O Who is Watching You Eat?: A Noir Preferences Thriller

Judy Goldsmith (University of Kentucky; goldsmit@cs.uky.edu)
Nicholas Mattei (NICTA and UNSW; nicholas.mattei@nicta.com.au)
Robert H. Sloan (University of Illinois at Chicago; sloan@uic.edu)
DOI: 10.1145/2757001.2757004

Introduction

> "By the time a guest walks through the front doors at Ping Pong Dim Sum in Washington D.C., marketing manager Myca Ferrer can already be fairly certain what he or she will order. Ferrer isn't psychic, but he is using a guest intelligence platform called Venga to gain a deeper understanding of his most frequent customers" (Miles, 2013).

Many software packages are available to restauranteurs today, including Venga, OpenTable, BuzzTable, NoshList, and FiveStars. Each of these business solutions offer a variety of services, which may include marketing, table booking, wait-list management apps, social network integration, or restaurant customizable diner profiles, which should be composed of preferences. Food preferences involving an overall meal often serve as canonical examples in the preference handling community (Boutilier et al., 2004; Kaci, 2011). These examples are often quite simple, but there is a more interesting question: can we capture the preference knowledge that an expert server might have about a regular customer—he may order one of these three things, and would be interested in the special of the night if it is tripe or liver but not if it is sweetbreads (Mariani, 2011)—and can we do this within the field of preference handling?

The preference handling research community organized as the Advances in Preference Handling Multidisciplinary Working Group[1], starting with a Dagstuhl seminar in 2004. It includes researchers from operations research, economics, artificial intelligence, database systems, and other fields. The working group runs the annual MPREF Workshop. In 2015, MPREF will take place in conjunction with IJCAI 2015.

We have learned what we could about the restaurant software discussed by Miles (2013) as well as other similar systems. Many of these systems have reservation management and table-waiting lists as their primary function and are not directly connected to the Point of Sale (POS, i.e., the cash register including its software system), and therefore cannot automatically record what a diner ordered. Of the six systems Miles discusses, only FiveStars and Venga are connected to the POS. FiveStars can track the total dollars spent by a customer (for loyalty programs), but does not track specific items ordered. All other systems we looked at, such as Europe's Livebookings, are also focused primarily on reservations and marketing, rather than gathering customer intelligence.

The preferences that are collected by software are gathered via explicit elicitation, and they tend to be for such things as table preferences. So far, the state of the art is for more complex preferences to be entered manually as free text by managers, and those preferences are primarily what one would consider archetypes—WW for wine whale, one who spends lots on expensive wine, or HSM for heavy set man, needs bigger chairs (Craig, 2012). Craig (2012) goes on to note that, "Managers are usually the ones to enter notes, and they concede that too much information can be a problem." We speculate that the software companies see reservations and wait-list management as the single best opportunity to make a profit. For instance, Venga received a $1 million round of Series A venture capital funding in 2014.

We view these software systems as an interesting challenge and motivation for researchers in the preference handling community. We offer a piece of original fiction (typeset in *italics*) to illustrate the future possibilities coupled with a broad

[1] http://preferencehandling.free.fr

overview of the field of preference handling. This cautionary tale demonstrates the extremely grave danger to individual safety and to society of allowing so much as our preferences for butter versus olive oil to be stored online.

Readers looking for conventional surveys of the preferences literature may want to consult any of a number of excellent and more conventional works. Kaci (2011) provides an introduction and comprehensive overview of preferences in multiple domains including logics, decision making, constraint programming, and planning. Domshlak et al. (2011) cover preferences broadly in AI, while Rossi et al. (2011) describe how preferences link many fields in computer science and artificial intelligence including constraint reasoning, multi-agent systems, and computational social choice. Indeed, preferences play a key role in multi-agent systems, where it is often necessary to make decisions based on the opinions of multiple agents (Conitzer, 2010). Brafman and Domshlak (2009) provide an excellent and comprehensive introduction to the notion of computational preferences and focus on computational and modeling issues. Many fields employ preferences in their reasoning and decision making. Öztürk et al. (2005) provide an extensive overview of preference modeling from the point of view of decision analysis, in the spirit of management information science. Goldsmith and Junker, in their introduction to the special issue of AI Magazine on preference handling for artificial intelligence (Goldsmith & Junker, 2009), provide applications of preference handling and a set of research topics in the area. The database community has done considerable work on preference handling. Stefanidis et al. (2011) survey both the uses of preferences in database query handling, and preference representation, expressivity, and handling. Interestingly, the very extensive literature covered by that survey is largely disjoint from that covered by the other surveys here, and there are surely research opportunities available in the synthesis of these literatures. As preference handling has moved towards application, extensive repositories of preference data have been put online including the UCI Machine Learning Database (Bache & Lichman, 2013) and PrefLib (Mattei & Walsh, 2013).

