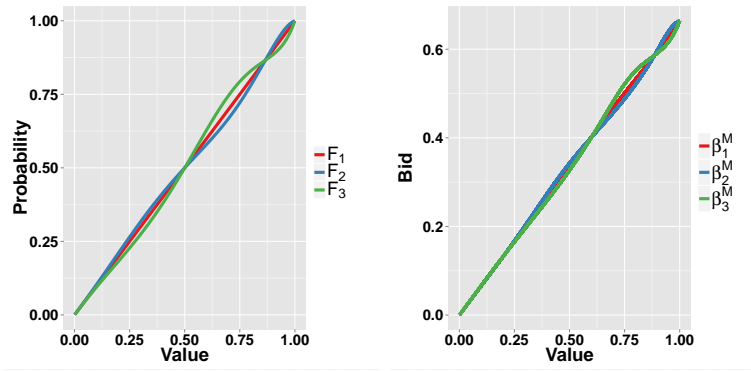


FIGURE 2.9: The BID algorithm applied to a three bidder situation. Left: The type distributions which cross twice in the interior (at $v_n = \sqrt{0.25}$ and $v_n = \sqrt{0.75}$). Right: The finite-action PSNE solution constructed by the BID algorithm.

Fibich and Gavish investigated this example with their boundary-value method, which we previously discussed in Section 2.3. In their analysis, they concluded that because the three type distributions cross each other exactly twice in the interior (at $v_n = \sqrt{0.25}$ and $v_n = \sqrt{0.75}$), by Kirkegaard (2009), the PSNE bidding functions must also cross exactly twice in the interior. However, several flaws exist in their application of Kirkegaard’s results.

First, as we have already discussed and as Kirkegaard first proved, the number of times the type distributions cross is not indicative of the number of crossings for

the PSNE bidding functions. Indeed, Example 2 can be used as a counter-example to their argument.

Second, the result from Kirkegaard (2009) that Fibich and Gavish (2011) cited does not actually base the number of crossings on the type distributions themselves to begin with. Instead, as we discussed in Section 2.7, the number of crossings for the PSNE functions is based on the number of interior peaks that the *ratio* of type distributions has. Furthermore, Kirkegaard’s results are meant to be applied to *pairwise comparisons* of the bidders—not all three bidders simultaneously as Fibich and Gavish have done.

Finally, the exact number of crossings is known only when the diminishing wave property is satisfied. From the bottom panels of Fig. 2.10, however, we see that none of the ratios exhibits this property since $\lim_{v \rightarrow 0} F_{i,j}(v) = 1$ for every pair of bidders, which violates the diminishing wave property’s third condition that $F_{i,j}$ is either maximized or minimized as $v \rightarrow 0$. Therefore, we only have an upper bound on the number of crossings—which, in this case, happens to be three for each pair of PSNE bidding functions (since each pairwise ratio $F_{i,j}$ has three interior peaks).

Incidentally, from Fig. 2.10 we see the finite-action PSNE functions produced by the BID algorithm for each pair of bidders do end up crossing exactly three times—a result that disagrees with Fibich and Gavish (2011). More importantly, every pair of the BID algorithm’s PSNE functions passed the visual test. This suggests that our results are consistent with economic theory.

In this section, we have presented several numerical examples suggesting that the BID algorithm produces finite-action PSNE that converge to the continuum-action PSNE solution. In the next section we briefly discuss other types auction asymmetries that have typically been ignored in the literature. We also offer some additional numerical evidence regarding the convergence of the BID algorithm to the

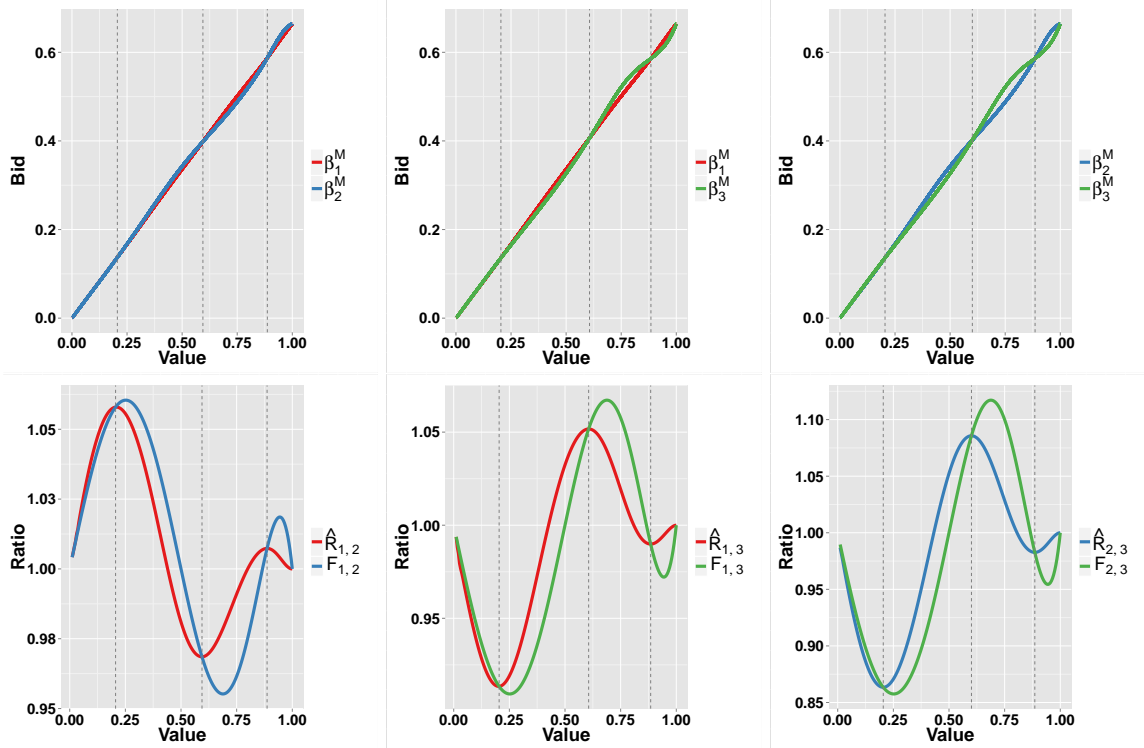


FIGURE 2.10: The BID algorithm applied to a three bidder situation. Top Row: The finite-action PSNE constructed by the algorithm for each pair of bidders (crossings indicated by the vertical dotted lines). Bottom Row: The BID algorithm passed the necessary test proposed by Hubbard et al. (2013) for each pair of bidders (the vertical dotted lines that appears in each panel are the same as the ones in the panel directly above it).

continuous PSNE solution.

2.8 Extensions of the Standard Model

In this section, we discuss some extensions of the standard auction model that was introduced in Section 2.2. Deriving the PSNE bidding functions for these extensions requires solving a system of differential equations that may present an entirely new set of challenges. Given the difficulty that numerical methods have already had in analyzing the standard model's poorly behaved differential system that was given in (2.4), it is perhaps not too surprising to learn that the numerical auction literature

has been slow to consider these extensions.

Recall, however, that Athey (2001) was able to establish more general PSNE existence results using her finite-action PSNE approach that completely circumvented the need to consider a system of differential equations. In particular, these existence results extended to situations where bidders have different supports for their type distributions or different utility functions. Because the BID algorithm is motivated in part by Athey's results, we evaluated its performance in these two situations by using numerical examples with known closed-form solutions.

2.8.1 Different Supports for the Type Distributions

Vickrey (1961) was the first to analyze auctions as noncooperative games with incomplete information. He considered the case of two bidders whose types are uniformly distributed on $[\underline{v}_1, \bar{v}_1]$ and $[\underline{v}_2, \bar{v}_2]$ where, without loss of generality, he assumed that $\underline{v}_1 \leq \underline{v}_2$. In his paper, Vickrey also presented the first PSNE solution by solving for the symmetric case. However, he was unable to solve for the more general asymmetric situation where the supports of the type distributions differed. A couple of years later, Griesmer et al. (1967) managed to find the PSNE solution for when the lower end point of the two supports was the same: $\underline{v}_1 = \underline{v}_2$.

It was not until Kaplan and Zamir (2012), however, that Vickrey's problem was solved. In fact, by showing how to reduce the differential system into a single differential equation, Kaplan and Zamir were able to provide an even more general solution that also accounts for the presence of a reserve price r . Disregarding the trivial situation where the reserve price is greater than the supports for the type distributions ($r > \bar{v}_2 \geq \bar{v}_1$), there are four possibilities to consider for the reserve price:

1. $r \leq \frac{v_1 + v_2}{2}$,
2. $r > \frac{v_1 + v_2}{2}$ and $r \neq v_2$,
3. $r > \frac{v_1 + v_2}{2}$ and $r = v_2$,
4. $\bar{v}_i \leq r \leq \bar{v}_j$.

The first case, $r \leq \frac{v_1 + v_2}{2}$, includes the no reserve price (i.e., $r = 0$) problem that Vickrey (1961) initially considered. Kaplan and Zamir (2012) showed that the inverse PSNE bidding function for bidder 1 in this situation is given by:

$$\phi_1(b_1) = \underline{v}_1 + \frac{(v_2 - v_1)^2}{(v_2 + v_1 - 2b_1)c_1 e^{\frac{v_2 - v_1}{v_2 + v_1 - 2b_1}} + 4(v_2 - b_1)}, \quad (2.29)$$

where

$$c_1 = \frac{\frac{(v_2 - v_1)^2}{\bar{v}_1 - v_1} + 4(\bar{b} - v_2)}{-2(\bar{b} - \underline{b})} e^{\frac{v_2 - v_1}{2(\bar{b} - \underline{b})}}, \quad (2.30)$$

and

$$\begin{aligned} \underline{b} &= \frac{v_1 + v_2}{2}, \\ \bar{b} &= \frac{\bar{v}_1 \bar{v}_2 - \left(\frac{v_1 + v_2}{2}\right)^2}{(\bar{v}_1 - v_1) + (\bar{v}_2 - v_2)}. \end{aligned} \quad (2.31)$$

Meanwhile, bidder 2's inverse PSNE function $\phi_2(b_2)$ can be obtained from $\phi_1(b_1)$ by simply interchanging the roles of \underline{v}_1 , \bar{v}_1 and \underline{v}_2 , \bar{v}_2 in equations (2.29) and (2.30). Furthermore, notice that the minimum bid in this situation is given by $\underline{b} = \frac{v_1 + v_2}{2}$. Therefore, if $v_1 < v_2$, then $\underline{b} > v_1 > r$ and there will be some types of bidder 1 who will choose not to participate in the auction even if there is no reserve price—something that did not occur in the standard model.

Next, Kaplan and Zamir showed that the solution to the second case, $r > \frac{\underline{v}_1 + \underline{v}_2}{2}$ and $r \neq \underline{v}_2$, for bidder 1 in terms of his inverse PSNE function is given by:

$$\phi_1(b_1) = \underline{v}_1 + \frac{(r - \underline{v}_1)(r - \underline{v}_2)}{r - \underline{v}_2 - c_3(b_1 - r)^\theta (b_1 + r - \underline{v}_1 - \underline{v}_2)^{(1-\theta)}}, \quad (2.32)$$

where

$$c_3 = \frac{(\bar{v}_2 - \underline{v}_2)}{(\bar{v}_1 - \underline{v}_1)} \frac{\left(\frac{\bar{v}_1 - r}{r - \underline{v}_1 + \bar{v}_2 - \underline{v}_2}\right)^{1-\theta}}{\left(\frac{\bar{v}_2 - r}{r - \underline{v}_2 + \bar{v}_1 - \underline{v}_1}\right)^\theta}, \quad (2.33)$$

$$\theta = \frac{r - \underline{v}_1}{(r - \underline{v}_1) + (r - \underline{v}_2)},$$

and

$$\underline{b} = r, \quad (2.34)$$

$$\bar{b} = \frac{\bar{v}_1 \bar{v}_2 - (\underline{v}_1 + \underline{v}_2)r + r^2}{(\bar{v}_1 - \underline{v}_1) + (\bar{v}_2 - \underline{v}_2)}.$$

Like before, bidder 2's inverse PSNE function $\phi_2(b_2)$ can be obtained from $\phi_1(b_1)$ by swapping the roles of $\underline{v}_1, \bar{v}_1$ and $\underline{v}_2, \bar{v}_2$ in (2.32) and (2.33).

Kaplan and Zamir then proved that the PSNE solution for the third situation, $r > \frac{\underline{v}_1 + \underline{v}_2}{2}$ and $r = \underline{v}_2 > \underline{v}_1$, for bidder 1 in terms of his inverse PSNE function is given by:

$$\phi_1(b_1) = \underline{v}_1 + \frac{\underline{v}_2 - \underline{v}_1}{1 - \left(\frac{r - \underline{v}_2}{\underline{v}_2 - \underline{v}_1}\right) \left[c_5 + \log\left(\frac{r - \underline{v}_1}{r - \underline{v}_2}\right) \right]}, \quad (2.35)$$

where

$$c_5 = \frac{(\bar{v}_1 - \underline{v}_2)(\underline{v}_2 - \underline{v}_1)}{(\bar{v}_1 - \underline{v}_1)(\bar{b} - \underline{v}_2)} - \log\left(\frac{\bar{v} - \underline{v}_1}{\bar{b} - \underline{v}_2}\right), \quad (2.36)$$

and

$$\underline{b} = r, \quad (2.37)$$

$$\bar{b} = \frac{\bar{v}_1 \bar{v}_2 - \underline{v}_1 \underline{v}_2}{(\bar{v}_1 - \underline{v}_1) + (\bar{v}_2 - \underline{v}_2)}.$$

Once again, bidder 2's inverse PSNE function $\phi_2(b_2)$ is obtained from $\phi_1(b_1)$ by switching the roles of $\underline{v}_1, \bar{v}_1$ and $\underline{v}_2, \bar{v}_2$ in (2.35) and (2.36).

Finally, in the PSNE for the fourth case, $\bar{v}_i \leq r \leq \bar{v}_j$, only types $v_j > r$ participate in the auction by bidding r .

We evaluated the performance of the BID algorithm by investigating whether its solutions agreed with the known closed-form solutions of Kaplan and Zamir (2012). Note that the results of Section 2.7 do not apply here since we are no longer assuming a common support for the type distributions.¹⁵

Example 5

Consider the following example that was first presented by Kaplan and Zamir (2012):

$$V_1 \sim \text{Uniform}(0, 3),$$

$$V_2 \sim \text{Uniform}(3, 6),$$

$$r = 2.$$

The inverse PSNE bidding functions can be obtained from (2.32), (2.33), and (2.34), which gives:

$$\phi_1(b_1) = \frac{8(b_1 - 1)}{8 + b_1(b_1 - 4)},$$

$$\phi_2(b_2) = 3 + \frac{10(b_2 - 2)}{4 + 2b_2 - b_2^2}.$$

Inverting these functions gives us the following PSNE solution:

$$\beta_1(v_1) = \frac{2 \left(2 + v_1 - \sqrt{4 + 2v_1 - v_1^2} \right)}{v_1}, \quad v_1 \in [2, 3],$$

$$\beta_2(v_2) = \frac{v_2 - 8 + \sqrt{5} \sqrt{8 - 4v_2 + v_2^2}}{v_2 - 3}, \quad v_2 \in [3, 6].$$

¹⁵ Although an earlier version of Kirkegaard (2009) does contain some relevant results, we do not discuss them in this chapter.

Fig. 2.11 shows the finite-action PSNE produced by the BID algorithm. Notice that they coincide with the known closed-form analytic solutions.

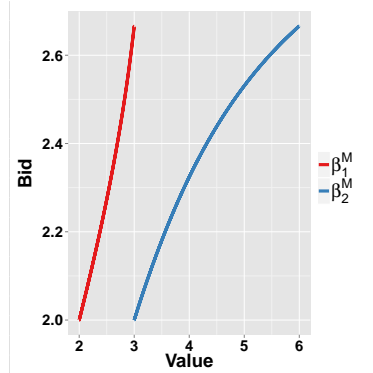


FIGURE 2.11: The BID algorithm applied to Example 5, where the type distributions have different supports. These results agree with the known closed-form analytic solution derived in Kaplan and Zamir (2012).

Closed-form solutions have also been derived for other asymmetric support situations. For example, Plum (1992) characterized the PSNE solution when types are drawn from the power distributions $F_1(v_1) = v_1^x$ and $F_2(v_2) = \left(\frac{v_2}{y}\right)^x$ when they have the same lower bound on the support. Meanwhile Cheng (2006) derived the analytic solution for $F_1(v_1) = v_1^x$ and $F_2(v_2) = \left(\frac{v_2}{y}\right)^z$ and where the distributions have the same lower bound and the upper bound is given by $y = \frac{z(x+1)}{x(z+1)}$. Although not depicted here, the BID algorithm also produced results that agreed with these known closed-form solutions.

2.8.2 Different Utility Functions

Another assumption imposed in the standard model is that all bidders are risk neutral. In reality, however, people will have different degrees of risk aversion. This introduces a new type of asymmetry to the standard auction game.

Let $U_n(v_n, \mathbf{b})$ denote bidder n 's utility function where:

$$U(v_n, \mathbf{b}) = \begin{cases} g_n(v_n, b_n) & \text{if } b_n > b_m \text{ for all } n \neq m \\ 0 & \text{otherwise.} \end{cases}$$

Here $g_n(v_n, b_n)$ is a utility function that is specific to bidder n , giving the payoff that he receives from winning the auction with a bid of b_n . Meanwhile, we still assume that he receives a utility of 0 if he loses the auction. Finally, we assume that the utility function of each bidder is common knowledge.

Each bidder is then looking to submit the bid that maximizes his expected utility, which is now given by:

$$\max_{b_n} EU_n(v_n, b_n) = \max_{b_n} \left\{ g(v_n, b_n) \prod_{m \neq n} F_m(\phi_m(b_n)) \right\}. \quad (2.38)$$

Like the standard model, this leads to system of poorly behaved differential equations that generally do not have a closed-form solution. See Hubbard and Paarsch (2014) for a more rigorous treatment of this problem.

Under these new assumptions on the utility functions, however, the results derived in Athey (2001) still apply. Therefore, a PSNE exists in every finite-action game, and there exists a sequence of finite-action PSNE that converges to the continuum-action PSNE solution. Consequently, the analysis of a difficult differential system can be avoided by finding jump points instead.