A Noir Preferences Thriller

There are two things you should know about me: I like to eat, and I'm a contract killer. That's not my day job, of course—gotta keep the tax man happy. I am the sole proprietor of Safe Kitchens Ltd. I'm in the business of restaurant security. I vet suppliers, check software, change door locks, and watch the kitchen and wait staff at work.

Imagine a restaurant that can compute your preferred meals, based on your order history, elicited preferences, or wait-staff observations. You are seated and your waiter says, "Hello, Dr. Smith, and welcome. Would you like a Manhattan cocktail? We suggest you might be interested in the duck à l'orange or the rabbit stew tonight, but here's the full menu." You are delighted to accept the Manhattan; after a careful browse of the menu, you agree that the duck is exactly what you prefer. What sort of internal representation would the restaurant need to be using? How can they get it right all the time, for each customer?

I eat out a lot. I could write a restaurant review blog in my spare time, if I had any. I don't. I know how easy those review sites, TripAdvisor and Yelp and all, are to prejudice. Why add one honest voice in a sea of cousins, uncles, cozeners, and people with scores to settle?

While restaurants study and rate us, we are certainly returning the favor. Yelp, TripAdvisor, BeerAdvocate, Amazon and a number of other resources exist to provide data for and information about restaurants, beer, wine, DVDs, and every other object we may want to purchase (Mariani, 2011). The data from these sites is used by a number of research communities in the machine learning, data mining, preference learning, and recommendation systems fields; each of which has overlap with the preference handling community. Recommender systems of the sort Amazon and Netflix rely on their company's store of information, and on the sort of good but not perfect recommendation that can be obtained by looking at ratings across a large number of different individuals (Marlin & Zemel, 2009).

Within the field of machine learning, the preference learning community is the most closely related to the preference handling in AI community. It seeks to predict “complex objects such as ... [preference] orders, rather than single elements” (Fürnkranz & Hüllermeier, 2010). This contrasts with the score, distance, or latent feature based techniques found more commonly in machine learning, data mining, and recommendation systems tasks (Ricci et al., 2011). These communities employ a variety of sophisticated techniques, most famously variants of collaborative filtering such as Amazon (Linden et al., 2003) or Netflix (Bennett & Lanning, 2007; Koren et al., 2009) and other matrix factorization techniques. Some recommender systems predict how users will rate unseen objects from a sparse set of ratings (Ricci et al., 2011), aggregating the feelings of many users and matching the current user to ones like him or her. Often this is done without an explicit notion of the “preference” of an individual user. Other techniques often used for recommendations include learning (and suggesting) items or patterns that frequently occur together or in sequence (Agrawal & Srikant, 1994; Han et al., 2006).

Understanding how to interpret the signals from users and communicate recommendations back are important areas for future research in all the communities that use restaurants as motivation. In machine learning, McAuley and Leskovec (2013) use latent factor analysis automated interpretation of free text to map scores to keywords, generating explicit preferences. An intriguing question posed in the RecSys community is the effectiveness of different kinds of arguments for a recommendation (Sharma & Cosley, 2013). Say we have decided that a person would like the fish. There are many ways we can present (argue for) this recommendation. It turns out that, in many cases, independent of the objective quality of a recommendation, social recommendations are more often implemented (“Try the fish; six of your friends liked it”). The effectiveness of social recommendations has been noticed: most of the commercial software packages including BuzzTable and Venga support restaurant recommendations by making it easy for customers to push status updates to their social media accounts.

The current contract is a puzzle. I only take on targets that eat out. This guy, let’s call him “Frank,” eats out 3–5 nights a week. That we know of. The good restaurants, the ones that use the software packages I’ve worked with, they have biometrics. Put one hand on a table and they know all about you. What you ordered every time you were in there; who you ate with; how long you had to wait, how nice you were to the server (and I don’t just mean the size of your tips); what social media you use, and how many friends and followers you have; when you were born, and a probabilistic spread on when and of what you will die. It’s all right there in the software, especially if the restaurant buys several different packages.

As almost all of the software websites remind us, restaurant preferences include non-food features such as table location, special occasion status, size of party, etc. Consequently, we are interested in succinct, feature-based preference representations, rather than explicit listings of all possible menu offerings and exogenous factors. Currently, software packages like OpenTable allow restaurant staff to enter free text about regular clients; the staff must draw their own conclusions from that text, without so much as keyword searches.

And I can access all of that, from all those restaurants. Every time a company hires me to review the security features of their software, I leave myself a trap door. Just as I’m in and out of loading docks and kitchen doors, sharing a smoking break with the assistant cooks, I’m in and out of databases. I guess you could call me a backdoor man. I know more than any one restaurant, because I have access to all *the data. I download it in the midst of the dinner rush, when everything runs slower anyway, and run my diagnostics.*

Broadly speaking, the AI preference handling community’s focus on qualitative preference rep-

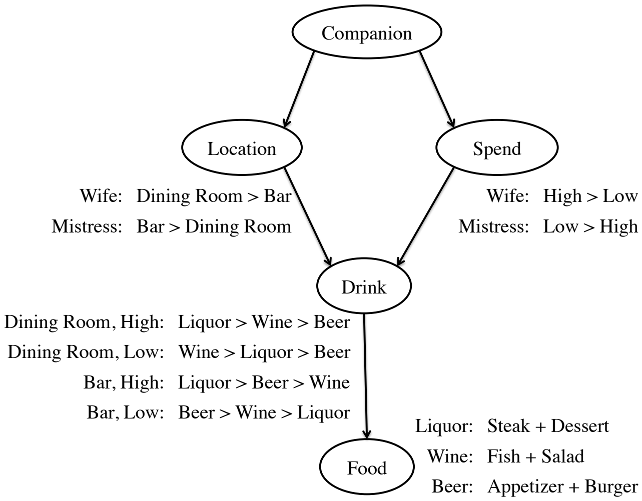

Figure 1. A CP-net for a simplified dining example inspired by wife/mistress example (Mariani, 2011).

resentations falls into two categories: graphical models, and logic-programming based representations. Turning first to graphical models, the example in Mariani (2011) shows a client whose preferences about eating in the dining room or bar depends on whether he is with his wife or mistress. CP-nets (Boutilier et al., 2004), as shown in Figure 1 provide a data structure for representing such conditional preferences.

A CP-net consists of a number of edges and nodes with edges denoting dependencies between the nodes. Each node represents a variable that can take an assignment from an independent domain. For each node we are given a set of conditional preference (or cp-) statements which, for each set of possible assignments to the variables of the parent nodes, provides a strict ordering over all possible elements of the domain. In Figure 1 the companion node has no incoming edges and hence, no dependencies. This means that the presence of the customers mistress or wife does not depend on his choice of location. However, since there is an edge from companion to spend, there is a different ordering over the assignments to the spend variable based on the presence of the wife or the mistress. We can read off the most preferred elements of the dependent nodes given assignments to the independent nodes to learn that, if the mistress is present, then the customer prefers to dine in the bar, not spend very much money, have a beer and hence an appetizer and a burger. Any other assignment to the variables results in a less preferred situation.

My last case was a lady always ordered elderflower wine from her local bistro. They kept a bottle just for her, imported from England. I had to buy a bottle for myself, on my last junket. Think a good long time about what slow-acting poison would be disguised by the taste. Elderflower wine is a subtle flavor, so I needed something tasteless, something that worked by accumulation. You can't have the mark just keel over in the restaurant. It would be bad for business—theirs and mine. My restaurants need sterling safety records.

It's a lonely business. I find stuff that the restaurants want to know, but I can't tell them I've been analyzing their data. That's not in my contract. For instance, and this is just a trivial example: I noticed one old guy, marked "b.t." for "bad tipper," always tipped well—unless he had dessert. All the waiter needed to know was to discourage him from ordering one more course. Kind of counter-intuitive, if you think your tip will be a percentage of the bill. It didn't really matter, since I managed to drop something into his floating island dessert one night. Sugar masks all sorts of things.