Before the BID algorithm can be applied to this situation, however, step 2 of its core part requires some slight modifications. In particular, rather than (2.16), we

will now be trying to solve the following nonlinear system:

$$\left\{ \begin{array}{l} g_1(v_1^m, b^{m-1}) \cdot \left[\prod_{n \neq 1} F_n(v_n^m) \right] = g_1(v_1^m, b^m) \cdot \left[\prod_{n \neq 1} F_n(v_n^{m+1}) \right] \\ g_2(v_2^m, b^{m-1}) \cdot \left[\prod_{n \neq 2} F_n(v_n^m) \right] = g_2(v_2^m, b^m) \cdot \left[\prod_{n \neq 2} F_n(v_n^{m+1}) \right] \\ \vdots \\ g_N(v_N^m, b^{m-1}) \cdot \left[\prod_{n \neq N} F_n(v_n^m) \right] = g_N(v_N^m, b^m) \cdot \left[\prod_{n \neq N} F_n(v_n^{m+1}) \right], \end{array} \right. \quad (2.39)$$

Meanwhile, the rest of the BID algorithm proceeds as normal.

Unfortunately, it is now arguably even more difficult to evaluate the performance of the BID algorithm under this circumstance: not only is there still no general closed-form solution, the results of Kirkegaard (2009) now no longer apply. In fact, to our knowledge, the only known closed form solution to this situation is the case of symmetric risk-averse bidders that was provided by Krishna (2002).

Specifically, Krishna assumed two symmetric bidders with the same type distribution $F(v)$. In addition, these bidders also shared the same constant relative risk aversion (CRRA) utility of the form $g_n(v_n, b_n) = g(v_n, b_n) = (v_n - b_n)^\alpha$, where $\alpha \in (0, 1)$ indexes the degree of risk aversion: as $\alpha \rightarrow 0$, the bidder becomes more averse. Krishna then showed that the PSNE solution for this situation coincides with the PSNE solution for two symmetric risk-neutral bidders drawing their types from $[F(v)]^{\frac{1}{\alpha}}$, where the relevant closed-form PSNE solution was given in (2.3).

Example 6

Suppose that we have two symmetric bidders who share the following type distribution and utility function:

$$F(v) = \sqrt{v},$$
$$g(v, b) = (v - b)^{\frac{1}{2}},$$

where $v \in [0, 1]$. Observe that $\alpha = \frac{1}{2}$ in this example.

By the results derived in Krishna (2002), we know that the PSNE should coincide with the PSNE for two symmetric risk-neutral bidders drawing their types from $[F(v)]^{\frac{1}{\alpha}} = [\sqrt{v}]^{\frac{1}{1/2}} = v$, where $v \in [0, 1]$. But we have already found the corresponding PSNE in Example 1. Therefore, the PSNE solution for this situation is given by $\beta(v) = \frac{v}{2}$.

Fig. 2.12 shows the results of the BID algorithm, which produced results that agreed with this closed-form solution.

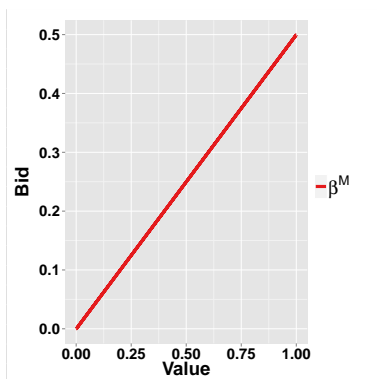


FIGURE 2.12: The BID algorithm applied to a symmetric risk-averse bidders situation. Results agreed with the known closed-form analytic solution derived in Krishna (2002).

Although there is no known general closed-form solution to the risk-averse bidders model that we are considering in this section, Maskin and Riley (1984) derived some theoretical results that allow us to make some predictions about the behavior of

the bidders. In particular, they showed that higher levels of risk aversion leads to more aggressive bidding behavior. The next example investigated whether the BID algorithm produced solutions that were consistent with this result.

Example 7

Suppose that we have two bidders who both draw their types from the Uniform(0, 1) distribution, but have different degrees of risk aversion:

$$g(v_1, b) = (v_1 - b)^{\frac{3}{4}},$$

$$g(v_2, b) = (v_2 - b)^{\frac{1}{2}}.$$

Notice that bidder 2 is more risk averse than bidder 1. Therefore, by Maskin and Riley (1984), we know that bidder 2 will be more aggressive in his bids (i.e., given the same valuation, bidder 2 will choose to place a higher bid).

Fig. 2.13 shows the results of the BID algorithm, which agree with Maskin and Riley’s result.

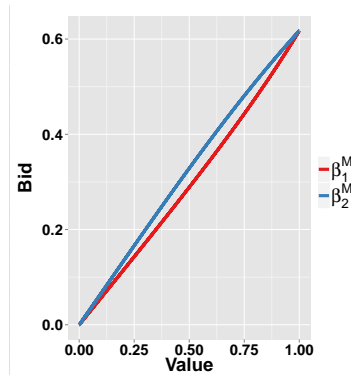


FIGURE 2.13: The BID algorithm applied to a situation where bidders are symmetric in their type distributions, but asymmetric in their degrees of risk aversion. Results, which show the more risk averse bidder (bidder 2) bidding more aggressively, are consistent with Maskin and Riley (1984).

Finally, we note that because this subsection describes a nonstandard assumption in the literature, there is still no general consensus on how to model this problem.

Our approach mirrors that of Hubbard and Paarsch (2014). Cox et al. (1982), on the other hand, considered a different situation where each bidder knows his own utility function but treats the utility function of others as random variables. Meanwhile, others that have also investigated bidders with different degrees of risk-aversion under different model assumptions include Maskin and Riley (1984) and Guerre et al. (2009).

2.9 Applications

The rapid growth of Internet advertising has led to the creation of a unique ad exchange auction environment that must reconcile the interests of the many different stakeholders in the industry. This has led to the introduction of new concepts to the auction literature, such as the generalized second price auction that is employed by search engines to sell sponsored “pay-per-click” advertisements that appear alongside the organic results (Edelman et al., 2007).¹⁶

In this section, however, we focus on the real-time bidding (RTB) auctions that have recently emerged from the incredible increase in the amount of “pay-per-impression” ad opportunities being offered.¹⁷ Specifically, every time a user accesses a website with an eligible pay-per-impression ad slot, a request is sent to an ad exchange that then holds an RTB auction that transpires over milliseconds. This auction begins with the ad exchange sending advertisers data associated with the user (e.g., his demographic information, his location, the webpage being loaded, etc.), which advertisers then use to decide how much to bid on the ad slot. The highest bidding advertiser then wins the right to display their ad, which is loaded onto the webpage before being displayed to the user.

¹⁶ Organic search results are the results that naturally appear because of their relevance to the keywords being searched.

¹⁷ In a pay-per-impression scheme, the advertiser pays for each showing (i.e., impression) of the advertisement, regardless of whether or not it elicited a response (e.g., a click or a purchase).

Despite the impressive technology behind RTB auctions, however, they still operate via the familiar second-price sealed-bid auction format where the highest bidder still wins, but only pays the amount that the second highest bidder bid. Auction theory provides several justifications for this format choice. Indeed, the most well-known property of the second-price sealed-bid auction is that it induces truthful bidding among the participants (i.e., the PSNE for all bidders is to bid their type: $\beta_n(v_n) = v_n$ for all n). Consequently, this auction format also results in efficient allocations of the good (i.e., the winner is the bidder with the highest valuation). Furthermore, under some fairly general assumptions, the Revenue Equivalence Theorem—one of auction theory’s most remarkable and important concepts—guarantees that the auctioneer’s expected revenue under the second-price sealed-bid auction format will be the same as it would be under many other auction formats.

Unfortunately, one of the assumptions of the Revenue Equivalence Theorem is that bidders are symmetric and risk neutral, which are tenuous assumptions at best—particularly in the online advertising world given the number of advertisers participating in the industry. Consequently, additional revenue can be potentially generated in an RTB auction by modifying the format. As we have already seen, however, auction theory is currently hard pressed to provide concrete answers as to how this revenue can be optimized when asymmetries are present. Therefore, numerical approximations are required (Marshall et al., 1994; Fibich and Gavious, 2003; Li and Riley, 2007).

Motivated by this, we applied the BID algorithm in order to investigate how an auctioneer’s expected revenue is affected by switching to a first-price auction or by manipulating the reserve price r .¹⁸ In addition, we also considered how the variance of the revenue is affected by these changes. This was accomplished via the following

¹⁸ We assume the following in the presence of a reserve price. In a first-price auction, the winning bidder (if there is one) pays his bid. Meanwhile, in a second-auction, the winning bidder (if there is one) pays the maximum of the the second highest bidder’s bid and the reserve price r .

Monte Carlo simulation:

Monte Carlo Simulation to Approximate Expected Revenues

1. Given: the reserve price r and the type distributions \mathbf{F} .
2. For $t = 1, \dots, T$:
 - (a) Draw a sample from each type distribution: $v_n^{(t)} \sim F_n$ for all n .
 - (b) Simulate a first-price auction equilibrium:
 - (i) Use BID algorithm to obtain finite-action PSNE functions β^M .
 - (ii) First-price PSNE bids are given by $b_n^{(t)} = \beta_n^M(v_n^{(t)})$ for all n .
 - (iii) Determine the first-price revenue $R_1^{(t)}$.
 - (c) Simulate a second-price auction equilibrium:
 - (i) Second-price PSNE bids are given by: $b_n^{(t)} = v_n^{(t)}$ for all n .
 - (ii) Determine the second-price revenue $R_2^{(t)}$.
3. Calculate Monte Carlo approximations:
 - (a) Expected Revenues:

$$E(R_1) = \frac{1}{T} \sum_{t=1}^T R_1^{(t)},$$

$$E(R_2) = \frac{1}{T} \sum_{t=1}^T R_2^{(t)}.$$

- (b) Variance of the Revenues:

$$Var(R_1) = \frac{1}{T-1} \sum_{t=1}^T \left(R_1^{(t)} - E(R_1) \right)^2,$$

$$Var(R_2) = \frac{1}{T-1} \sum_{t=1}^T \left(R_2^{(t)} - E(R_2) \right)^2.$$

Example 8

Consider the following example introduced by Marshall et al. (1994), who also examined expected revenues:

$$F_1(v_1) = v_1, \quad v_1 \in [0, 1],$$
$$F_2(v_2) = v_2^4, \quad v_2 \in [0, 1].$$

In our Monte Carlo simulations, we used $r = 0, .01, .02, \dots, .99$ and $T = 1,000,000$. Results appear in Fig. 2.14.

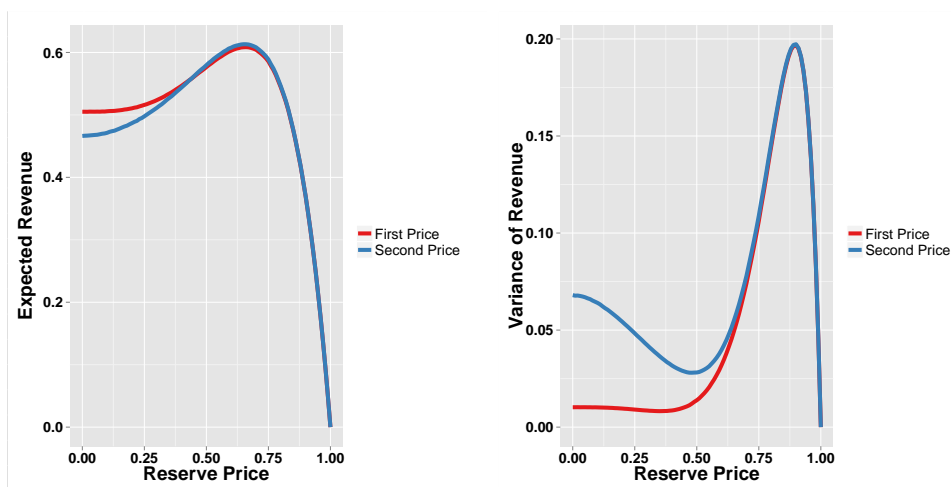


FIGURE 2.14: Monte Carlo simulations for Example 8. Left: When there is no reserve price, the first-price auction generates more expected revenue than the second price auction; under the optimal reserve price of $r = .66$, however, the second-price auction actually generated more revenue. Right: The variances of the revenues. The second-price auction always had a higher variance.

When there is no reserve price, the expected revenues were:

$$E(R_1) = 0.5050,$$

$$E(R_2) = 0.4664.$$

Therefore, the first-price auction generates more expected revenue than the second-price auction in this situation—a result that is consistent with Marshall et al. (1994).

Meanwhile, the optimal reserve price for both auction formats occurred when $r = .66$, in which case the expected revenues were:

$$E(R_1) = 0.6087,$$

$$E(R_2) = 0.6132.$$

So in this situation, the second-price auction actually generates slightly more expected revenue than the first price auction—a result that agrees with Fibich and Gaviious (2003).

Furthermore, the right panel in Fig 2.14 shows that the second-price auction always has a higher variance in the revenue generated than the first-price auction, with this variance being substantially higher when no reserve price is imposed. And although it was not investigated here, Monte Carlo simulations can also be used to approximate other auction properties such as efficiency or bidder surplus.

Consequently, this example highlights the fact that auctioneers may stand to benefit from modifying their current format. And although numerical methods like the BID algorithm can help auctioneers to make this decision, they also arguably serve an even greater purpose by filling in some of the gaps that currently exist in auction theory.

2.10 Concluding Remarks

In this chapter, we proposed the BID algorithm that numerically approximated the continuous PSNE solution through a sequence of finite-action PSNE that were constructed by finding where bidders were indifferent between two different actions. We then used numerical examples to evaluate the performance of the BID algorithm—including examples that are not typically considered by other numerical methods—and results showed that the BID algorithm produced solutions that were consistent with economic theory. Afterwards, we applied the BID algorithm to show how auc-

tions could be modified in order to generate more revenue for the auctioneer.

Consequently, although we have not formally established the convergence result that has been so elusive in the numerical auction literature, we have presented a compelling alternative to the prevailing approach in numerical methods.

Cross-Domain Recommender Systems

3.1 Introduction

Although the rapid growth of the Internet has made an enormous amount of information and choices available to users, it has also frequently overwhelmed them and caused them to make poor decisions (e.g., in buying or comparing products). At the same time, this information overload has also made it difficult for advertisers to reach users who are likely to be interested in their products. In order to address these challenges, recommender systems have been developed to help filter the information and display only the content that a user is likely to enjoy.

Although recommender systems have been successfully implemented in many different settings, research in this area has largely focused on models that make recommendations within a single domain (e.g., movies in the case of Netflix or music in the case of Pandora Radio). Oftentimes, however, companies such as Amazon.com would like to make recommendations that span multiple domains. In these situations, recommendations in one domain can potentially be improved by leveraging knowledge from other domains—a concept which has motivated the recent development of cross-

domain recommender systems.

In this chapter, we propose a cross-domain recommender system that is a multiple-domain extension of the Bayesian Probabilistic Matrix Factorization model that was first proposed by Salakhutdinov and Mnih (2008b). We begin by briefly reviewing the literature on recommender systems in Section 3.2. This overview focuses primarily on the models that are most closely related to the cross-domain model that we propose in Section 3.3. Afterwards, in Section 3.4, we provide experimental results showing that our proposed model outperforms other related models in two different cross-domain recommendation situations. Finally, Section 3.5 concludes.

3.2 Related Work

This section primarily focuses on the recommender systems that are most closely related to the cross-domain model that we propose in the next section. For a more comprehensive overview of the recommender system literature, see Ricci et al. (2011).

Suppose that we have N users and M items, where for the time being we treat the items as belonging to a single domain. Furthermore, assume that the users rate items on some fixed scale that depends on the constraints set by the recommender system—such as integers between 1 and S , where S is frequently equal to 5, 10, or 100. The models we discuss, however, are sufficiently general enough to handle all of the most popular ratings scales.

Let $\mathbf{R} \in \mathbb{R}^{N \times M}$ denote the user-item rating matrix. Here entry R_{ij} is the rating that user i gave to item j , where this entry is blank if the actual rating is “unknown” (i.e., if the user has not rated the item). The goal of the recommender system is to predict these unknown ratings, and subsequently recommend the items that it believes its users will rate highly.

However, in general \mathbf{R} will be extremely sparse since users will typically have rated only a very small fraction of the items that are being offered. For example,

the training dataset in the popular Netflix Prize competition contained 100,480,507 observed ratings that 480,189 users gave to 17,770 movies. This results in a user-item rating matrix that is over 98% empty.

Two main approaches have been used in designing recommender systems: content-based filtering and collaborative filtering. In general, content-based filtering algorithms rely on a description of each item and a profile for each user characterizing his preferences. Specifically, these models attempt to create a content-based profile for each user that is based on a weighted average of item features that the user has previously liked, and subsequently recommends the items that best match the user's profile. Statistical techniques used in this approach have included neural networks (Jennings and Higuchi, 1992), decision trees (Kim et al., 2001), and Bayesian classifiers (Pazzani et al., 1996). Meanwhile, an example of a content-based filtering model that has found success in recent years is the music recommender system of Pandora Radio.¹ In their "Music Genome Project," each song in the system is analyzed using up to 450 distinct musical characteristics ("genes"), and user profiles are created by using feedback on songs previously heard.²

The collaborative filtering approach, on the other hand, attempts to design a recommender system without having to rely on an actual explicit description of the items. Although this approach may consider item-to-item similarity, it is also motivated by the idea that users will enjoy the items that other like-minded users have enjoyed. Consequently, more emphasis is placed on using the rating matrix to identify these groups of users with similar preferences. Perhaps the most successful implementation of a collaborative filtering approach is Amazon.com's recommender system ("Customers who bought this item also bought"). Meanwhile, tools that have been used in this approach have included Bayesian networks (Breese et al.,

¹ <http://www.pandora.com/>

² <http://www.pandora.com/about/mgp>

1998), clustering methods (Ungar and Foster, 1998), and matrix factorization methods (Salakhutdinov and Mnih, 2008a).

In particular, factor-based matrix factorization models have found much recent success in the collaborative filtering literature. These models begin by positing that user preferences are determined by a small number of unobserved factors, which are modeled as vectors belonging to some low-dimensional latent space. Meanwhile, the items in the system are also conceptualized as vectors in this low-dimensional space based on how strongly they “match” these unobserved factors. A user’s rating of an item is then modeled by combining his latent feature vector with the latent feature vector of the item.

In a linear factor-based model, which is the primary focus for the remainder of this chapter, this combination is chosen to be the inner product of the two latent feature vectors. Specifically, the $N \times M$ rating matrix \mathbf{R} is modeled as the product of a $D \times N$ user feature matrix \mathbf{U} and a $D \times M$ item feature matrix \mathbf{V} , where D corresponds to the dimensionality of the latent space that is fixed beforehand (Rennie and Nati, 2005; Nati and Jaakkola, 2003). Training such a model then consists of finding the best rank- D approximation to the observed $N \times M$ rating matrix \mathbf{R} under some given loss function.

3.2.1 Probabilistic Matrix Factorization (PMF)

Perhaps the most popular and influential linear factor-based model is the Probabilistic Matrix Factorization (PMF) model that was recently introduced by Salakhutdinov and Mnih (2008a). Although other probabilistic factor-based models have been proposed prior to PMF (Hofmann, 1999; Marlin, 2004; Marlin and Zemel, 2004), they all suffered from either slow or inaccurate approximations to the posterior distribution over their hidden factors.

PMF begins by assuming the following likelihood over the observed ratings:

$$p(\mathbf{R}|\mathbf{U}, \mathbf{V}, \alpha) = \prod_{i=1}^N \prod_{j=1}^M [N(R_{ij}|\mathbf{U}_i^T \mathbf{V}_j, \alpha^{-1})]^{I_{ij}}, \quad (3.1)$$

where α is the precision parameter and I_{ij} is the indicator variable that is equal to 1 if user i rated item j and equal to 0 otherwise.

Next, zero-mean spherical Gaussian prior distributions are placed over the user feature matrix \mathbf{U} and the item feature matrix \mathbf{V} :

$$p(\mathbf{U}|\alpha_U) = \prod_{i=1}^N N(\mathbf{U}_i|\mathbf{0}, \alpha_U^{-1} \mathbf{I}),$$

$$p(\mathbf{V}|\alpha_V) = \prod_{j=1}^M N(\mathbf{V}_j|\mathbf{0}, \alpha_V^{-1} \mathbf{I}).$$

The model is then learned by maximizing the log-posterior over the user and item features with fixed hyperparameters α , α_U , and α_V :

$$\log p(\mathbf{U}, \mathbf{V}|\mathbf{R}, \alpha, \alpha_U, \alpha_V) = \log p(\mathbf{R}|\mathbf{U}, \mathbf{V}, \alpha) + \log p(\mathbf{U}|\alpha_U) + \log p(\mathbf{V}|\alpha_V) + C,$$

where \log denotes the natural log and C is some constant that does not depend on the model parameters \mathbf{U} and \mathbf{V} . Maximizing this log-posterior is equivalent to minimizing the sum of squared errors objective function with quadratic regularization terms:

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (R_{ij} - \mathbf{U}_i^T \mathbf{V}_j)^2 + \frac{\lambda_U}{2} \sum_{i=1}^N \|\mathbf{U}_i\|_F^2 + \frac{\lambda_V}{2} \sum_{j=1}^M \|\mathbf{V}_j\|_F^2, \quad (3.2)$$

where $\lambda_U = \alpha_U/\alpha$, $\lambda_V = \alpha_V/\alpha$, and $\|\cdot\|_F^2$ denotes the squared Frobenius norm. A local minimizer of equation (3.2) can then be found by performing gradient descent in \mathbf{U} and \mathbf{V} .

Salakhutdinov and Mnih (2008a) also proposed several extensions to their standard PMF model. The first extension was to pass the inner product between user

and item feature vectors through the logistic function $g(x) = 1/(1 + \exp(-x))$. In this situation, the observed data likelihood becomes:

$$p(\mathbf{R}|\mathbf{U}, \mathbf{V}, \alpha) = \prod_{i=1}^N \prod_{j=1}^M [N(R_{ij}|g(\mathbf{U}_i^T \mathbf{V}_j), \alpha^{-1})]^{I_{ij}}.$$

In addition, the integer ratings $1, \dots, S$ are mapped to the $[0, 1]$ unit interval using the function $t(x) = (x - 1)/(S - 1)$. These slight modifications to the standard PMF model help to improve performance by bounding the range of predictions and ensuring that they match the range of valid rating values. Like before, this logistic PMF model can be learned by performing gradient descent in \mathbf{U} and \mathbf{V} .

Salakhutdinov and Mnih also proposed the constrained PMF model, an extension that attempts to improve the predictions for infrequent users (i.e., users who have not rated many items). Specifically, let $\mathbf{W} \in \mathbb{R}^{D \times M}$ be a latent similarity constraint matrix whose entries measure the similarities between the items and the latent variables. In this constrained model, the latent feature vector for user i is instead defined as:

$$\mathbf{U}_i = \mathbf{Y}_i + \frac{\sum_{k=1}^M I_{ik} \mathbf{W}_k}{\sum_{k=1}^M I_{ik}}, \quad (3.3)$$

where I_{ik} is 1 if user i rated item k and 0 otherwise. Intuitively, the k^{th} column of \mathbf{W} can be interpreted as the effect of a user having rated item k has on the prior mean of the user's feature vector. Consequently, users that have rated similar items will tend to have similar prior distributions for their feature vectors. Meanwhile, \mathbf{Y}_i can be viewed as an offset that is added to the mean of the prior distribution to get the feature vector \mathbf{U}_i for user i .

The conditional likelihood over the observed ratings for this constrained model

is then:

$$p(\mathbf{R}|\mathbf{Y}, \mathbf{W}, \mathbf{V}, \alpha) = \prod_{i=1}^N \prod_{j=1}^M [N(R_{ij}|g(\mathbf{U}_i^T \mathbf{V}_j), \alpha^{-1})]^{I_{ij}},$$

where \mathbf{U}_i is given by (3.3) and g is once again the logistic function. Meanwhile, the latent similarity constraint matrix \mathbf{W} is regularized by placing a zero-mean spherical Gaussian prior on it:

$$p(\mathbf{W}|\alpha_W) = \prod_{k=1}^M N(\mathbf{W}_k|\mathbf{0}, \alpha_W^{-1}\mathbf{I}).$$

As with the other PMF models, this constrained model is trained by minimizing the sum of squared errors objective function with quadratic regularization terms:

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (R_{ij} - g(\mathbf{U}_i^T \mathbf{V}_j))^2 + \frac{\lambda_Y}{2} \sum_{i=1}^N \|\mathbf{Y}_i\|_F^2 - \frac{\lambda_V}{2} \sum_{j=1}^M \|\mathbf{V}_j\|_F^2 + \frac{\lambda_W}{2} \sum_{k=1}^M \|\mathbf{W}_k\|_F^2,$$

where $\lambda_Y = \alpha_Y/\alpha$, $\lambda_V = \alpha_V/\alpha$, and $\lambda_W = \alpha_W/\alpha$. Once again, a local minimizer of this objective function can be found by performing gradient descent in \mathbf{Y} , \mathbf{W} , and \mathbf{V} .

Salakhutdinov and Mnih then showed that all of their PMF models perform extremely well on the Netflix Prize competition dataset, handily outperforming Netflix's own CineMatch recommender system at the time of the competition. Furthermore, when the predictions of their PMF models were linearly combined, they achieved an error rate (in terms of RMSE) that was nearly 7% better than CineMatch. To help put this achievement into perspective, recall that Netflix was offering a 1 million US dollar prize to a team that was able to improve upon CineMatch's RMSE score by at least 10%, which was a feat that was only accomplished by combining the results from over 100 different models (Koren, 2009).

However, Salakhutdinov and Mnih pointed out that a major disadvantage of training their PMF models is the need for manual complexity control in order to make

the models generalize well to sparse and imbalanced datasets. Consequently, the process of finding appropriate and optimal values for the regularization parameters (i.e., λ_U and λ_V) is extremely computationally expensive, and typically involves training a multitude of models before choosing the model that performs the best. This eventually led the authors to propose a fully Bayesian treatment of their PMF model, which we discuss in the next subsection.

3.2.2 Bayesian Probabilistic Matrix Factorization (BPMF)

A major drawback of the PMF models was having to find and specify appropriate values for the regularization parameters. In order to address this issue, Salakhutdinov and Mnih (2008b) proposed Bayesian Probabilistic Matrix Factorization (BPMF), which is a fully Bayesian extension of their standard PMF model that provides automatic complexity control by using Markov Chain Monte Carlo (MCMC) methods to integrate out model parameters and hyperparameters. Furthermore, BPMF uses a predictive distribution for the ratings, which handles uncertainty more effectively than the *maximum a posteriori* (MAP) estimates of the PMF models.

BPMF begins by assuming the same conditional likelihood of the observed ratings as the standard PMF model that was given in (3.1):

$$p(\mathbf{R}|\mathbf{U}, \mathbf{V}, \alpha) = \prod_{i=1}^N \prod_{j=1}^M [N(R_{ij}|\mathbf{U}_i^T \mathbf{V}_j, \alpha^{-1})]^{I_{ij}}.$$

Unlike the PMF models, however, BPMF places the following prior distributions over the user and item latent feature vectors:

$$p(\mathbf{U}|\boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U) = \prod_{i=1}^N N(\mathbf{U}_i|\boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U^{-1}),$$

$$p(\mathbf{V}|\mathbf{m}_V, \boldsymbol{\Omega}_V) = \prod_{j=1}^M N(\mathbf{V}_j|\mathbf{m}_V, \boldsymbol{\Omega}_V^{-1}).$$

Afterwards, BPFM assumes the following Gaussian-Wishart priors for the hyperparameters:

$$p(\Theta_U|\Theta_0) = N(\boldsymbol{\mu}_U|\boldsymbol{\mu}_0, (\beta_0\boldsymbol{\Lambda}_U)^{-1})W(\boldsymbol{\Lambda}_U|\mathbf{W}_0, \nu_0),$$

$$p(\Phi_V|\Phi_0) = N(\mathbf{m}_V|\mathbf{m}_0, (b_0\boldsymbol{\Omega}_V)^{-1})W(\boldsymbol{\Omega}_V|\Psi_0, n_0),$$

where we have defined $\Theta_U = \{\boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U\}$, $\Theta_0 = \{\boldsymbol{\mu}_0, \nu_0, \mathbf{W}_0\}$, $\Phi_V = \{\mathbf{m}_V, \boldsymbol{\Omega}_V\}$, and $\Phi_0 = \{\mathbf{m}_0, n_0, \Psi_0\}$ for notational convenience. In addition, here $W(\mathbf{W}_0, \nu_0)$ denotes the Wishart distribution parametrized with $\nu > D - 1$ degrees of freedom and a $D \times D$ positive definite scale matrix \mathbf{W}_0 :

$$W(\boldsymbol{\Lambda}|\mathbf{W}_0, \nu_0) \propto |\boldsymbol{\Lambda}|^{\frac{\nu_0-D-1}{2}} \exp\left(-\frac{1}{2}\text{tr}(\mathbf{W}_0^{-1}\boldsymbol{\Lambda})\right).$$

The data generating process of BPFM can then be summarized as follows:

1. For each user i , draw a vector of latent features $\mathbf{U}_i \sim N(\mathbf{U}_i|\boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U^{-1})$.
2. For each item j , draw a vector of latent features $\mathbf{V}_j \sim N(\mathbf{V}_j|\mathbf{m}_V, \boldsymbol{\Omega}_V^{-1})$.
3. For each item j rated by user i , draw a rating $R_{ij} \sim N(R_{ij}|\mathbf{U}_i^T\mathbf{V}_j, \alpha^{-1})$.

The goal of BPFM is to invert this generative model and obtain the posterior predictive distribution for the rating R_{ij}^* that user i gives to item j . This distribution is obtained by integrating over the model parameters and hyperparameters:

$$p(R_{ij}^*|\mathbf{R}, \Theta_0, \Phi_0) = \int \int p(R_{ij}^*|\mathbf{U}_i, \mathbf{V}_j)p(\mathbf{U}, \mathbf{V}|\mathbf{R}, \Theta_U, \Phi_V)$$

$$p(\Theta_U, \Phi_V|\Theta_0, \Phi_0)d\{\mathbf{U}, \mathbf{V}\}d\{\Theta_U, \Phi_V\}.$$
(3.4)

However, this expression is analytically intractable. Therefore, Salakhutdinov and Mnih turned to MCMC-based methods in order to generate samples $\{\mathbf{U}_i^{(t)}, \mathbf{V}_j^{(t)}\}$

that are used to approximate the predictive distribution in equation (3.4). This is accomplished by using the following Monte Carlo approximation:

$$p(R_{ij}^* | \mathbf{R}, \Theta_0, \Phi_0) \approx \frac{1}{T} \sum_{t=1}^T p(R_{ij}^* | \mathbf{U}_i^{(t)}, \mathbf{V}_j^{(t)}),$$

which asymptotically yields exact results. Because the full conditional distributions for the BPMF model can be easily sampled from, Salakhutdinov and Mnih opted for a Gibbs sampling algorithm in their paper.

Specifically, the full conditional distribution over the item feature vector \mathbf{V}_j is given by:

$$\begin{aligned} p(\mathbf{V}_j | \mathbf{R}, \mathbf{U}, \Phi_V, \alpha) &\propto \prod_{i=1}^N [N(R_{ij} | \mathbf{U}_i^T \mathbf{V}_j, \alpha^{-1})]^{I_{ij}} N(\mathbf{V}_j | \mathbf{m}_V, \Omega_V^{-1}) \\ &\propto N(\mathbf{V}_j | \mathbf{m}_j^*, [\Omega_j^*]^{-1}), \end{aligned} \quad (3.5)$$

where

$$\begin{aligned} \Omega_j^* &= \Omega_V + \alpha \sum_{i=1}^N [\mathbf{U}_i \mathbf{U}_i^T]^{I_{ij}}, \\ \mathbf{m}_j^* &= [\Omega_j^*]^{-1} \left(\alpha \sum_{i=1}^N [\mathbf{U}_i R_{ij}]^{I_{ij}} + \Omega_V \mathbf{m}_V \right). \end{aligned}$$

Furthermore, notice that the latent item feature matrix V factorizes into the product of the full conditional distributions of all of the items in the system:

$$p(\mathbf{V} | \mathbf{R}, \mathbf{U}, \Phi_V) = \prod_{j=1}^M p(\mathbf{V}_j | \mathbf{R}, \mathbf{U}, \Phi_V, \alpha).$$

Consequently these distributions can be sampled in parallel, which greatly improves the computational speed of the MCMC.

Next, the full conditional distribution over the item hyperparameters also has a

closed form that is easy to sample:

$$\begin{aligned}
p(\Phi_V | \mathbf{V}, \Phi_0) &\propto p(\mathbf{V} | \mathbf{m}_V, \Omega_V) p(\Phi_V | \Phi_0) \\
&\propto N(\mathbf{m}_V | \mathbf{m}_0^*, (b_0^* \Omega_V)^{-1}) W(\Omega_V | \Psi_0^*, n_0^*),
\end{aligned} \tag{3.6}$$

where

$$\begin{aligned}
\mathbf{m}_0^* &= \frac{b_0 \mathbf{m}_0 + M \bar{\mathbf{V}}}{b_0 + M}, \quad b_0^* = b_0 + M, \quad n_0^* = n_0 + M, \\
[\Psi_0^*]^{-1} &= \Psi_0^{-1} + M \bar{\mathbf{S}} + \frac{b_0 M}{b_0 + M} (\mathbf{m}_0 - \bar{\mathbf{V}})(\mathbf{m}_0 - \bar{\mathbf{V}})^T, \\
\bar{\mathbf{V}} &= \frac{1}{M} \sum_{j=1}^M \mathbf{V}_j, \quad \bar{\mathbf{S}} = \frac{1}{M} \sum_{j=1}^M (\mathbf{V}_j - \bar{\mathbf{V}})(\mathbf{V}_j - \bar{\mathbf{V}})^T.
\end{aligned}$$

Meanwhile, using the symmetry of the BPMPF model, the full conditional distribution for the user feature vectors can also be easily sampled in parallel from the following distributions:

$$\begin{aligned}
p(\mathbf{U}_i | \mathbf{R}, \mathbf{V}, \Theta_U, \alpha) &\propto \prod_{j=1}^M [N(R_{ij} | \mathbf{U}_i^T \mathbf{V}_j, \alpha^{-1})]^{I_{ij}} N(\mathbf{U}_i | \boldsymbol{\mu}_U, \Lambda_U^{-1}) \\
&\propto N(\mathbf{U}_i | \boldsymbol{\mu}_i^*, [\Lambda_i^*]^{-1}),
\end{aligned} \tag{3.7}$$

where

$$\begin{aligned}
\Lambda_i^* &= \Lambda_U + \alpha \sum_{j=1}^M [\mathbf{V}_j \mathbf{V}_j^T]^{I_{ij}}, \\
\boldsymbol{\mu}_i^* &= [\Lambda_i^*]^{-1} \left(\alpha \sum_{j=1}^M [\mathbf{V}_j R_{ij}]^{I_{ij}} + \Lambda_U \boldsymbol{\mu}_U \right).
\end{aligned}$$

Finally, the posterior over the user hyperparameters is given by the Gaussian-Wishart distribution:

$$\begin{aligned}
p(\Theta_U | \mathbf{U}, \Theta_0) &\propto p(\mathbf{U} | \boldsymbol{\mu}_U, \Lambda_U) p(\Theta_U | \Theta_0) \\
&\propto N(\boldsymbol{\mu}_U | \boldsymbol{\mu}_0^*, (\beta_0^* \Lambda_U)^{-1}) W(\Lambda_U | \mathbf{W}_0^*, \nu_0^*),
\end{aligned} \tag{3.8}$$

where

$$\begin{aligned}\boldsymbol{\mu}_0^* &= \frac{\beta_0 \boldsymbol{\mu}_0 + N \bar{\boldsymbol{U}}}{\beta_0 + N}, \quad \beta_0^* = \beta_0 + N, \quad \nu_0^* = \nu_0 + N, \\ [\mathbf{W}_0^*]^{-1} &= \mathbf{W}_0^{-1} + N \bar{\mathbf{S}} + \frac{\beta_0 N}{\beta_0 + N} (\boldsymbol{\mu}_0 - \bar{\boldsymbol{U}})(\boldsymbol{\mu}_0 - \bar{\boldsymbol{U}})^T, \\ \bar{\boldsymbol{U}} &= \frac{1}{N} \sum_{i=1}^N \mathbf{U}_i, \quad \bar{\mathbf{S}} = \frac{1}{N} \sum_{i=1}^N (\mathbf{U}_i - \bar{\boldsymbol{U}})(\mathbf{U}_i - \bar{\boldsymbol{U}})^T.\end{aligned}$$

Therefore, the Gibbs sampling scheme for the BPFM model can be described as follows:

Gibbs Sampling for Bayesian Probabilistic Matrix Factorization

1. Initialize the model parameters \mathbf{U}^1 and \mathbf{V}^1 at some values.
2. For iterations $t = 1, \dots, T$:
 - Sample the hyperparameters from their full conditional distributions given in equations (3.6) and (3.8):

$$\boldsymbol{\Phi}_V^t \sim p(\boldsymbol{\Phi}_V | \mathbf{V}^t, \boldsymbol{\Phi}_0),$$

$$\boldsymbol{\Theta}_U^t \sim p(\boldsymbol{\Theta}_U | \mathbf{U}^t, \boldsymbol{\Theta}_0).$$

- In parallel, sample the feature vector for item $j = 1, \dots, M$ from its full conditional distribution given in equation (3.5):

$$\mathbf{V}_j^{t+1} \sim p(\mathbf{V}_j | \mathbf{R}, \mathbf{U}^t, \boldsymbol{\Phi}_V, \alpha).$$

- In parallel, sample the feature vector for user $i = 1, \dots, N$ from his full conditional distribution given in equation (3.7):

$$\mathbf{U}_i^{t+1} \sim p(\mathbf{U}_i | \mathbf{R}, \mathbf{V}^{t+1}, \boldsymbol{\Theta}_U, \alpha).$$

Salakhutdinov and Mnih then compared the BPFM model against its PMF model counterparts. Once again using the Netflix Prize dataset, they found that the BPFM

model significantly outperformed the MAP-trained PMF models. Furthermore, they observed that the MAP-trained models began to overfit as the latent feature dimensionality D grew. Conversely, the performance of the BPMF model continued to steadily improve as D increased.