I did, however, suggest some simple machine learning tools to one of the software manufacturers. I hear they're getting better reviews in the trade journals. It's all good for business: theirs, mine, and the restaurants'. Some funny things came out of it, like a correlation — for most people! — between noise levels and spend levels. For most people, relative quiet lets them linger. That could be the kind of music that's playing — everyone has their aural sweet spot — or it could be where they're sitting. I like the marks who prefer a table tucked into a niche. Less conversational spill-over for them, and for me, well, I can wander by, add something to their food. Just have to be aware of the security cameras.

So that's how I work. The elderflower wine case was a rarity, a special bottle just for her. But you never know. When the obit hit, I realized there was a bottle of elderflower wine with enough poison to kill whoever drank the rest of it. I men-

tioned to the barkeep that I recognized her from the picture in the paper as someone I'd seen there. There was a record of my eating there one night when she was in, so that was legit. They told me about the wine, and I bought the rest of the bottle off of them.

Mostly, the reservations systems flag the mark, tell me where he'll be when. I know where he sits, what he likes, and I know what he's ordered so far. I have a good guess what he'll be ordering, long before he arrives at the restaurant, so I know which chemicals will be disguised by the flavors. Even if the mark likes variety, he's going to have trends, favorites, or he'll have patterns I can exploit. One I took out had a strict rotation of fish, chicken, veal. That was easy. Most aren't so determined. Sometimes I show up with several small vials or powders, ready to go.

CP-nets are appealing as they are a compact human-readable representation with considerable expressive power and have a more intuitive appeal than numerical rankings. However, they cannot handle preferences that are not fixed. One model of preference variance is that each of us has a set of rational preferences, and we choose amongst them, perhaps probabilistically (Regenwetter & Popova, 2011). Another is that each choice we make is probabilistic. We can model the latter with a probabilistic conditional preference network (PCP-net) (Cornelio et al., 2013; Bigot et al., 2013). In a PCP-net, we give probabilities over lines in the conditional preference table: consider the man whose preferences are described in Figure 1 and consider the condition that he's drinking beer. His conditional probability table could be the one shown in Table 1. We can thus conclude, among other things, that the probability distribution on the man's top choice of food (given that he is drinking beer) is: Burger 60%, Steak 30%, Fish 10%.

What about this Frank? It seemed like he threw darts at a list of restaurants. Didn't make reservations in his own name, just grabbed a name out of the phone book, created an email account, booked a table. Sometimes for one, sometimes for a few. Showed up and apologized, his lady stood him up. Once in a while, he had friends with him, but never the same ones twice. Sometimes he just walked in, got squeezed in at the bar or the table back by the kitchen.

Table 1. A conditional probability table (CPT) for one node of a larger PCP-net.

Beer	Burger $\succ$ Steak $\succ$ Fish	0.5
Beer	Burger $\succ$ Fish $\succ$ Steak	0.1
Beer	Steak $\succ$ Burger $\succ$ Fish	0.2
Beer	Steak $\succ$ Fish $\succ$ Burger	0.1
Beer	Fish $\succ$ Burger $\succ$ Steak	0.1
Beer	Fish $\succ$ Steak $\succ$ Burger	0.0

Most people have patterns in what they order, like I said. They drink white wine with fish, or they always have the red. Fish or something light means chocolate for dessert. Pasta is followed by fruit and nuts. Some have favorite dishes or favorite restaurants. Some dine promptly at 6:15 and heaven help a waiter or cook who's slow. Some dine fashionably late, at 9, and start slowly, with cocktails and little nibbles. When I'm following one of those, I am glad that I don't need to be up and at a desk by 9 the next morning!

CP-nets which require cp-statements to encode complete and strict relations always have a unique most-preferred outcome, at least when all items are on the menu. Turning to logic-programming representations, Weighted logic representations allow for explicit ties. For example, in penalty logic (De Saint-Cyr et al., 1994) the representation consists of a set of propositional logic formulas, each with its own weight (penalty). The penalty for an outcome is the sum of the penalties of all the formulas it violates; outcome with smaller penalties are preferred to outcomes with larger penalties. Consider the following penalty logic set:

$$\{(cocktails \wedge (redWine \vee whiteWine), 10), \\ (fish \vee meat, 4), (\neg meat \vee redWine, 6)\}$$

The most preferred meals are any that include cocktails, wine, and either meat or fish, with the additional restriction that if there is meat then the wine must be red. All of those meals pay zero penalty. The second-most preferred meals are

Generator (hard constraints)
1{nibbles, salad, soup}1 %First Course
1{fish, pasta, meat}1 %Main Course
0{white wine, red wine, beer, cocktails}3 %Drink
0{chocolate, crème brûlée, fruit+nuts}1 %Dessert
1{early, late}1 %Dinner Time
1{yes, no}1 %Whether works starts early tomorrow

Preferences (soft constraints)
white wine $\succ$ not white wine :- fish.
red wine $\succ$ not red wine :- not fish.
chocolate $\succ$ not chocolate :- fish.
fruit+nuts $\succ$ not fruit+nuts :- pasta.
cocktails $\succ$ wine :- late.
early $\succ$ late :- yes.

Figure 2: Answer set program for a 3-course dinner.

those with cocktails, wine, and pasta, and they pay a penalty of 4.

Another approach is to rank the importance of logical formulas, and to consider the rank of the most important formula that is violated, as is done in *possibilistic logic* (Dubois et al., 1991). Two other logics, *leximin* and *discrimin* (Benferhat et al., 1993) leverage the numbers or sets of violated formulae at each importance level, to compare preferability. In each case, some sort of logic programming engine, such as an answer set program, is needed for preference reasoning. Consider the answer set program shown in Figure 2 (from Zhu and Truszczynski (2013)), where "1{·}1" means that exactly one element in the set is true. Figure 2 begins with hard constraints: the meal consists of exactly one each of first course, main course, and dinner time, and one answer to the time work starts the next day. In addition, the diner can have no more than three types of drink, and at most one dessert. Next, there are condition preferences: if this diner has fish as a main course, she would rather have a chocolate dessert than a non-chocolate dessert. If dinner begins late in the evening, she prefers cocktails to wine. An answer set solver returns the set of stable models for the given answer set program; answer set optimizers return a set of optimal (with respect to soft constraints) stable models (Zhu & Truszcznski, 2013).

Frank was a difficult case. One night, it was steak and all the trimmings, chocolate cake, and port wine. Another time it was red wine with fish, and the cheese plate. Once it was just lamb chops, another time it was the deep-fried appetizer plate, a salad, and then a hamburger. The waitress would have been scratching her head if the health inspector wasn't dogging her footsteps. We don't need hair oil and dandruff on the plates at a high-end Asian fusion place. I can tell you, too, that's not the right place to order a hamburger, though they plate it up nice.

Consider a customer who patronizes nearby restaurants for lunch each workday. The customer may prefer not to eat at any restaurant on two consecutive days, or not to eat pizza more than once in a given week, or to eat seafood at least once a week. Such preferences involve *recency*, the desire to repeat, or not, recently chosen alternatives, and *frequency*, how often something is preferred. Some of us prefer the foods of our childhood, others look for new tastes. Such preferences involve the desire for *familiarity*, or *novelty* respectively. In addition to *temporal preferences* such as these, we also expect that the preferences of such customers will change over time. A patron may tire of salmon and begin to order beef instead.

There are many approaches to preference change and variability in the AI literature and related fields. One could start with the reason for changes. Lang and van der Torre (2008) consider preferences that change in reaction to the acquisition of new knowledge: I have learned that the chef at this restaurant has become too liberal with the peppercorns for my taste. An alternative approach is to hypothesize that we have a set of rational preferences that we switch between, perhaps because of hidden variables (we might not model the weather in our restaurant preferences, but it likely affects whether we want hot soup) or context (Konigsberg & Asherov, 2014). Without having access to all context variables, we might model such switching as choosing at random with a fixed distribution from a set of preferences (Regenwetter & Davis-Stober, 2012), or we might choose another form of uncertainty, such as probabilistic or fuzzy logic (Xu, 2007).