3.2.3 Other Single-Domain PMF and BPMF Extensions

Given how well the PMF and BPMF models perform, it should come as no surprise that several single-domain extensions of these models have been recently proposed. In this subsection, we briefly discuss some of them.

Motivated by the belief that a user’s social connections will affect his judgment and interest in items, Ma et al. (2008) proposed the Social Recommendation (SoRec) algorithm in order to integrate user social network information into the PMF model. Specifically, their algorithm hoped to learn a shared user latent feature space that incorporates both the user social network structure and the user-item rating matrix. This was accomplished by factoring the observed $N \times N$ social network matrix into a latent user feature matrix \mathbf{U} and a latent social network feature matrix \mathbf{Z} , where this matrix \mathbf{U} is the same as the one that is used in the factorization of the observed user-item rating matrix \mathbf{R} in PMF (hence the shared user latent feature space). A zero-mean spherical Gaussian prior was then placed on this \mathbf{Z} matrix and, like the PMF models, MAP estimates were obtained by minimizing an objective function by performing gradient descent.

Meanwhile, Shan and Banerjee (2010) proposed two generalizations of the PMF and BPMF models. First, they investigated using a different prior distribution that was more complex than the PMF model by relaxing the diagonal covariance assumption (i.e., the independent latent features assumption), but simpler than the BPMF model that maintains a distribution over all possible covariance matrices. In doing so, they hoped to achieve a realistic model with a simpler learning process than a

fully Bayesian treatment. Afterwards, Shan and Banerjee tried to leverage side information (e.g., movie cast, user gender, user marital status, etc.) in their predictions by utilizing topic modeling algorithms such as correlated topic models and latent Dirichlet allocation.

Similarly, Porteous et al. (2010) proposed their own extension of the BPMF model that incorporates side information. Unlike Shan and Banerjee (2010), however, they proposed regressing and linearly combining the side information with the final prediction. In addition, the authors introduced a nonparametric Dirichlet process mixture extension to BPMF in order to capture clusters of users or items that were more similar to one another than to others in the population.

Finally, motivated by the idea that rating behavior may evolve over time, Xiong et al. (2010) proposed Bayesian Probabilistic Tensor Factorization (BPTF) as a fully Bayesian extension of BPMF that incorporates temporal relational factors such as seasonality or trends. This was accomplished by indexing the time dimension. Along with the user and item dimensions, this resulted in a three dimensional tensor. BPTF then attempts to factor this tensor into three matrices: the user matrix \mathbf{U} , the item matrix \mathbf{V} , and the time matrix \mathbf{T} . As such, the probabilistic structure and computational scheme of their proposed BPTF model is reminiscent of the standard BPMF model.

3.2.4 Multi-Domain Collaborative Filtering

As we have seen, one of the biggest challenges in the area of recommender systems is the fact that the user-rating matrix is extremely sparse because the number of items being offered vastly outnumbers the number of items that a user has rated. Zhang et al. (2010) attempted to address this issue with their Multi-Domain Collaborative Filtering (MCF) model, which is a multiple-domain extension of the standard PMF model. Specifically, by using PMF to model the ratings in each domain and then

adaptively transferring knowledge across different domains through a learned domain correlation structure, Zhang et al. hoped to alleviate the sparsity problem in any single domain.

We begin by introducing some new notation in order to account for these additional domains. Suppose that we have N users and M total items, where the items can be partitioned into $k = 1, \dots, K$ different domains with M_k items belonging to domain k . In this chapter we only consider the situation where items belong to exactly one domain. Let $\mathbf{R}^k \in \mathbb{R}^{N \times M_k}$ denote the user-item rating matrix for domain k where R_{ij}^k is the rating that user i gave to item j in domain k . As was the case when we were considering single-domain recommender systems, the multiple-domain models that we discuss are sufficiently general enough to handle the most popular rating scales.

Similar to PMF, the goal of MCF is to factorize the user-item rating matrix for each domain R_{ij}^k into a latent user feature matrix $\mathbf{U}^k \in \mathbb{R}^{D \times N}$ and a latent item feature matrix $\mathbf{V}^k \in \mathbb{R}^{D \times M_k}$. That is, column \mathbf{U}_i^k represents user i 's latent feature vector in domain k and column \mathbf{V}_j^k represents item j 's latent feature vector in domain k . Notice that for every domain k , the i^{th} column vector \mathbf{U}_i^k always corresponds to the same user. On the other hand, since items belong to exactly one domain, the j^{th} column vector \mathbf{V}_j^k corresponds to a different item in every domain k . Finally, observe that the dimensionality D is assumed to be the same across all domains.

MCF begins by assuming the following likelihood over the observed ratings in the k^{th} domain:

$$p(\mathbf{R}^k | \mathbf{U}^k, \mathbf{V}^k, \sigma_k^2) = \prod_{i=1}^N \prod_{j=1}^{M_k} \left[N \left(R_{ij}^k | (\mathbf{U}_i^k)^T \mathbf{V}_j^k, \sigma_k^2 \right) \right]^{I_{ij}^k}, \quad (3.9)$$

where I_{ij}^k is the indicator variable that is equal to 1 if user i rated item j in domain k and equal to 0 otherwise. Observe that this likelihood is similar to the likelihood

of the standard PMF model given in equation (3.1), with the main differences being that MCF accounts for multiple domains and parametrizes the normal distribution with a variance parameter instead of a precision parameter.

Like the standard PMF model, MCF proceeds by placing zero-mean spherical Gaussian priors on the user and item features:

$$p(\mathbf{U}^k | \lambda_k) = \prod_{i=1}^N N(U_i^k | \mathbf{0}_D, \lambda_k^2 \mathbf{I}_D),$$

$$p(\mathbf{V}^k | \eta_k) = \prod_{j=1}^{M_k} N(V_j^k | \mathbf{0}_D, \eta_k^2 \mathbf{I}_D).$$

Finally, in order to learn the correlation structure between the different domains, MCF first defines the matrix $\mathbf{U} = [\text{vec}(\mathbf{U}^1), \dots, \text{vec}(\mathbf{U}^K)] \in \mathbb{R}^{ND \times K}$, where $\text{vec}(\cdot)$ denotes the vectorization operator that converts a matrix into a column vector by stacking the columns of the matrix on top of one another, moving from left to right. The following matrix-variate normal distribution is then placed on \mathbf{U} :

$$p(\mathbf{U} | \mathbf{\Omega}) = MN_{ND, K}(\mathbf{U} | \mathbf{0}_{ND \times K}, \mathbf{I}_{ND} \otimes \mathbf{\Omega}),$$

where $MN_{a,b}(\mathbf{M}, \mathbf{A} \otimes \mathbf{B})$ denotes the matrix-variate normal with mean $\mathbf{M} \in \mathbb{R}^{a \times b}$, row covariance matrix $\mathbf{A} \in \mathbb{R}^{a \times a}$, and column covariance matrix $\mathbf{B} \in \mathbb{R}^{b \times b}$. Consequently, the row covariance matrix \mathbf{I}_{ND} models the relationship between user latent features while the column covariance matrix $\mathbf{\Omega}$ captures the relationships between different \mathbf{U}^k terms.

The log-posterior over $\{\mathbf{U}^k\}$ and $\{\mathbf{V}^k\}$ is then given by:

$$\begin{aligned}
& \log p \left(\{\mathbf{U}^k\}, \{\mathbf{V}^k\} \mid \{\mathbf{R}^k\}, \boldsymbol{\sigma}, \boldsymbol{\lambda}, \boldsymbol{\eta}, \boldsymbol{\Omega} \right) \\
&= - \sum_{k=1}^K \frac{1}{2\sigma_k^2} \sum_{i=1}^N \sum_{j=1}^M I_{ij}^k (R_{ij}^k - (\mathbf{U}_i^k)^T \mathbf{V}_j^k)^2 \\
&\quad - \sum_{k=1}^K \frac{1}{2\lambda_k^2} \sum_{i=1}^N (\mathbf{U}_i^k)^T \mathbf{U}_i^k - \sum_{k=1}^K \frac{1}{2\eta_k^2} \sum_{j=1}^M (\mathbf{V}_j^k)^T \mathbf{V}_j^k \\
&\quad - \frac{1}{2} \sum_{k=1}^K \left(\log \sigma_k^2 \sum_{i=1}^N \sum_{j=1}^M I_{ij}^k \right) - \frac{ND}{2} \sum_{k=1}^K \log \lambda_k^2 \\
&\quad - \sum_{k=1}^K \frac{M_k D}{2} \log \eta_k^2 - \frac{1}{2} \text{tr}(\mathbf{U} \boldsymbol{\Omega}^{-1} \mathbf{U}^T) - \frac{ND}{2} \log |\boldsymbol{\Omega}| + C.
\end{aligned}$$

Training the MCF model involves finding MAP estimates of $\{\mathbf{U}^k\}$ and $\{\mathbf{V}^k\}$, and MLE estimates of $\boldsymbol{\sigma}$, $\boldsymbol{\lambda}$, $\boldsymbol{\eta}$, and $\boldsymbol{\Omega}$ by minimizing the following objective function:

$$J \left(\{\mathbf{U}^k\}, \{\mathbf{V}^k\} \mid \{\mathbf{R}^k\}, \boldsymbol{\sigma}, \boldsymbol{\lambda}, \boldsymbol{\eta}, \boldsymbol{\Omega} \right) = -\log p \left(\{\mathbf{U}^k\}, \{\mathbf{V}^k\} \mid \{\mathbf{R}^k\}, \boldsymbol{\sigma}, \boldsymbol{\lambda}, \boldsymbol{\eta}, \boldsymbol{\Omega} \right).$$

The numerical routine used by Zhang et al. is an alternating method where the optimization of each parameter is treated as a separate subproblem: the first derivative of J with respect to a parameter is taken, set equal to 0, and then solved with respect to that parameter. This process is then iterated through for each parameter in the model and repeated until convergence. For more specific details on the MCF model and estimation scheme, as well as an extension of MCF that incorporates a link function, see their paper.

Due to the lack of a publicly available dataset with distinct domains, Zhang et al. evaluated the performance of MCF using movie data from MovieLens and defining domains via the movie genres (e.g., Action or Drama).³ On this constructed dataset, MCF showed promising results: compared to the standard PMF model that

³ <http://grouplens.org/datasets/movielens/>

it extends, the MCF model was able to obtain a higher predictive accuracy (in terms of RMSE) in every single domain.

3.3 Proposed Model

Whereas Zhang et al. (2010) introduced a multiple-domain extension of the standard PMF model, in this section we propose a multiple-domain extension of the standard BPFM model. Like MCF, we allow each domain k to have its own latent user feature matrix \mathbf{U}^k and latent item feature matrix \mathbf{V}^k . Unlike MCF, however, we allow the dimensionality D_k to vary across domains. This is potentially beneficial because it allows us to alleviate overfitting by specifying a lower dimensionality in certain domains. Therefore, $\mathbf{U}^k \in \mathbb{R}^{D_k \times N}$ and $\mathbf{V}^k \in \mathbb{R}^{D_k \times M_k}$, where once again column \mathbf{U}_i^k represents user i 's latent feature vector in domain k and column \mathbf{V}_j^k represents item j 's latent feature vector in domain k .

We begin by assuming the following likelihood over the observed ratings in the k^{th} domain:

$$p(\mathbf{R}^k | \mathbf{U}^k, \mathbf{V}^k, \alpha_k) = \prod_{i=1}^N \prod_{j=1}^{M_k} \left[N \left(R_{ij}^k | (\mathbf{U}_i^k)^T \mathbf{V}_j^k, \alpha_k^{-1} \right) \right]^{I_{ij}^k}, \quad (3.10)$$

where α_k is a precision parameter for domain k . Although this likelihood is similar to the one for MCF that was given in equation (3.9), notice that we adopt the BPFM convention and parametrize the normal distribution using a precision term instead of a variance term.

Then, like BPFM, we place the following hierarchical structure on the item feature vectors in domains $k = 1, \dots, K$:

$$p(\mathbf{V}^k | \mathbf{m}_V^k, \Omega_V^k) = \prod_{j=1}^{M_k} N \left(\mathbf{V}_j^k | \mathbf{m}_V^k, (\Omega_V^k)^{-1} \right),$$

$$p(\Phi_V^k | \Phi_0^k) = N(\mathbf{m}_V^k | \mathbf{m}_0^k, (b_0^k \Omega_V^k)^{-1}) \mathcal{W}(\Omega_V^k | \Psi_0^k, n_0^k),$$

where we have defined $\Phi_V^k = \{\mathbf{m}_V^k, \Omega_V^k\}$ and $\Phi_0^k = \{\mathbf{m}_0^k, n_0^k, \Psi_0^k\}$ for each domain $k = 1, \dots, K$. Notice that there are K different parameter sets for both Φ_V^k and Φ_0^k , and that they do not necessarily have to be equal across all of the domains. This provides us with the flexibility to specify different prior beliefs across the different domains if we so choose (e.g., perhaps we believe that movie preferences tend to be more spread out than book preferences).

Next, recall that each user has K different latent feature vectors (one for each domain). Ideally, we would like to share information amongst the K different feature vectors belonging to the same user. To accomplish this, we propose first stacking the K user feature matrices in a column-wise manner. This forms the stacked matrix $\mathbf{U} \in \mathbb{R}^{(D_1 + \dots + D_K) \times N}$ as follows:

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}^1 \\ \mathbf{U}^2 \\ \vdots \\ \mathbf{U}^K \end{pmatrix},$$

where the i^{th} column \mathbf{U}_i is the column vector that arises had we stacked the latent feature vectors of user i across all K domains. Notice that we have defined this matrix \mathbf{U} slightly differently than the one that was introduced in Section 3.2.4 for the MCF model, which explains why the matrices have different dimensions.

In order to capture correlations among a user's latent features (both within and across domains), we place the following hierarchical structure over this stacked matrix:

$$\begin{aligned} p(\mathbf{U} | \boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U) &= \prod_{i=1}^N N(\mathbf{U}_i | \boldsymbol{\mu}_U, (\boldsymbol{\Lambda}_U)^{-1}), \\ p(\boldsymbol{\Theta}_U | \boldsymbol{\Theta}_0) &= N(\boldsymbol{\mu}_U | \boldsymbol{\mu}_0, (\beta_0 \boldsymbol{\Lambda}_U)^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_U | \mathbf{W}_0, \nu_0), \end{aligned} \quad (3.11)$$

where we have once again defined $\boldsymbol{\Theta}_U = \{\boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U\}$ and $\boldsymbol{\Theta}_0 = \{\boldsymbol{\mu}_0, \nu_0, \mathbf{W}_0\}$. Notice that unlike our item hyperparameters where we had K different sets (one for each

domain), here there is only one set of user hyperparameters since users belong to all domains.

Finally, observe that if there is only 1 domain in the dataset, then the proposed model reduces down to the standard BPFM model.

3.3.1 Posterior Inference

Like BPFM, due to the ease of sampling from the full conditional distributions, we opt for a Gibbs sampling algorithm in order to approximate the intractable predictive distribution of rating R_{ij}^{k*} that user i gives to item j in domain k .

The full conditional for \mathbf{V}_j^k is straightforward, and it has virtually the same form as the full conditional for item features in the standard BPFM model—we just need to make the domain explicit. That is, for domains $k = 1, \dots, K$:

$$\begin{aligned} p(\mathbf{V}_j^k | \mathbf{R}^k, \mathbf{U}^k, \Phi_V^k, \alpha^k) &\propto \prod_{i=1}^N \left[N \left(R_{ij}^k | (\mathbf{U}_i^k)^T \mathbf{V}_j^k, \alpha_k^{-1} \right) \right]^{I_{ij}^k} N \left(\mathbf{V}_j^k | \mathbf{m}_V^k, (\Omega_V^k)^{-1} \right) \\ &\propto N \left(\mathbf{V}_j^k | \mathbf{m}_j^{k*}, [\Omega_j^{k*}]^{-1} \right), \end{aligned} \quad (3.12)$$

where

$$\begin{aligned} \Omega_j^{k*} &= \Omega_V^k + \alpha^k \sum_{i=1}^N \left[\mathbf{U}_i^k (\mathbf{U}_i^k)^T \right]^{I_{ij}^k}, \\ \mathbf{m}_j^{k*} &= [\Omega_j^{k*}]^{-1} \left(\alpha^k \sum_{i=1}^N \left[\mathbf{U}_i^k R_{ij}^k \right]^{I_{ij}^k} + \Omega_V^k \mathbf{m}_V^k \right). \end{aligned}$$

And like the BPFM model, notice that the full conditional over the item latent feature matrix \mathbf{V}^k for domain k factorizes:

$$p(\mathbf{V}^k | \mathbf{R}^k, \mathbf{U}^k, \Phi_V^k, \alpha^k) = \prod_{j=1}^{M_k} p(\mathbf{V}_j^k | \mathbf{R}^k, \mathbf{U}^k, \Phi_V^k, \alpha^k).$$

Therefore, we can speed up the MCMC by sampling from the full conditional distributions for item feature vectors *within the same domain* in parallel. In fact, because

the full conditional distributions for items *in different domains* are also independent, additional speedups can be obtained by sampling from all M item feature vector full conditional distributions in parallel, regardless of their domain.