The preferences community has been slow to consider preferences that depend on the temporal history. Any successful preference-driven restaurant software will need to capture biases for novelty, “Ooh, pig face! I’ve never had pig face!”, and variety, “I haven’t had rabbit since last summer!”

I like to drop in on the restaurant, to be in the kitchen when the mark’s last course is plated. Or to brush by the table and drop something into the olive oil if I know they’ll eat the bread, and prefer olive oil to butter. Once, I used an aerosol on the back of someone’s neck while I sneezed, right behind them. Forcing a sneeze is a painful thing, and I probably won’t do that again.

But I have a day job, so to speak. Most nights, I’m dropping in on some restaurant, often on a schedule to catch a particular waiter, bad-tempered patron, supplier, or assistant cook doing something they oughtn’t. I promise you that there’s a lot less death and illness in my restaurants, despite my occasional marks. As I said, I use slow-acting stuff. The only mark I’ve seen die was a middle-aged woman dropped into a diabetic coma when we sent her a birthday surprise dessert. No one knew she was diabetic, been avoiding docs for years. She dropped out of that coma a couple minutes before the ambulance arrived. And for once, there was no doctor in the house. The only waitstaff with CPR training were out that night.

Would any of the current software systems have been useful to the killer? Venga’s tracking of POS data would have told him that she often ordered dessert. A stronger connection to Facebook might have shown him her birthday, that she was lonely with very few friends, and would eat a gift dessert from a stranger.

I was lucky that time. Someone could have scooped up her crème brûlée and analyzed the crust. But her daughter had suspected the diabetes for years, had tried to talk her out of the ice cream on top, at least. They tested her blood sugar, and that was all. The daughter didn’t want her momma cut up for autopsy.

So I’d get a notification that Frank was in one of my restaurants, and I’d be in the midst of interviewing a cook about his hand-washing routine. Or I’d be running diagnostics on software, and be unwilling to leave the premises while my machine was online and connected to their servers. Software security isn’t just about the software and the communications protocols. It’s about not letting people walk off with the physical servers, or sit in front of a display and write down what scrolls past. I don’t trust the locks on my door any more than I trust the encryption pack- ages. It’s all a game of discouraging the would-be thief.

Finally, I had had enough of Frank’s unpredictability. I cleared my calendar and sat down to wait for notification. The man had to eat, and it didn’t look like he was going to live on microwave burritos and pizza slices to go.

Sure enough, he showed up that night at one of my restaurants. I packed my gloves, different poison in a tiny bag at each finger tip, and set off. I should have known that something was funny when he made the reservation in his own name, same place he’d been eating every Thursday for weeks. I just thought he liked their rack of lamb. Never occurred to me I might be the mark.

And that’s how they caught me.

Epilogue

As mentioned in the introduction, many of today’s restaurant software systems are not directly connected to the POS, and therefore cannot automatically record what a diner ordered. Typically, preferences can be elicited and entered in text boxes, but preferences actually entered tend to be quite simple.

However, the restaurant scenario presents many interesting challenges to the preference community. There is the meta-preference question of whether a patron actually appreciates being known and having their preferences known. A reasonably alert waiter might detect a level of discomfort if a single, or very small set of, choices is offered; the waiter could even ask “Would Mr. Jones prefer that we present the complete list of specials, including those dishes to which he might be allergic?” Such information should definitely be recorded for Mr. Jones' next visit.

Based on a number of articles about the the restaurant industry and on anecdotal surveys, that many restaurant-goers do appreciate personalized presentations and service. This article attempts to highlight some of the opportunities to tune the restaurant experience to the individual's preferences; we leave to later authors in the multi-agent community the problem of tuning an experience to preferences of individuals considered as part of a group.

Developing the preferences models and algorithms that would form the heart of software for highly personalized service at high-end restaurants poses many challenging problems for the preferences research community. We see the following 8 key research and implementation challenges for the community:

1. recognize individuals' hard constraints (no pork, only Maker's Mark bourbon, seating only with back to the wall);
2. recognize correlations (iced tea → no dessert; dessert → no tip; beef if and only if red wine);
3. define all the observable factors (food, drink, noise level, music style, table placement, size of party, special occasion);
4. recognize dynamic patterns, such as novelty or variety biases;
5. develop preference learning algorithms that can leverage small data sets and sparsely elicited information;
6. define suitable dynamic preference models that balance expressivity and computational complexity;
7. develop algorithms to find most preferred, relatively preferable, and required items;
8. develop software and interfaces that make preference handling useful and usable by wait staff, maître d's, and chefs.