Similarly, the full conditional distributions for the item hyperparameters have closed forms that are almost identical to their BPMF model counterparts. Once again, we just need to make the domain explicit. For domains $k = 1, \dots, K$:

$$\begin{aligned} p(\Phi_V^k | \mathbf{V}^k, \Phi_0^k) &\propto p(\mathbf{V}^k | \mathbf{m}_V^k, \Omega_V^k) p(\Phi_V^k | \Psi_0^k) \\ &\propto N(\mathbf{m}_V^k | \mathbf{m}_0^{k*}, (b_0^{k*} \Omega_V^k)^{-1}) W(\Omega_V^k | \Phi_0^{k*}, n_0^{k*}), \end{aligned} \quad (3.13)$$

where

$$\begin{aligned} \mathbf{m}_0^{k*} &= \frac{b_0^k \mathbf{m}_0^k + M_k \bar{\mathbf{V}}^k}{b_0^k + M_k}, \quad b_0^{k*} = b_0^k + M_k, \quad n_0^{k*} = n_0^k + M_k, \\ \left[\Psi_0^{k*} \right]^{-1} &= (\Psi_0^k)^{-1} + M_k \bar{\mathbf{S}}^k + \frac{b_0^k M_k}{b_0^k + M_k} (\mathbf{m}_0^k - \bar{\mathbf{V}}^k) (\mathbf{m}_0^k - \bar{\mathbf{V}}^k)^T, \\ \bar{\mathbf{V}}^k &= \frac{1}{M_k} \sum_{j=1}^{M_k} \mathbf{V}_j^k, \quad \bar{\mathbf{S}}^k = \frac{1}{M_k} \sum_{j=1}^{M_k} (\mathbf{V}_j^k - \bar{\mathbf{V}}^k) (\mathbf{V}_j^k - \bar{\mathbf{V}}^k)^T. \end{aligned}$$

And because the full conditionals for these item hyperparameters Φ_V^k are independent of each other for all domains k , they can also be sampled in parallel in order to speed up the MCMC.

On the other hand, the full conditional distribution for user i 's feature vector \mathbf{U}_i^k is not as straightforward as the BPMF case since his preferences in domain k are now related to his preferences in all of the other domains, which we will denote as \mathbf{U}_i^{-k} . Observe, however, that if we let $\Sigma_U = (\Lambda_U)^{-1}$ be the corresponding covariance matrix for the multivariate normal prior that was placed on the stacked vector \mathbf{U}_i

in equation (3.11), then we can partition this prior distribution as follows:

$$\begin{aligned}
p(\mathbf{U}_i | \boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U) &= N(\mathbf{U}_i | \boldsymbol{\mu}_U, (\boldsymbol{\Lambda}_U)^{-1}) \\
&= N(\mathbf{U}_i | \boldsymbol{\mu}_U, \boldsymbol{\Sigma}_U) \\
&= N\left(\begin{pmatrix} \boldsymbol{\mu}_U^k \\ \boldsymbol{\mu}_U^{-k} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_U^{k,k} & \boldsymbol{\Sigma}_U^{k,-k} \\ \boldsymbol{\Sigma}_U^{-k,k} & \boldsymbol{\Sigma}_U^{-k,-k} \end{pmatrix}\right)
\end{aligned}$$

Therefore, using the properties of multivariate normal distributions, the prior distribution in (3.11) elicits the following multivariate normal distribution for \mathbf{U}_i^k conditional on \mathbf{U}_i^{-k} (i.e., user i 's latent features in domain k conditional on his latent features in all of the other domains):

$$p(\mathbf{U}_i^k | \mathbf{U}_i^{-k}, \boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U) = N\left(\mathbf{U}_i^k | \hat{\boldsymbol{\mu}}_U^k, [\hat{\boldsymbol{\Lambda}}_U^k]^{-1}\right), \quad (3.14)$$

where

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_U^k &= \boldsymbol{\mu}_U^k + \boldsymbol{\Sigma}_U^{k,-k} (\boldsymbol{\Sigma}_U^{-k,-k})^{-1} (\mathbf{U}_i^{-k} - \boldsymbol{\mu}_U^{-k}), \\
\hat{\boldsymbol{\Lambda}}_U^k &= \left[\boldsymbol{\Sigma}_U^{k,k} - \boldsymbol{\Sigma}_U^{k,-k} (\boldsymbol{\Sigma}_U^{-k,-k})^{-1} \boldsymbol{\Sigma}_U^{-k,k} \right]^{-1}.
\end{aligned}$$

Using equation (3.14), we can now derive the full conditional distribution for the vector \mathbf{U}_i^k . For domains $k = 1, \dots, K$:

$$\begin{aligned}
p(\mathbf{U}_i^k | \mathbf{R}^k, \mathbf{V}^k, \mathbf{U}_i^{-k}, \boldsymbol{\Theta}_U, \alpha_k) &\propto \prod_{j=1}^{M_k} [N(R_{ij}^k | (\mathbf{U}_i^k)^T \mathbf{V}_j^k, \alpha_k^{-1})]^{I_{ij}^k} N\left(\mathbf{U}_i^k | \hat{\boldsymbol{\mu}}_U^k, [\hat{\boldsymbol{\Lambda}}_U^k]^{-1}\right) \\
&\propto N\left(\mathbf{U}_i^k | \boldsymbol{\mu}_i^{k*}, [\boldsymbol{\Lambda}_i^{k*}]^{-1}\right), \quad (3.15)
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{\Lambda}_i^{k*} &= \hat{\boldsymbol{\Lambda}}_U^k + \alpha^k \sum_{j=1}^{M_k} [\mathbf{V}_j^k (\mathbf{V}_j^k)^T]^{I_{ij}^k}, \\
\boldsymbol{\mu}_i^{k*} &= [\boldsymbol{\Lambda}_i^{k*}]^{-1} \left(\alpha^k \sum_{j=1}^{M_k} [\mathbf{V}_j^k R_{ij}^k]^{I_{ij}^k} + \hat{\boldsymbol{\Lambda}}_U^k \hat{\boldsymbol{\mu}}_U^k \right).
\end{aligned}$$

Therefore, to sample a new stacked latent feature vector for user i , we simply have to sample from the full conditional for his latent preferences \mathbf{U}_i^k in every domain $k = 1, \dots, K$. And although a user's feature vectors *in different domains* cannot be sampled in parallel, the feature vectors for different users *within each domain* k can be sampled in parallel in order to speed up the MCMC.

Finally, the full conditional distribution over the user feature vector hyperparameters is given by the following Gaussian-Wishart distribution:

$$\begin{aligned} p(\Theta_U | \mathbf{U}, \Theta_0) &\propto p(\mathbf{U} | \boldsymbol{\mu}_U, \boldsymbol{\Lambda}_U) p(\Theta_U | \Theta_0) \\ &\propto N(\boldsymbol{\mu}_U | \boldsymbol{\mu}_0^*, (\beta_0^* \boldsymbol{\Lambda}_U)^{-1}) W(\boldsymbol{\Lambda}_U | \mathbf{W}_0^*, \nu_0^*), \end{aligned} \quad (3.16)$$

where

$$\begin{aligned} \boldsymbol{\mu}_0^* &= \frac{\beta_0 \boldsymbol{\mu}_0 + N \bar{\mathbf{U}}}{\beta_0 + N}, \quad \beta_0^* = \beta_0 + N, \quad \nu_0^* = \nu_0 + N, \\ [\mathbf{W}_0^*]^{-1} &= \mathbf{W}_0^{-1} + N \bar{\mathbf{S}} + \frac{\beta_0 N}{\beta_0 + N} (\boldsymbol{\mu}_0 - \bar{\mathbf{U}})(\boldsymbol{\mu}_0 - \bar{\mathbf{U}})^T, \\ \bar{\mathbf{U}} &= \frac{1}{N} \sum_{i=1}^N \mathbf{U}_i, \quad \bar{\mathbf{S}} = \frac{1}{N} \sum_{i=1}^N (\mathbf{U}_i - \bar{\mathbf{U}})(\mathbf{U}_i - \bar{\mathbf{U}})^T. \end{aligned}$$

Therefore, the Gibbs sampling scheme for our proposed model proceeds as follows:

Gibbs Sampling for the Proposed Model

1. Initialize the model parameters $\{\mathbf{U}^k\}^1$ and $\{\mathbf{V}^k\}^1$ at some values.
2. For iterations $t = 1, \dots, T$:
 - In parallel, sample the hyperparameters from their full conditional distributions given in equations (3.13) and (3.16):
$$(\Phi_V^k)^t \sim p(\Phi_V^k | (\mathbf{V}^k)^t, \Phi_0^k) \text{ for each domain } k = 1, \dots, K,$$

$$(\Theta_U)^t \sim p(\Theta_U | (\mathbf{U})^t, \Theta_0).$$

- For each domain $k = 1, \dots, K$:
 - In parallel, sample the feature vector for item $j = 1, \dots, M_k$ from its full conditional distribution given in equation (3.12):

$$(\mathbf{V}_j^k)^{t+1} \sim p(\mathbf{V}_j^k | \mathbf{R}^k, (\mathbf{U}^k)^t, (\Phi_V^k)^t, \alpha^k).$$

- In parallel, sample the feature vector for user $i = 1, \dots, N$ from his full conditional distribution given in equation (3.15):

$$(\mathbf{U}_i^k)^{t+1} \sim p(\mathbf{U}_i^k | \mathbf{R}^k, (\mathbf{V}^k)^{t+1}, (\mathbf{U}_i^{-k})^t, (\Theta_U)^t, \alpha_k),$$

$$\text{where } (\mathbf{U}_i^{-k})^t = ((\mathbf{U}_i^1)^{t+1}, \dots, (\mathbf{U}_i^{k-1})^{t+1}, (\mathbf{U}_i^{k+1})^t, \dots, (\mathbf{U}_i^K)^t).$$

3.4 Experiments

One of the biggest obstacles to research in cross-domain recommender systems has been the lack of a publicly available dataset with distinct domains. For this reason, we use movie ratings from the MovieLens 100K dataset and create domains based upon the movie genres. Although not ideal, this specific dataset and approach has been used previously in the emerging cross-domain literature.

In its original form, the MovieLens 100K dataset consists of 100,000 ratings from 943 users on 1682 movies.⁴ Movies can fall into 19 genres (including an “unknown” genre), and they can belong to several different genres at once (e.g., an action-comedy movie).

We compared the following models in a multiple-domain setting at various levels of the latent dimensionality D :

- **Multi-Domain Collaborative Filtering (MCF):** This is the model proposed by Zhang et al. (2010). For this model, the only parameter that needs to be explicitly set is the latent dimensionality D ; the other parameters were found via the alternating method described in Section 3.2.4.
- **Independent BPMF (BPMF-I):** This is the standard BPMF model learned separately on each domain with latent dimensionality D . Following Salakhutdinov and Mnih (2008b), in each domain we set $\boldsymbol{\mu}_0 = \mathbf{m}_0 = \mathbf{0}$, $\nu_0 = n_0 = D$, $\mathbf{W}_0 = \boldsymbol{\Phi}_0 = \mathbf{I}_D$, and we fixed the observation noise precision at $\alpha = 2$.
- **Pooled BPMF (BPMF-P):** This is the standard BPMF model learned with latent dimensionality D after pooling all of the data into a single domain. The same hyperparameter values as BPMF-I were used.
- **Multi-Domain BPMF (BPMF-MD):** This is the proposed model that generalizes the standard BPMF model to multiple domains. For this model, we tried to set the hyperparameters to values that were comparable to the ones set in the BPMF-I and BPMF-P models described above. Therefore, we restricted our attention to the situation where the latent feature dimension is the same in all domains (i.e., $D_k = D$ for all k). In addition, we set the user hyperparameters as $\boldsymbol{\mu}_0 = \mathbf{0}$, $\nu_0 = KD$, and $\mathbf{W}_0 = \mathbf{I}_{KD}$. Meanwhile, for the item

⁴ <http://grouplens.org/datasets/movielens/>

hyperparameters we use $\mathbf{m}_0^k = \mathbf{0}$, $n_0^k = D$, and $\Phi_0^k = \mathbf{I}_D$ for all domains k .

Finally, for all domains k the observation noise precision was fixed at $\alpha^k = 2$.

In our experiments, each model was learned for latent dimensions $D = 1, 2, \dots, 20$. Notice that for any fixed D , every model’s rating prediction is computed through the inner product of two D dimensional vectors.

The code used to learn the MCF and the standard BPFM models were provided by the authors. Finally, notice that the BPFM-I and BPFM-P models can be looked at as two “naive” ways of handling the multiple-domain setting.

3.4.1 Cross-Domain Recommendations

In order to evaluate the performance of the models in a cross-domain situation, we first wanted to construct a dataset with domains that were as distinct as possible—which would typically be the case in a truly multiple-domain setting.⁵

To accomplish this, we first determined the subset of movies in our dataset that fell uniquely into any one genre, leaving us with 831 movies. Of these movies, the five genres with the most ratings were Drama, Comedy, Horror, Thriller, and Action. We then further reduced our dataset to the movies that fell distinctly into one of these five genres. A summary of this constructed dataset appears in Table 3.1.

Next, we randomly selected 80% of the ratings as the training set while the remaining 20% was used as the test set. Each model’s overall performance on this test set (in terms of RMSE) for varying levels of the latent feature dimensionality D is shown in Fig. 3.4.1. From this figure, we see that the performance of the Bayesian models continues to improve as the feature dimensionality grows. The MAP-trained MCF model, however, starts to heavily overfit as D gets too large. These results are consistent with the findings in Salakhutdinov and Mnih (2008b).

⁵ Consequently, the dataset we end up constructing differs from the one that Zhang et al. (2010) consider.

Table 3.1: Summary of the multiple-domain MovieLens data that we have constructed.

	Action	Comedy	Drama	Horror	Thriller	Total
Number of Movies	32	210	376	43	36	697
Number of Ratings	879	9828	13257	1558	1094	26616

Furthermore, we see that proposed BPFM-MD model outperforms every other model for all values of D . In particular, by outperforming the BPFM-I and BPFM-P models that naively handle the multiple-domain setting, the proposed BPFM-MD model shows the promising future that cross-domain recommender systems hold.

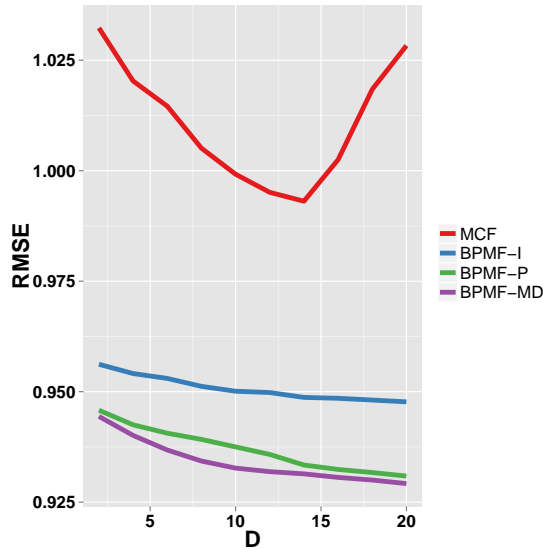


FIGURE 3.1: Comparison of each model’s overall test RMSE in the cross-domain recommendation situation for different levels of the latent feature dimensionality D . Notice that the performance of the Bayesian models continues to improve as D increases, but the MAP-trained MCF model begins to overfit. Finally, observe that the proposed BPFM-MD model outperforms every other model for all values of D .

Finally, Table 3.2 provides a more detailed look at each model’s best overall performance and the feature dimensionality under which it occurred. Interestingly, although BPFM-MD outperforms BPFM-P overall, it does perform slightly worse on some domains.

Table 3.2: Breakdown of each method’s best RMSE scores and the feature dimensionality under which it occurred in the cross-domain recommendation situation.

Model	D	Action	Comedy	Drama	Horror	Thriller	Total
MCF	14	1.0200	1.0338	0.9649	1.0026	0.9211	0.9931
BPMF-I	20	0.9732	1.0025	0.9008	0.9675	0.9507	0.9477
BPMF-P	20	0.9632	0.9782	0.8974	0.9673	0.8652	0.9317
BPMF-MD	20	0.9600	0.9803	0.8905	0.9474	0.8807	0.9297

3.4.2 *Semi-cold-start Recommendations*

Cross-domain recommender systems may also be useful in addressing the semi-cold-start problem. As defined in Winoto and Tang (2008), the semi-cold-start problem occurs when the recommender system has ratings from a group of users on one domain (e.g., movies), but does not have their ratings in a different target domain (e.g., music).⁶ Therefore, the goal of the recommender system is to leverage the information that it has available to improve predictions in the target domain. From a business perspective, these semi-cold-start recommendations can be used to provide a serendipitous aspect to the recommender system that can help improve user experience by suggesting novel products that they do not expect, but might still enjoy.

To simulate this situation, we considered the ratings from the largest two domains (drama and comedy) of the dataset that we constructed in the previous example. As can be seen from Table 3.1, this left us with 23,085 ratings from 942 users on 586 movies. We then randomly selected 150 users, placing all of their drama ratings in the training set and all of their comedy ratings in the test set. Afterwards, we randomly selected 150 different user, putting all of their drama ratings in the test set and all of their comedy ratings in the training set. Finally, the drama and comedy ratings for all of the other users was placed in the training set.

⁶ The recommender system is also assumed to have a group of users with ratings in both domains

Fig. 3.4.2 shows the overall performance of the models on the test set (in terms of RMSE) under varying levels of the latent feature dimensionality D . As in the previous example, we see that the performance of the Bayesian models improves as D increases, while the MAP-trained MCF model begins to overfit. And once again, we see that the proposed BPMF-MD model continues to outperform the other models at all levels of D .

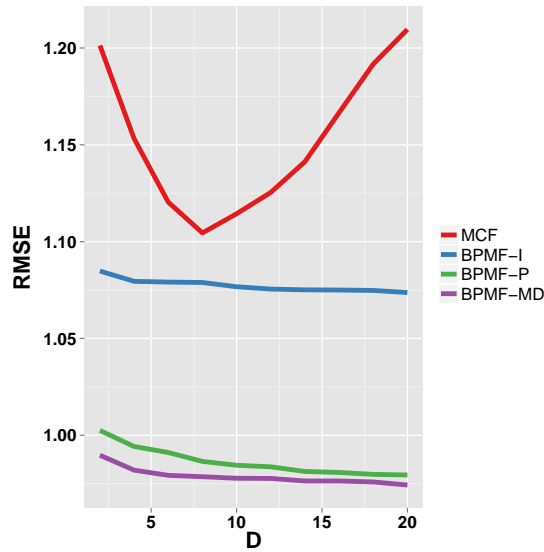


FIGURE 3.2: Comparison of each model’s overall test RMSE in the semi-cold-start recommendation situation for different levels of the latent feature dimensionality D . Once again, the performance of the Bayesian models continues to improve as D increases, while the MAP-trained MCF model begins to overfit. And like the previous example, the proposed BPMF-MD model outperforms every other model for all values of D .