Knowledge acquisition and usability are some of the most important and thorniest issues in this domain. While every visit is an opportunity for more data, and perhaps a few more queries from the waiter, there are enormous cold-start challenges. Consider the idea of using the available ingredients, the style of the chef, and the preferences of the regular patrons with reservations to decide on a set of targeted (and presumably expensive) specials for the evening. To a researcher in computational preferences, this sounds appealing. However, if Fully Committed (Mode, 2001) or Kitchen Confidential (Bourdain, 2007) are to be believed, most chefs would despise having a computer tell them what to cook, how to do their job. Similarly, an excellent waiter already tunes her presentation to the patron, and might resent a computer-produced script. However, restaurant staff already use open text boxes to record notes on their regulars, and might appreciate an algorithm that organizes those notes and presents the relevant information as needed.

As we've indicated in this brief survey, the restaurant industry has some preference software in place, but does not at all leverage the power of preference reasoning. On the other hand, we have also indicated ways in which the current state of preference reasoning is not yet sufficient to handle the full range of personal preferences about food and the restaurant experience. The restaurant software companies have started with the low-hanging fruit, namely, improving scheduling and constraint solving for reservations and wait lists. Since only a fraction of their customers are "regulars," they have not yet turned their focus to personalization menus or presentations by waiters. All this preference reasoning can be used to target the restaurant goer's experience, for good or for ill.

Bon appétit!

Acknowledgments

This work is partially supported by the National Science Foundation, under grants CCF-1215985 and CCF-0916708. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. We thank Thomas E. Allen, Sam Saarinen, and Matthew Spradling for discussions of temporal preferences, and Ying Zhu for supplying the answer set program.

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proc. of the 20th International Conference on Very Large Databases*, pp. 487–499.

Bache, K., & Lichman, M. (2013). *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml. (University of California, Irvine, School of Information and Computer Sciences.)

Benferhat, S., Cayrol, C., Dubois, D., Lang, J., & Prade, H. (1993). Inconsistency management and prioritized syntax-based entailment. In *Proc. of the 13th International Joint Conference on Artificial Intelligence,* pp. 640–645.

Bennett, J., & Lanning, S. (2007). The Netflix prize. In *Proc. of the Knowledge Discovery and Data Mining KDD Cup Workshop.*

Bigot, D., Zanuttini, B., Fargier, H., & Mengin, J. (2013). Probabilistic conditional preference networks. In *Proc. of the 29th Annual Conference on Uncertainty in Artificial Intelligence*.

Bourdain, A. (2007). *Kitchen confidential updated edition: Adventures in the culinary underbelly*. Ecco.

Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., & Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional *ceteris paribus* preference statements. *Journal of Artificial Intelligence Research*, 21, 135–191.

Brafman, R. I., & Domshlak, C. (2009). Preference handling—an introductory tutorial. *AI Magazine*, 30(1), 58.

Conitzer, V. (2010). Making decisions based on the preferences of multiple agents. C*ommunications of the ACM*, 53(3), 84–94.

Cornelio, C., Goldsmith, J., Mattei, N., Rossi, F., & Venable, K. B. (2013). Updates and uncertainty in CP-nets. In *Proc. of the 26th Australasian Joint Conference on Artificial Intelligence*, pp. 301–312.

Craig, S. (2012). What restaurants know (about you). *New York Times*. http://www.nytimes.com/2012/09/05/dining/what-restaurants-know-about-you.html

De Saint-Cyr, F. D., Lang, J., & Schiex, T. (1994). Penalty logic and its link with Dempster-Shafer theory. In *Proc. of the 10th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 204–211.

Domshlak, C., Hüllermeier, E., Kaci, S., & Prade, H. (2011). Preferences in AI: An overview. *Artificial Intelligence*, 175(7), 1037–1052.

Dubois, D., Lang, J., & Prade, H. (1991). A brief overview of possibilistic logic. In *Symbolic and quantitative approaches to uncertainty* (pp. 53–57). Springer.