Finally, Table 3.3 presents a more detailed look at each model’s best overall performance and the feature dimensionality under which it occurred. This time we see that the proposed BPMF-MD model performs the best in all domains.

Table 3.3: Breakdown of each method’s best RMSE scores and the feature dimensionality that it occurred under in the semi-cold-start recommendation situation.

Model	D	Comedy	Drama	Total
MCF	8	1.1221	1.0933	1.1045
BPMF-I	20	1.0353	1.0969	1.0737
BPMF-P	20	1.0126	0.9583	0.9795
BPMF-MD	20	0.9953	0.9552	0.9713

3.5 Conclusions

In this chapter, we presented a multiple-domain extension of the BPMF model that was first introduced by Salakhutdinov and Mnih (2008b). Afterwards, we showed that the proposed model outperforms other similar linear-factor based models in two different cross-domain recommendation situations, which were created by partitioning a movie dataset into separate domains based upon movie genres. Consequently, although a major obstacle to research in this area has been the lack of a publicly available multiple-domain dataset, our results highlight the promising future that cross-domain recommender systems holds.

Text Mining: A Trayvon Martin and Political Blogs Case Study

4.1 Introduction

At its core, computational advertising is about deciding which online ad would be best to display to any given user. One element that often influences this decision is the textual information on the websites that a user is browsing. Consequently, although it was originally developed in the 1980s, text mining has found new life in this past decade. Indeed, many of its recent advances are directly attributable to the rise of the Internet and the emergence of computational advertising. In this chapter, we discuss some of the tools and challenges of text mining. In particular, we use the Trayvon Martin shooting incident as a case study in analyzing the lexical content and network connectivity structure of the political blogosphere.

Although the Trayvon Martin event and the set of websites that we investigate are not necessarily aligned with computational advertising, the tools that we use can be applied more broadly in order to better understand consumer preferences, and to track how online content evolves over time. Indeed, a growing front in com-

putational advertising seeks to use text mining and network analysis to help select relevant online ads. For example, a user searching for the phrase “cars” could either be interested in purchasing a new vehicle or in buying a new DVD. Here the lexical content of a user’s recently traversed web pages—whether related to vehicles or animated films—can be used to help decide which advertisement would be most suitable to display. Furthermore, emerging online trends—such as a viral car review (e.g., the controversial review of the Tesla Model S that appeared in the *New York Times*) or a renewed interest in Disney movies—can be used to select the best advertisement for a user with a sparser browsing history, or to identify influential web properties that may be particularly worthwhile to include in advertising campaigns.

Because it will be referenced extensively throughout this chapter, we begin by reviewing the Trayvon Martin incident in more detail in Section 4.2. Afterwards, Section 4.3 discusses how the data we use in this chapter was collected, what information the data contains, and what issues of data quality exist. Section 4.4 then presents the results of our data analysis. Finally, Section 4.5 concludes.

4.2 Trayvon Martin Shooting Incident

The Trayvon Martin shooting was a recent event that ignited protests across the United States, and sparked national debate over issues such as racism, gun control, media coverage, and statutory self-defense laws. This section briefly reviews this incident. For a more detailed overview, see the Wikipedia (2014) entry from which much of this summary was derived.

4.2.1 *The Shooting*

On the rainy night of February 26, 2012, George Zimmerman (a 28 year old Hispanic American neighborhood watchman) observed Trayvon Martin (a 17 year old African American teenager) walking through his gated community in Sanford, Florida. Due

to a recent series of break-ins in the neighborhood, and believing Martin to be acting suspiciously, Zimmerman called the Sanford police non-emergency number to report the activity. Shortly afterwards, Martin began running and Zimmerman promptly followed. However, after he was told by the dispatcher to stop chasing after Martin, Zimmerman ended his pursuit and awaited police assistance. At this point, the call between Zimmerman and the dispatcher concluded.

Minutes later and before police arrived, however, a violent encounter between Martin and Zimmerman took place, which resulted in Zimmerman fatally shooting Martin. As it turns out, Martin was unarmed and had been temporarily staying in the neighborhood with his father and his father's fiancée.

Upon arriving at the scene, police observed that Zimmerman was bleeding from the nose and the back of his head. Upon questioning, Zimmerman admitted to shooting Martin in self-defense—claiming that he had been attacked first, and that he had yelled for help prior to discharging his firearm. Exactly whether it had been Zimmerman or Martin that had been calling for help, however, has been the subject of much debate and remains a mystery, with witnesses and experts offering conflicting testimonies of what actually transpired.

Nevertheless, after a Sanford Police investigation, it was determined that there was not enough evidence to dispute Zimmerman's account of the event, and he was not charged with any crime.

4.2.2 Media Coverage and Public Response

In the weeks following the incident, Zimmerman remained a free man and the case received only local coverage. Starting in the beginning of March, however, efforts by the Martin team eventually proved successful in garnering national attention—with notable outlets such as Reuters, *The Huffington Post*, and CNN providing this early coverage.

Unfortunately, early reporting of the story was largely criticized for prematurely influencing public opinion of the case, and some accused the media of trying to frame the incident as a racially motivated killing and as a “Good vs. Evil” struggle. For example, initial images of Martin were provided by his family, and depicted him as a smiling teenager. In contrast, early images of Zimmerman were taken from his 2005 mugshot, and portrayed him as an unhappy and angry man. Furthermore, initial reports characterized Zimmerman’s race as white, despite the fact that he had personally identified himself as Hispanic on public documents such as his driver’s license and voting records.

There were also issues with how the media presented Zimmerman’s phone conversation with the police dispatcher in the moments leading up to the incident. In particular, NBC eventually was forced to apologize for mistakenly editing the call in such a way that made it appear as if Zimmerman had racially profiled Martin by readily volunteering the information that Martin was black. In reality, Zimmerman had provided this information only after the dispatcher specifically asked him “is [Martin] black, white, or Hispanic?”, to which Zimmerman responded “He looks black.”

Meanwhile, the Martin camp also objected to some aspects of the coverage, and later blamed the media for running a smear campaign against Martin. For example, Martin’s mother accused the media of “trying to kill [Trayvon’s] reputation” after it was reported Martin had been serving a ten-day suspension for a marijuana incident at the time of his death. Furthermore, Martin’s supporters also accused the media of trying to portray Martin as a gangster after several pictures of drugs, guns, and Martin making obscene gestures were released, while pictures of him dressed up for prom, fishing with his father, and celebrating his birthday were not.

After the case exploded onto the national scene, protests were held across the country calling for Zimmerman’s arrest—with hoodies (which Martin was wearing

at the time of his death) being the most notable and extensively used rallying symbol. Other popular symbols also included bags of Skittles and cans of Arizona Iced Tea, which were items that Martin had reportedly purchased from a nearby convenience store shortly before his death (although, in actuality, the beverage that Martin had bought was a can of Arizona Watermelon Fruit Juice Cocktail).

Statements made by several prominent public figures and organizations also served to further inflame the situation. Spike Lee, for example, accidentally forced an elderly couple from their home when he erroneously identified their address as Zimmerman's in a public tweet. And when asked to give his thoughts on the incident, President Barack Obama famously commented "If I had a son, he would look like Trayvon"—a remark for which he was both lauded and criticized.

The intense media coverage and public outcry eventually led to a reinvestigation of the case. Finally, on April 11, 2012, and more than a month after the fatal shooting actually occurred, Zimmerman was arrested and charged with second-degree murder.

4.2.3 Court Case and Statutory Self-Defense Laws

State prosecutors opened the case by filing an affidavit of probable cause: alleging that Zimmerman had profiled, confronted, and fatally shot Martin even though he had not been committing any crime. On the other hand, Zimmerman's defense team initially sought to have the charges dismissed based on Florida's "stand your ground" law, which permits someone to use deadly force—without requiring them to first attempt to retreat—when he reasonably feels at risk of great bodily harm. Ultimately, however, they decided to use "self-defense" as their official argument, arguing that Zimmerman had not even had the opportunity to retreat as he was being restrained by Martin at the time.

Finally, on July 13, 2013 the jury returned a verdict of not guilty for the charge of second-degree murder, as well as for the lesser charge of manslaughter. Public

response to the verdict was predictably mixed: Zimmerman’s supporters rejoiced while his critics protested and called on the Department of Justice to intervene. Furthermore, national polls found that the public reaction was sharply divided along racial and political lines.

The Zimmerman trial also brought controversial “stand your ground” laws and other statutory self-defense laws (e.g., the “castle doctrine,” which 46 states have adopted) under intense national scrutiny. Supporters of these laws have argued that they protect a necessary right, while their detractors have called them “shoot first” laws that needlessly expand the definition of self-defense. Although these laws have not yet been repealed, many are now under review to have their language clarified in order to prevent such a situation from occurring again.

4.3 Data

4.3.1 Data Collection

Justin Gross, Assistant Professor of Political Science at the University of North Carolina at Chapel Hill provided us with a list of the top 1509 political blog domains as ranked by Technorati as of January 13, 2013.¹

Technorati is an Internet search engine that indexes over 100 million blog domains that are broken down into several categories (e.g., politics, technology, business, etc.). Technorati then computes a proprietary “Technorati Authority” metric for each blog, which is a rating between 0 and 1000 (with 1000 being the highest) measuring a domain’s “standing & influence in the blogosphere.” According to Technorati’s Frequently Asked Questions (FAQ),² factors that contribute to this rating include “linking behavior, categorization and other associated data over a short, finite period of time.” Consequently, a domain’s rating “may rapidly rise and fall

¹ www.technorati.com

² www.technorati.com/what-is-technorati-authority/

depending on what the blogosphere is discussing at the moment, and how often a site produces content being referenced by other sites.”

This list of political blog domains was then given to MaxPoint Interactive,³ a startup technology company in the Research Triangle Area who agreed to scrape and tokenize text from the blog entries that were posted on these domains between January 1, 2012 and December 31, 2012. While accessing and scraping the websites, MaxPoint declared itself as a robot and followed each domain’s Robots Exclusion Protocol (i.e., the set of instructions and guidelines that the site owners give to any third-party robots). At the time of this chapter, the corpus contained 114,611 documents (i.e., individual blog entries) coming from 467 of the 1509 top ranked political domains.

Due to the size of the corpus, as well as the desire to analyze the political blogosphere with regards to a singular event, we decided to focus on the Trayvon Martin shooting incident. There were several reasons for selecting this particular event over others.⁴ First, coverage of the story was extremely prevalent and sustained throughout the 2012 calendar year. Second, the incident was characterized by issues that were particularly polarizing across the nation—such as the issues of racism and gun control. Finally, it was thought that the uniqueness of the name “Trayvon” would help to simplify the search for the documents that were relevant to the event: a document referring to the shooting would almost surely contain the word “Trayvon,” and it is unlikely that a document using the word “Trayvon” would be referring to a different event. In the full corpus, 1103 documents coming from 145 different domains mentioned the word “Trayvon” at least once.

³ www.maxpoint.com

⁴ Other events that we considered included the 2012 presidential election, the Benghazi terrorist attack, the Sandy Hook Elementary School shooting, and the Aurora movie theater shooting.

4.3.2 Data Fields

After the raw HTML of a blog post has been scraped and parsed by MaxPoint, the data fields associated with it are as follows:

- The name of the domain that the blog post appeared on.
- The estimated date on which the blog entry was posted.
- How the estimated date of the blog post was obtained.
- The raw text that was scraped from the blog post.
- The tokenized text that was obtained by using the Snowball stemmer (Porter, 2001) on the raw text.
- Any links to other domains that appeared in the blog post.

We now discuss each of these data fields in more detail.

The name of the domain that the blog post appeared on is simply a string identifying the unique organization or entity owning the entry (e.g., *The Huffington Post* or Business Insider). In our dataset, these domains belong to one of the original 1509 top ranked political blog domains from Technorati that was provided to us by Professor Gross. As mentioned previously, however, the full dataset (i.e., not just the documents related to the Trayvon Martin incident) currently contains only 467 of these domains. This is a consequence of obstacles that have existed in scraping some of these domains—for example the domain requested not to be scraped in their Robots Exclusion Protocol, the domain had an unusual HTML structure that made it particularly difficult to scrape and/or parse, etc.

Among the domains that were scraped, however, nuances in the HTML structure made it challenging to obtain accurate timestamps for some of the blog entries.

Consequently, MaxPoint attempted to estimate these dates via the following the sequential search method:

1. **URL:** Check the URL to see whether a date is embedded there;
2. **Body:** Check the body of the blog post and take the first date found;
3. **Global:** Check the page as a whole and take the first date found.

Notice that this search method is prioritized in order of perceived accuracy. For each document, both the date and the method used to acquire it were recorded.

The raw text of the blog post was also stored. Afterwards, MaxPoint employed the Snowball stemmer (Porter, 2001) to trim inflected words to their root form. This process, known as stemming and tokenizing the text, is statistically helpful because it reduces the number of features (words) present in the data by removing tense, pluralization, hyphenation, and other linguistic subtleties. For example, after stemming and tokenization, words like “jump”, “jumps”, and “jumping” will all be identified by the same token of “jump.”. According to studies done by the Text REtrieval Conference (TREC) at the National Institute of Standards and Technology (NIST) (such as Robertson et al. (2000)), simple stemming rules can adequately handle approximately 85% of cases, with further improvement requiring case-by-case handling.

Finally, any links to other domains that appeared in the blog entry were also recorded.

4.3.3 Data Quality and Data Cleaning

Due to the sheer number of domains being analyzed, considerable differences existed in the HTML codes that were being scraped. This made it particularly difficult to parse the data in a consistent manner, leading to some data quality challenges that we discuss in this subsection.

Because blog entries coming from the same domain tended to have a similar HTML structure, the types of errors being committed were thought to be highly correlated among documents coming from the same domain. Consequently, a random sample of 145 entries (one from each unique domain mentioning “Trayvon”) were hand-checked by undergraduate majors in political science for any errors in document content, dates, etc. Their results were then used to help identify and clean any data quality issues that were discovered (e.g., by including domain-specific rules for the parser to follow).

One of the major challenges in parsing the data involved difficulties in determining which part of the scraped HTML actually corresponded to the blog post’s entry. Specifically, errors fell into one of three categories: discarding parts of the actual blog entry, including elements that were not part of the actual blog entry (e.g., reader comments), or identifying an erroneous part of the page as being the blog entry. The second and third issues were particularly problematic as they introduced a considerable amount of noise into the data—such as discussions occurring that are not directly related to the blog entry, advertisements by “spambots,” new vocabulary resulting from linguistic oddities in the reader comments , etc.

Another data quality issue that was identified involved the sequential search method that was employed to estimate the document timestamps. For example, consider Fig. 4.1, which shows a histogram of the timestamps for all of the documents in the Trayvon Martin corpus. Notice that two of the documents have been dated as February 2, 2012 and February 26, 2012—highly suspect, as these timestamps correspond to dates that occur on or before the actual shooting, which is represented in the figure by a vertical dotted line. Upon closer inspection, it was observed that these two erroneous timestamps were obtained from the body of the blog entry (i.e., the second step in the sequential search method). Specifically, one of the documents references an earlier call made by Zimmerman to the police regard-

ing suspicious activity that occurred on February 2 while the other document cites February 26 as the actual day of the fatal shooting.

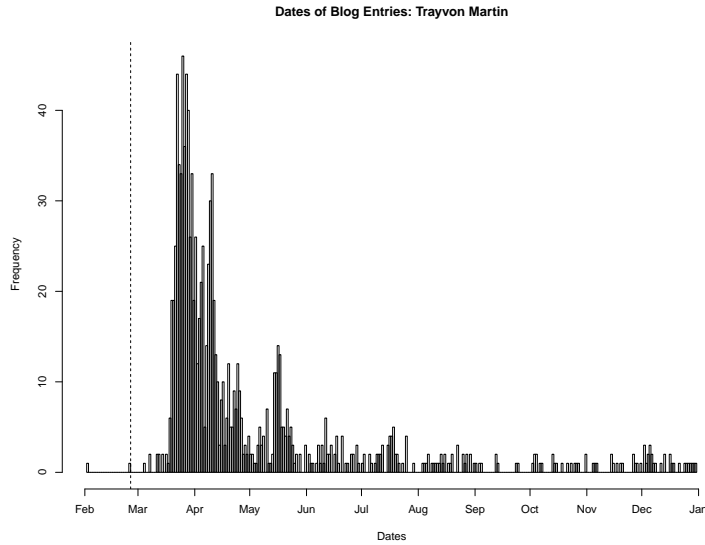


FIGURE 4.1: Histogram of the estimated timestamps for the documents in the Trayvon Martin corpus, where the vertical dotted line corresponds to February 26, 2012 (the date the shooting actually occurred). Notice that two of the documents have been dated before the actual incident occurred, which is clearly an error committed by the sequential search method.

In total, approximately 15% of the posts were found to be problematic to some degree. Although some adjustments were made to how the data was processed, issues and errors still remain. However, because the tools that we apply are expected to be able to cope with noisy data to some extent, we proceeded with our analysis.

4.4 Data Analysis

In this section we discuss the text mining tools that we applied to the Trayvon Martin corpus. These tools allowed us to identify n -grams, infer document sentiment, discover the abstract “topics” that are being discussed, and investigate the network structure of the blogosphere.

4.4.1 Multi-word Expressions: n -grams

The first goal of our analysis was to identify the important n -grams characterizing the Trayvon Martin incident, which are the contiguous sequences of n tokens having an unusually high probability of occurring together when compared to other sequences of the same length. Traditionally, approaches to this problem have followed the method introduced in Manning and Schütze (1999), and are based on hypothesis tests of a multinomial contingency table. These hypothesis tests, however, typically introduce a bias because they test all n -grams of the same length simultaneously.

Blei and Lafferty (2009) recently proposed their turbo topics model in order to address this issue. Although turbo topics was originally presented as an extension of topic modeling that aids in the visualization of topics by identifying topic-specific multi-word expressions, our application of the model assumed only a single topic. This allowed us to identify n -grams for the entire Trayvon Martin corpus while still maintaining the statistical properties and computational efficiency of their model. Consequently, our discussion differs slightly from the one presented in their paper.

Turbo topics is based on a language model of arbitrary length expressions, and it recursively uses distribution-free nested permutation tests in order to find significant phrases. Specifically, given a corpus of N total words coming from a vocabulary containing V distinct words, the log likelihood is given by:

$$\log p(w) = \sum_{n=1}^N \log p(w_n | w_1, \dots, w_{n-1}). \quad (4.1)$$

There are two extremes when considering this model. The first is to specify the fully parametrized model containing a conditional distribution of words given each possible history.⁵ Unfortunately, this approach is computationally intractable as it requires specifying at least V^N parameters, and there will typically not be enough

⁵ A word's history are the words preceding it.

data to estimate these parameters. Meanwhile, the second extreme is to model each word as being independent of its history, $p(w_n|w_1, \dots, w_{n-1}) = p(w_n)$. This model, however, is incapable of capturing the dependencies between words.

Turbo topics is a compromise between these two extremes, and it attempts to parametrize the full model by conditioning on word histories of varying lengths. This is accomplished by “backing off” to models with smaller histories under certain conditions, which is a procedure that was first introduced by Katz (1987). In order to use this procedure to evaluate the likelihood in equation (4.1), we require a distribution over words that can be conditioned on an arbitrary history of previous words. Let $w_{1:n}$ denote a length n history, and let $S_{w_{1:n}}$ be the set of words that are governed by history-specific probabilities (e.g., if the history $w_{1:n} = \text{“new”}$, then $S_{\text{“new”}}$ might be $\{\text{“york”}, \text{“jersey”}, \text{“hampshire”}, \text{“mexico”}\}$). This distribution is assumed to be:

$$p(w_{n+1}|w_{1:n}) = \begin{cases} \pi_{w_{n+1}|w_{1:n}} & \text{if } w_{n+1} \in S_{w_{1:n}} \\ \gamma_{w_{1:n}} p(w_{n+1}|w_{2:n}) & \text{otherwise,} \end{cases} \quad (4.2)$$

where the constant $\gamma_{w_{1:n}}$ ensures that the distribution sums to one,

$$\gamma_{w_{1:n}} = \frac{1 - \sum_{v \in S_{w_{1:n}}} \pi_v|u}{1 - \sum_{v \in S_{w_{1:n}}} p(v|w_{2:n})} \quad (4.3)$$

Intuitively, the turbo topics procedure then proceeds as follows:

1. Fix the order of the tokens in each document in the corpus

$$w_1, w_2, w_3, w_4, \dots$$

2. Given a token w , run a hypothesis testing procedure to identify the tokens w' that are likely to precede or follow w . When the test is significant, this results in either the n -gram (w', w) or (w, w') being created.
3. Repeat step 2 until no new significant tokens are added.

In step 2, turbo topics uses a greedy recursive search algorithm to look for n -grams. Specifically, turbo topics begins by assuming a basic unigram model. Then, given some current model, it attempts to expand to a more complicated model by trying to include the next most likely n -gram. When evaluating whether this n -gram should be added to the current model, turbo topics computes the log likelihood ratio under the current model both with and without the candidate n -gram. Afterwards, it performs a hypothesis test. If this test is significant, then the n -gram is added to the current model. This process is repeated until no new significant n -grams are added.

The exact hypothesis test that is used in the turbo topics model is a recursive permutation test. This test involves first randomly shuffling all of the words in such a way that still retains the sequence of words that are currently modeled. This is done in order to remove any spurious dependencies present in the data, while still retaining the dependencies that are already assumed by the current model. Afterwards, the log likelihood ratio described in the previous paragraph is computed for this shuffled data. A shuffled log likelihood ratio less than the original ratio serves as evidence that the n -gram was significant. Conversely, a shuffled log likelihood ratio greater than the original ratio is treated as evidence against. Repeating this process several times and computing the proportion of shuffled ratios greater than the original ratio provides a p -value. If this p -value is less than some specified threshold, then the n -gram is added to the current model.

Unlike many other hypothesis tests, this permutation test does not rely on assumptions about the asymptotic distribution of the test statistic—a particularly desirable property in this sparse data setting. Furthermore, another advantage of the recursive permutation tests used in turbo topics is that it tests each candidate n -gram individually while still accounting for other n -grams that have already been added to the model. This helps to avoid the bias that is introduced by some other

Table 4.1: The most frequently used n -grams in the Trayvon Martin corpus that were identified by the turbo topics model (table reproduced from Soriano et al. (2013)).

georg zimmerman	trayvon martin	self defens
year old	dont know	stand ground law
presid obama	stand ground	african american
look like	unit state	trayvon martin case
barack obama	dont think	law enforc
fox news	hate crime	civil right

methods that test all n -grams of the same length simultaneously.

To help illustrate this last point, suppose that the word “new” occurs 10,000 times in the corpus, the word “york” follows it 6000 times, and the word “jersey” follows it 2000 times. After adding “new york” as a bigram to the model, turbo topics will then see the 2000 out of 4000 occurrences of “new jersey” as a very strong signal of a bigram—and much stronger than the 2000 out of 10,000 occurrences of “new jersey” had we not updated our model with “new york.”

Table 4.1 shows the most frequently used n -grams in the Trayvon Martin corpus that were identified by the turbo topics model. Ideally, one would also like to be able to map certain n -grams to the same token—for example “presid obama” and “barack obama” both refer to President Barack Obama and should be treated as the same. The Latent Semantic Indexing (LSI) procedure described in Deerwester et al. (1990) that aims to identify words with similar meanings (synonymy) and words with multiple meanings (polysemy) provides one potential path forward, although we did not explore this further in our analysis.

4.4.2 *Sentiment Analysis and Word Usage*

Often political blogs have an element of subjectivity. Sentiment analysis is one method that attempts to classify a document’s polarity—whether the expressed opinion is positive, negative, or neutral—by comparing the words in the document against

a dictionary of scored terms. The dictionary used in our analysis of the Trayvon Martin corpus was the AFINN list proposed by Nielsen (2011), which scores 2477 terms rated on a -5 to 5 scale (negation terms such as “not” flip the score). In order to be consistent with the corpus, the Snowball stemmer was applied to this list of words.

Let d_w be the number of occurrences of word w in our dictionary, where w has a score of s_w . The polarity of a document is then defined as:

$$\text{polarity} = \frac{\sum_w d_w s_w}{\sum_w d_w |s_w|} \in [-1, 1], \quad (4.4)$$

so that $\text{polarity} \in [-1, 1]$. The more positive a document is, the closer its polarity will be to 1 . Similarly, the more negative a document is, the closer its polarity will be to -1 . Meanwhile, documents with a polarity near 0 are considered neutral.

When analyzing the Trayvon Martin corpus, however, inherently negative terms such as “gun” and “kill” may not necessarily be indicative of a document’s polarity. Therefore, terms like these were removed from the dictionary prior to any analysis. Some of these removed words included:

arrest	assault	attack	bullet	confront
crime	dead	death	defens	die
gun	fatal	ignor	injuri	kill
killer	manslaught	murder	punch	shoot

The polarity of all the documents in the Trayvon Martin corpus appears in Fig. 4.2, where we used ± 0.15 as the cutoffs for classifying documents as positive, negative, or neutral. In the end 199 documents were classified as negative, 286 documents were classified as positive, and 618 were classified as neutral.

After classifying the documents, we investigated whether there were terms that tended to appear more often in positive documents, and whether there were terms that tended to appear more often in negative documents. This was done using

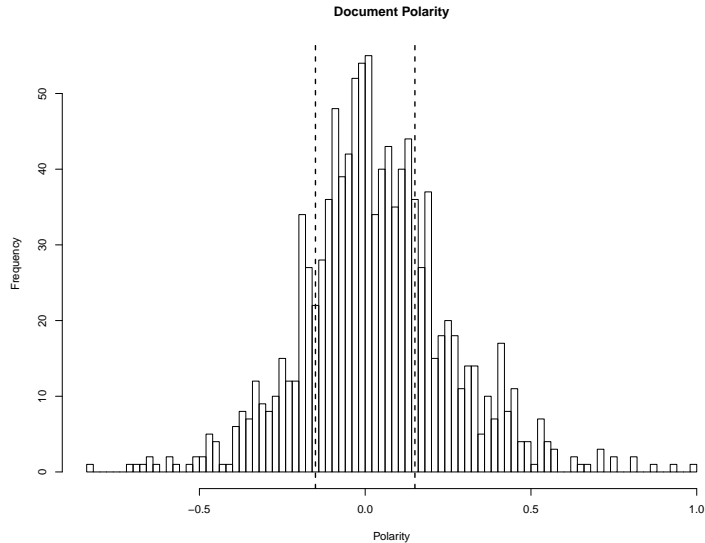


FIGURE 4.2: Histogram of the polarity scores of the documents in the Trayvon Martin corpus. Vertical dotted lines appear at ± 0.15 , and indicate the cutoff points used for classifying the documents as either positive, negative, or neutral.

a two-proportion z -test. Specifically, for each token in the corpus let \hat{p}_+ be the proportion of positive documents containing that token, \hat{p}_- be the proportion of negative documents containing that token, and \hat{p} be the pooled proportion of positive and negative documents containing that token. The two-proportion z -statistic for testing $H_0 : p_+ = p_-$ is then given by:

$$z = \frac{\hat{p}_+ - \hat{p}_-}{\sqrt{\hat{p}(1 - \hat{p})\left(\frac{1}{n_+} + \frac{1}{n_-}\right)}}. \quad (4.5)$$

Terms that had $z > 1.96$ were deemed to occur significantly more often in positive documents. Meanwhile, those that had $z < -1.96$ were labeled as occurring significantly more often in negative documents.

The wordclouds pictured in Fig. 4.3 help to illustrate how terms were used across documents in the Trayvon Martin corpus. Notice that documents with a negative sentiment seemed to focus on George Zimmerman, his actions, and their consequences

corpus is defined as the set of documents $\{D_1, \dots, D_m\}$ being analyzed. The goal of LDA is to use the textual information in the corpus to represent the documents in terms of a set of latent topics, which are modeled as unknown multinomial distributions over the vocabulary that need to be inferred from the data. Early LDA assumed that the number of unknown latent topics is fixed at some integer K .

The data generating process for the basic LDA model can be described as follows:

1. For each topic k :
 - (a) Draw a vector of word proportions $\phi_k \sim Dir_V(\alpha)$. This determines the relative “weights” of each term in topic k .
2. For each document D_j :
 - (a) Draw a vector of topic proportions $\theta_j \sim Dir_K(\beta)$. This determines the extent to which document D_j is composed of each of the K topics.
 - (b) For each word w_i in document j :
 - i. Draw a topic assignment $z_i \sim Mult(\theta_j)$.
 - ii. Draw a word from the topic $w_i \sim Mult(\phi_{z_i})$.

This generative process can be visually represented with the plate diagram in Fig. 4.4, which depicts the conditional dependence structure among the random variables. Notice that the only observed data are the words coming from the documents in the corpus.

The goal of LDA is to invert this generative model and find the posterior distri-

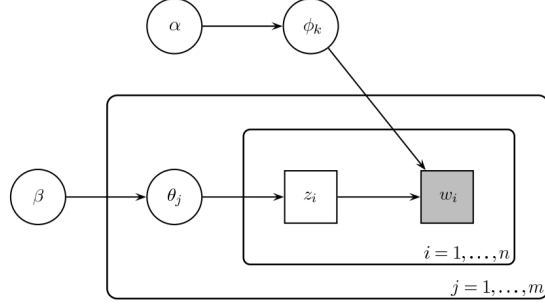


FIGURE 4.4: The plate diagram for the basic LDA generative process (figure reproduced from Soriano et al. (2013)).

bution of the latent variables conditional on the documents:

$$\begin{aligned}
 & p(\boldsymbol{\theta}_{1:m}; z_{1:m,1:n}; \boldsymbol{\phi}_{1:K} \mid w_{1:m,1:n}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \\
 & \frac{p(\boldsymbol{\theta}_{1:m}; z_{1:m}; \boldsymbol{\phi}_{1:K} \mid w_{1:m}; \boldsymbol{\alpha}, \boldsymbol{\beta})}{\int \boldsymbol{\phi}_{1:K} \int \boldsymbol{\theta}_{1:m} \sum_{z_{1:m,1:n}} p(\boldsymbol{\theta}_{1:m}; z_{1:m}; \boldsymbol{\phi}_{1:K} \mid w_{1:m}; \boldsymbol{\alpha}, \boldsymbol{\beta})}. \tag{4.6}
 \end{aligned}$$

Because the denominator is intractable, however, the solution requires Markov Chain Monte Carlo or variational inference methods. For example, Blei et al. (2003) used a variational Bayes approximation approach while Griffiths and Steyvers (2004) proposed a collapsed Gibbs sampler that integrates out the $\boldsymbol{\theta}_j$ and $\boldsymbol{\phi}_k$.

Besides the turbo topics model discussed in Section 4.4.1, there have also been numerous extensions of topic modeling that have been proposed. For example, Blei et al. (2004) proposed hierarchical LDA (hLDA) to join topics together in a hierarchy using the nested Chinese restaurant process. A couple of years later, Blei and Lafferty (2006) introduced a dynamic extension to topic models in order to help analyze and visualize the time evolution of topics. Meanwhile, Blei and McAuliffe (2008) proposed supervised LDA (sLDA) to jointly model the topics of a corpus and a response variable that is associated with each document (e.g., movie ratings, website view

counts, etc.). And more recently, Chang and Blei (2010) proposed the relational topic model (RTM) to summarize a network of documents, predict links between them, and predict words within them. Indeed, one of the strengths of topic modeling is that it is highly modular, and the extensions mentioned are just some of the models that an advertiser may be interested in.

For our Trayvon Martin corpus, the decision was made to fit the basic LDA model with five latent topics after attempts to fit the more sophisticated LDA models resulted in topics that were more difficult to interpret. The top 15 tokens in each topic can be seen in Table 4.2.

Table 4.2: The top 15 words in each topic found by LDA for our Trayvon Martin dataset (Table reproduced from Soriano et al. (2013)).

[1]	[2]	[3]	[4]	[5]
money	obama	think	georg zimmerman	trayvon martin
log	presid	dont	trayvon martin	black
scream	state	comment	polic	white
donat	law	would	law	georg zimmerman
dave	american	even	gun	media
voic	year	peopl	case	news
perjuri	alec	like	would	sharpton
expert	govern	one	self defens	racial
bond	gun	make	said	year
paypal	group	get	charg	hoodi
owen	republican	thing	prosecutor	look
bail	democrat	know	evid	said
forens	war	use	shot	fox
omara	nation	point	state	death
websit	legisl	say	shoot	race

Impressionistically, Topic 1 appears to focus on the legal trial (O’Mara is Zimmerman’s defense attorney, Owen is an audio technician who concluded that the scream was not Zimmerman’s, Zimmerman’s PayPal account became a legal issue, etc.) while Topic 2 seems to relate to the political aspects of the story (such as gun

control and statutory self-defense laws). Meanwhile, Topic 3 looks like it absorbs unspecific and ambiguous words—perhaps capturing the “blog structure” and the social function of blogging. Topic 4 seems to house the facts regarding the incident. Finally, Topic 5 reflects the racial aspect.

Next, from a dynamic network perspective, we wanted to investigate how the interactions among the blog domains involved the four primary topics (excluding Topic 3) identified by LDA. In addition, we wanted to study how real time events and a domain’s political orientation and stature affected the behavior of the blogosphere.

The following figures are stills taken from a movie that we created in order to show the evolution of the discourse among the 145 blog domains in the Trayvon Martin corpus. Here the domains are represented as nodes, while the directed edges indicate the direction of the reference: the edge originates from the domain being referenced and points to the domain that is making the reference. The domains are also grouped into separate arcs by political orientation, which were provided by Professor Justin Gross—the top arc consists of the conservative domains, the lower left arc is composed of the moderate domains, and the liberal domains are represented in the lower right arc. Within each arc, domains are also further organized according to their Technorati Authority, which we use as a measure of a domain’s stature within the political blogosphere. Here the more influential domains have larger nodes that are located closer to the center of their arcs. Finally, the colors of the nodes indicate the LDA topic that was most dominant on the blog domain for that day.

The first still in Fig. 4.5 shows the state of the blogosphere on March 21, 2012. Here we see that most blogs have not yet picked up on the Trayvon Martin incident, and the few that have appear to be focusing mainly on the actual event details. This is not too surprising, as the story had just started to become national news at this time. We can observe, however, that the more authoritative blogs were instrumental in garnering national attention to the incident: the majority of the domains that

reported on the event lie in the center of the arcs, and most of the links appear to be originating from these centers as well. Furthermore, we can also see that links are more likely to appear amongst those holding similar political paradigms, although the moderate blogs do appear to be more willing to cite those from both sides of the political aisle.

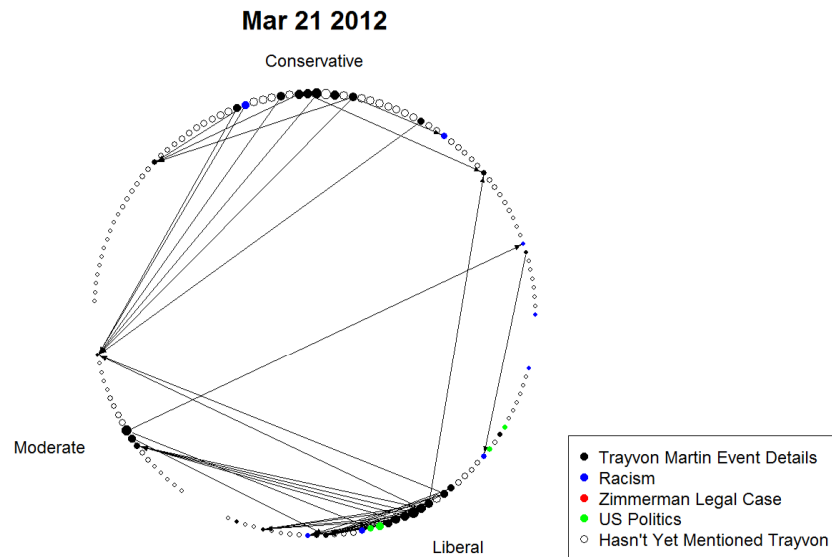


FIGURE 4.5: Trayvon Martin Blogosphere on March 21, 2012 just as the incident started to become national news (figure reproduced from Soriano et al. (2013)). Early coverage of the story seems to be mostly focused on the actual details of the event, and the more authoritative blogs appear to be driving the discussion.