Fürnkranz, J., & Hüllermeier, E. (2010). *Preference learning*. Springer.

Goldsmith, J., & Junker, U. (2009). Preference handling for artificial intelligence. *AI Magazine*, 29(4), 9–12.

Han, J., Kamber, M., & Pei, J. (2006). *Data mining: Concepts and techniques*. Morgan Kaufmann.

Kaci, S. (2011). *Working with preferences: Less is more*. Springer.

Konigsberg, A., & Asherov, R. (2014). A recommender system sensitive to intransitive choice and preference reversals. In *Proceedings of the Fourth International conference on Computer Science & Information Technology*, pp. 409–419.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30–37.

Lang, J., & van der Torre, L. (2008). From belief change to preference change. In *Proc. of the 18th European Conference on Artificial Intelligence*, pp. 351–355.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. I*EEE Internet Computing*, 7(1), 76–80.

Mariani, J. (2011). What your favorite restaurants know about you. *Esquire Eat Like A Man Blog*. http://www.esquire.com/blogs/food-for-men/great-restaurant-service-110811

Marlin, B. M., & Zemel, R. S. (2009). Collaborative prediction and ranking with non-random missing data. In *Proc. of the 3rd ACM Conference on Recommender Systems*, pp. 5–12.

Mattei, N., & Walsh, T. (2013). Preflib: A library for preferences, http://www.preflib.org. In *Proc. of the 3rd International Conference on Algorithmic Decision Theory*, pp. 259–270.

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating di- mensions with review text. In *Proc. of the 7th ACM Conference on Recommender Systems*, pp. 165–172.

Miles, S. (2013). 6 tools restaurants can use for better guest intelligence. *Street Fight*. http://streetfightmag.com/2013/07/22/6-tools-restaurants-can-use-for-better-guest-intelligence/

Mode, B. (2001). *Fully committed*. Dramatist's Play Service.

Öztürk, M., Tsoukiàs, A., & Vincke, P. (2005). Preference modelling. In J. Figueira, S. Greco, & M. Ehrgott (Eds.), *Multiple criteria decision analysis: State of the art surveys* (pp. 27–71). Springer.

Regenwetter, M., & Davis-Stober, C. P. (2012). Behavioral variability of choices versus structural inconsistency of preferences. *Psychological Review*, 119(2), 408.
Regenwetter, M., & Popova, A. (2011). Consensus with oneself: Within-person choice aggregation in the laboratory. In *Consensual processes* (pp. 95–121). Springer.
Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (Eds.). (2011). *Recommender systems handbook*. Springer.
Rossi, F., Venable, K. B., & Walsh, T. (2011). A short introduction to preferences: Between artificial intelligence and social choice. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(4), 1–102.
Sharma, A., & Cosley, D. (2013). Do social explanations work? Studying and modeling the effects of social explanations in recommender systems. In *Proc. of the 22nd International World Wide Web Conference,* pp. 1133–1144.
Stefanidis, K., Koutrika, G., & Pitoura, E. (2011). A survey on representation, composition and application of preferences in database systems. *ACM Transactions on Database Systems*, 36(3), 19.
Xu, Z. (2007). A survey of preference relations. *International Journal of General Systems*, 36(2), 179–203.
Zhu, Y., & Truszczynski, M. (2013). On optimal solutions of answer set optimization problems. In *Proc. of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning*, pp. 556–568. Springer.

Judy Goldsmith is a Professor of Computer Science at the University of Kentucky. Her research includes computational decision making (preferences, voting, planning under uncertainty), and using science fiction to teach AI and computer ethics. She, Nicholas Mattei, and Emanuelle Burton will edit a science fiction anthology for use in CS ethics class.

Nicholas Mattei is a research scientist in the Optimization Research Group at NICTA and a conjoint lecturer at the University of New South Wales in Sydney, Australia. His research focuses on theoretical, empirical, and behavioral aspects of social choice, preference aggregation, and optimization —- helping AI enable and augment decision making.

Robert Sloan is a Professor and Department Head of Computer Science at the University of Illinois at Chicago. His research is in two areas: problems near the boundary of theoretical computer science and artificial intelligence, and privacy and computer security public policy issues.