The second still in Fig. 4.6 depicts the blogosphere on April 11, 2012, which is the date that Zimmerman was arrested and charged with second degree murder. The blogosphere is incredibly active, and the discussion appears to have shifted towards the racial and political aspects of the story. Once again, the linking behavior suggests

that the more authoritative blogs appear to be driving the discussion, with links still more likely to occur along party lines—particularly amongst the Conservative blogs.

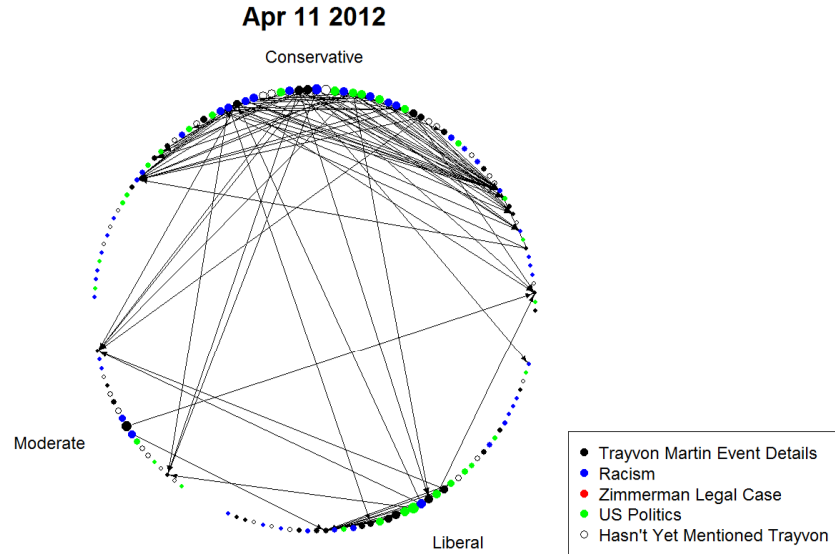


FIGURE 4.6: Trayvon Martin Blogosphere on April 11, 2012—the day that Zimmerman was finally charged and arrested (figure reproduced from Soriano et al. (2013)). The discussion seems to have shifted towards the racial and political topics.

The final still in Fig. 4.7 is from May 17, 2012, the day that prosecutors first publicly released evidence from the case (which included police and autopsy reports, surveillance videos, and witness statements). Although the linking behavior remains consistent with what we have already seen, interestingly the Zimmerman case itself does not appear to dominate the blogosphere as one may expect. In fact, over the course of the entire incident, we found that the legal case was rarely the main topic of discussion. Perhaps this is because it is hard to disentangle the legal topic from the

racial and political ones—which are arguably the topics that came to characterize the incident, and were responsible for bring it to national prominence in the first place.

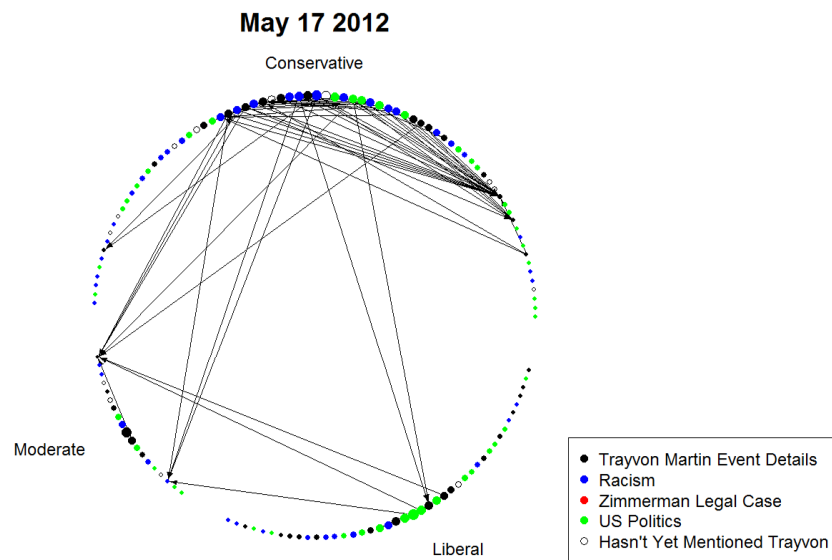


FIGURE 4.7: Trayvon Martin Blogosphere on May 17, 2012 when prosecutors first publicly released evidence from the case (figure reproduced from Soriano et al. (2013)). Despite this fact, discussion still seems to be focused on the racial and political aspects of the incident rather than the legal one.

4.5 Conclusions

Computational advertising is a rapidly emerging discipline, and text mining has been one extremely important component that has fueled its growth. In this chapter, we have explored some of the text mining tools that allow us to identify n -grams, infer document sentiment, discover abstract topics, and investigate network structures.

Unfortunately, as powerful as text mining is, it does have its limitations. As we have documented, there are significant challenges that can occur when trying to assemble and analyze a real dataset coming from websites.

Furthermore, it is important to keep in mind that no single off-the-shelf solution will meet the demands of every advertising campaign. Indeed, for the foreseeable future, the most successful campaigns will be the ones that are able to effectively incorporate human knowledge and product expertise alongside the insights that are gained from applying statistical models.

Concluding Remarks

This thesis discussed three problems in computational advertising whose solutions help to determine the “best” online ad to display to any given user.

Chapter 2 covered the topic of auctions, which is the mechanism that ad exchanges use to buy and sell online advertisements. In particular, we proposed the Backwards Indifference Derivation (BID) algorithm to numerically approximate the pure strategy Nash equilibrium bidding functions in an independent private value first-price sealed-bid auction where bidders draw their types from continuous and atomless distributions. The BID algorithm was motivated in part by the results derived in Athey (2001), and proceeded by attempting to construct a sequence of finite-action equilibria that converges to the continuum-action solution. Using numerical examples, we showed that the BID algorithm produces solutions that are consistent with economic theory, and afterwards we investigated how ad auctions can be modified in order to generate more revenue.

Chapter 3 focused on the area of recommender systems, which can be used to help identify the most cost-effective ad to display to users. We then proposed a cross-domain recommender system that was a multiple-domain extension of the

Bayesian Probabilistic Matrix Factorization model proposed by Salakhutdinov and Mnih (2008b). Afterwards, we showed that the proposed model outperforms other similar linear-factor based models in two different cross-domain recommendation situations, which highlights the promising future that cross-domain recommender systems holds.

Chapter 4 discussed some of the tools and challenges of text mining, a field which can be used to better understand consumer preferences and track how online content evolves over time. In particular, the Trayvon Martin shooting incident was used as a case study in analyzing the lexical content and network connectivity structure of the political blogosphere.

There are, of course, many other problems in the computational advertising space. For example, despite advertisers' best efforts, the vast majority of online ads do not elicit a desired response (e.g., a click or a sale). Consequently, ad retargeting has become one extremely important area of computational advertising research. On the other hand, advertisers have also become wary of online ads that receive too many responses, which may be the result of fraudulent activity (e.g., click fraud). This has motivated research into identifying and combating this malicious activity, which is just one example of a problem in computational advertising that is not necessarily concerned with determining the "best" online ad to display.

New opportunities for computational advertising research have also been created by recent advances in technology. The rapid growth of smartphones and tablets, for instance, has opened up new research areas in the realm of mobile advertising, such as location-based targeting and mobile game advertising. Meanwhile, startup companies like Coursera and Snapchat continue to present fresh challenges as they brainstorm new ways of generating revenue for their innovative products.

Indeed, one of the most exciting aspects of computational advertising has been the breadth, depth, and impact of the problems being considered. Given how quickly

technology emerges and develops, this is a characteristic of computational advertising that is likely to continue.

Bibliography

- Athey, S. (2001), “Single Crossing Properties and the Existence of Pure Strategy Equilibria in Games of Incomplete Information,” *Econometrica*, 69, 861–89.
- Bajari, P. (2001), “Comparing competition and collusion: a numerical approach,” *Economic Theory*, 18, 187–205.
- Blei, D. and McAuliffe, J. (2008), “Supervised Topic Models,” in *Advances in Neural Information Processing Systems 20*, eds. J. Platt, D. Koller, Y. Singer, and S. Roweis, pp. 121–128, MIT Press, Cambridge, MA.
- Blei, D., Griffiths, T., Jordan, M., and Tenenbaum, J. (2004), “Hierarchical Topic Models and the Nested Chinese Restaurant Process,” *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*.
- Blei, D. M. and Lafferty, J. D. (2006), “Dynamic Topic Models,” in *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, pp. 113–120, New York, NY, USA, ACM.
- Blei, D. M. and Lafferty, J. D. (2009), “Visualizing Topics with Multi-Word Expressions,” *Journal of the American Society for Information Science*.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003), “Latent dirichlet allocation,” *Journal of Machine Learning Research*, 3, 993–1022.
- Breese, J. S., Heckerman, D., and Kadie, C. (1998), “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98*, pp. 43–52, Morgan Kaufmann Publishers Inc.
- Chang, J. and Blei, D. (2010), “Hierarchical relational models for document networks,” *Annals of Applied Statistics*, 4, 124–150.
- Cheng, H. (2006), “Ranking sealed high-bid and open asymmetric auctions,” *Journal of Mathematical Economics*, 42, 471–498.
- Cox, J. C., Smith, V. L., and Walker, J. M. (1982), “Auction market theory of heterogeneous bidders,” *Economic Letters*, 9, 319–325.

- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990), "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, 41, 391–407.
- Edelman, B., Ostrovsky, M., and Schwarz, M. (2007), "Internet Advertising and the Generalized Second Price Auction: Selling Billions of Dollars Worth of Keywords," *American Economic Review*, 97, 242–259.
- Fibich, G. and Gaviols, A. (2003), "Asymmetric First-Price Auctions - A Perturbation Approach," *Mathematics of Operations Research*, 28, 836–852.
- Fibich, G. and Gavish, N. (2011), "Numerical simulations of asymmetric first-price auctions," *Games and Economic Behavior*, 73, 479–495.
- Gayle, W. and Richard, J. (2008), "Numerical Solutions of Asymmetric, First-Price, Independent Private Values Auctions," *Computational Economics*, 32, 245–278.
- Griesmer, J. H., Levitan, R. E., and Shubik, M. (1967), "Toward a study of bidding processes part IV - games with unknown costs," *Naval Research Logistics Quarterly*, 14, 415–433.
- Griffiths, T. L. and Steyvers, M. (2004), "Finding scientific topics," *Proceedings of the National Academy of Sciences*, 101, 5228–5235.
- Guerre, E., Perrigne, I., and Vuong, Q. (2009), "Nonparametric Identification of Risk Aversion in First-Price Auctions Under Exclusion Restrictions," *Econometrica*, 77, 1193–1227.
- Hofmann, T. (1999), "Probabilistic Latent Semantic Analysis," in *Proceedings of the 15th Conference Uncertainty in Artificial Intelligence*, pp. 289–296.
- Hubbard, T. P. and Paarsch, H. J. (2009), "Investigating bid preferences at low-price, sealed-bid auctions with endogenous participation," *International Journal of Industrial Organization*, 27, 1–14.
- Hubbard, T. P. and Paarsch, H. J. (2014), "On the Numerical Solution of Equilibria in Auction Models with Asymmetries within the Private-Values Paradigm," in *Handbook of Computational Economics Vol. 3*, vol. 3, pp. 37–115, Elsevier.
- Hubbard, T. P., Kirkegaard, R., and Paarsch, H. J. (2013), "Using Economic Theory to Guide Numerical Analysis: Solving for Equilibria in Models of Asymmetric First-Price Auctions," *Computational Economics*, 42, 241–266.
- Jackson, M. O. and Swinkels, J. M. (2005), "Existence of Equilibrium in Single and Double Private Value Auctions," *Econometrica*, 73, 93–139.

- Jennings, A. and Higuchi, H. (1992), “A Personal News Service based on a User Model Neural Network,” *IEICE Transactions on Information and Systems*, 75, 192–209.
- Kaplan, T. R. and Zamir, S. (2012), “Asymmetric first-price auctions with uniform distributions: analytic solutions to the general case,” *Economic Theory*, 50, 269–302.
- Karlin, S. (1968), *Total Positivity, Vol. I*, Standford University Press.
- Katz, S. M. (1987), “Estimation of probabilities from sparse data for the language model component of a speech recognizer,” in *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 400–401.
- Kim, J. W., Lee, B. H., Shaw, M. J., Chang, H.-L., and Nelson, M. (2001), “Application of Decision-Tree Induction Techniques to Personalized Advertisements on Internet Storefronts,” *International Journal of Electronic Commerce*, 5, 45–62.
- Kirkegaard, R. (2009), “Asymmetric first price auctions,” *Journal of Economic Theory*, 144, 1617–1635.
- Koren, Y. (2009), “The BellKor Solution to the Netflix Grand Prize,” .
- Krishna, V. (2002), *Auction Theory*, Academic Press.
- Lebrun, B. (1996), “Existence of an equilibrium in first price auctions,” *Economic Theory*, 7, 421–443.
- Lebrun, B. (1999), “First-price Auction in the Asymmetric N Bidder Case,” *International Economic Review*, 40, 125–142.
- Lebrun, B. (2006), “Uniqueness of the equilibrium in first-price auctionse,” *Games and Economic Behavior*, 55, 131–151.
- Li, H. and Riley, J. G. (2007), “Auction choice,” *International Journal of Industrial Organization*, 25, 1269–1298.
- Ma, H., Yang, H., Lyu, M. R., and King, I. (2008), “SoRec: Social Recommendation Using Probabilistic Matrix Factorization,” in *Proceedings of the ACM Conference on Information and Knowledge Management*, vol. 17.
- Manning, C. D. and Schütze, H. (1999), *Foundations of Statistical Natural Language Processing*, The MIT Press.
- Marlin, B. (2004), “Modeling User Rating Profiles For Collaborative Filtering,” in *Advances in Neural Information Processing Systems 16*, eds. S. Thrun, L. Saul, and B. Schölkopf, Cambridge, MA, MIT Press.

- Marlin, B. and Zemel, R. S. (2004), “The Multiple Multiplicative Factor Model for Collaborative Filtering,” in *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04.
- Marshall, R. C., Meurer, M. J., Richard, J., and Stromquist, W. (1994), “Numerical Analysis of Asymmetric First Price Auctions,” *Games and Economic Behavior*, 7, 193–220.
- Maskin, E. and Riley, J. (1984), “Optimal Auctions with Risk Averse Buyers,” *Econometrica*, 52, 1473–1518.
- Maskin, E. and Riley, J. (2000), “Equilibrium in Sealed High Bid Auctions,” *Review of Economic Studies*, 67, 439–454.
- Milgrom, P. and Shannon, C. (1994), “Monotone Comparative Statics,” *Econometrica*, 62, 157–180.
- Nati, N. S. and Jaakkola, T. (2003), “Weighted Low-Rank Approximations,” in *Proceedings of the International Conference on Machine Learning*, vol. 20.
- Nielsen, F. Å. (2011), “AFINN,” <http://www2.imm.dtu.dk/pubdb/p.php?6010>.
- Pazzani, M., Muramatsu, J., and Billsus, D. (1996), “Syskill & Webert: Identifying interesting web sites,” in *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, pp. 54–61, AAAI Press.
- Plum, M. (1992), “Characterization and computation of Nash-equilibria for auctions with incomplete information,” *International Journal of Game Theory*, 20, 393–418.
- Porteous, I., Asuncion, A., and Welling, M. (2010), “Bayesian Matrix Factorization with Side Information and Dirichlet Process Mixtures,” in *Association for the Advancement of Artificial Intelligence*.
- Porter, M. F. (2001), “Snowball: A language for stemming algorithms,” <http://snowball.tartarus.org/texts/introduction.html>.
- Rennie, J. D. and Nati, N. S. (2005), “Fast Maximum Margin Matrix Factorization for Collaborative Filtering,” in *Proceedings of the International Conference on Machine Learning*, vol. 22.
- Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (eds.) (2011), *Recommender Systems Handbook*, Springer.
- Robertson, S. E., Walker, S., and Beaulieu, M. H. (2000), “Experimentation as a way of life: Okapi at TREC,” *Information Processing & Management*, 36, 95–108.
- Salakhutdinov, R. and Mnih, A. (2008a), “Probabilistic Matrix Factorization,” in *Advances in Neural Information Processing Systems*, vol. 20.

- Salakhutdinov, R. and Mnih, A. (2008b), “Bayesian Probabilistic Matrix Factorization using Markov chain Monte Carlo,” in *Proceedings of the International Conference on Machine Learning*, vol. 25.
- Shan, H. and Banerjee, A. (2010), “Generalized Probabilistic Matrix Factorizations for Collaborative Filtering,” in *Proceedings of the IEEE International Conference on Data Mining*.
- Soriano, J., Au, T., and Banks, D. (2013), “Text Mining in Computational Advertising,” *Statistical Analysis and Data Mining*, 6, 273–285.
- Su, C. and Judd, K. L. (2012), “Constrained Optimization Approaches to Estimation of Structural Models,” *Econometrica*, 80, 2213–2230.
- Ungar, L. and Foster, D. (1998), “Clustering Methods For Collaborative Filtering,” in *Proceedings of the Workshop on Recommendation Systems*, AAAI Press, Menlo Park California.
- Vickrey, W. (1961), “Counterspeculation, Auctions and Competitive Sealed Tenders,” *Journal of Finance*, 16, 8–37.
- Wikipedia (2014), “Shooting of Trayvon Martin — Wikipedia, The Free Encyclopedia,” http://en.wikipedia.org/wiki/Shooting_of_Travon_Martin, [Online; accessed January 18, 2014].
- Winoto, P. and Tang, T. Y. (2008), “If You Like the Devil Wears Prada the Book, Will You also Enjoy the Devil Wears Prada the Movie? A Study of Cross-Domain Recommendations,” *New Generation Computing*, 26, 209–225.
- Xiong, L., Chen, X., Huang, T.-K., Schneider, J., and Carbonell, J. G. (2010), “Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization,” in *Proceedings of the SIAM International Conference on Data Mining*.
- Zhang, Y., Cao, B., and Yeung, D. (2010), “Multi-Domain Collaborative Filtering,” in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, vol. 26.

Biography

Timothy Chun-Wai Au was born in Summit, New Jersey on December 17, 1987. In August 2006, he enrolled at Cornell University in Ithaca, NY where he double majored in Mathematics and Economics. After receiving a B.A. with Distinction in January 2010, he continued his studies as a graduate student in the Department of Statistical Science at Duke University in Durham, North Carolina beginning in August 2010. He received his M.S. in Statistics in May 2013, and his Ph.D. in Statistics in May 2014.