

Who is Watching You Eat?

Judy Goldsmith

University of Kentucky
Lexington, KY
goldsmi@cs.uky.edu

Nicholas Mattei

NICTA and UNSW
Sydney, Australia
nicholas.mattei@nicta.com.au

Robert H. Sloan

University of Illinois at Chicago
Chicago, IL
sloan@uic.edu

Abstract

Many of the seminal papers in preference handling have used food preferences as motivating examples for their work; for example (Boutilier et al. 2004; Chomicki 2002). As foodies, the authors find this particularly motivating. While we think that there is both research and commercial potential in preference-based software for restaurants, we believe that serious application of the MPREF community’s technology to the problem of personal preference-driven presentation of menus, seating, etc., will require significant further innovation. We broadly survey the current use of preferences in making the dining-out experience more enjoyable, and we look at the states of the art for preference representation and reasoning, and for restaurant software. We illustrate some of our points with a short story.

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).

Miles lists six software packages available to restaurants: Venga, BuzzTable, OpenTable, NoshList, FiveStars, and QuickCue¹. 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. We are interested in the diner profiles. In this paper we detail (our best guess from the marketing material and informal interviews) the kinds of information that restaurants are currently making use of through these and other similar software solutions. We use this as a motivating scenario to examine the kinds of work in the MPREF community that could be leveraged to deliver a better customer experience, and to speculate on the things on the horizon.

When researching the products for this article we were struck by the amount of data that these apps could automat-

ically call up once a simple phone number or Facebook profile name was handed over (a requirement for the waiting list at a restaurant using BuzzTable). And we offer a cautionary tale to illustrate the potential danger to society of putting merely our preferences for butter versus olive oil online.

Food preferences involving an overall meal (e.g., “if main=fish, then prefer white wine to red wine”) often serve as a canonical example in the preference handling community. For instance, such preferences are the running example in the seminal paper on CP-nets (Boutilier et al. 2004), and also the running example for logic-based preference representations in Kaci’s book on preferences (Kaci 2011). However, in these examples, the preferences are quite simple, and typically have a single most-preferred meal for any fixed menu. We see capturing 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)—to be an excellent challenge for the field.

We have learned what we could about the restaurant software discussed by Miles and a few other similar systems. Many of these systems have reservations and table-waiting lists as their primary target and are not directly connected to Point of Sale (POS, i.e., the cash register), and therefore *cannot* automatically record what a diner ordered. Of the six systems Miles discusses, only FiveStars and Venga are connected to 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.

Those preferences that are collected by software are gathered via explicit elicitation, and tend to be for such things as table preferences. So far, the state of the art is for more complex preferences to be entered by hand 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). We speculate that the software companies see reservations and wait-list management as the single best opportunity to make a profit. Today, many of these companies are still in the early stages of starting up,

¹QuickCue was acquired by OpenTable in December 2013.

with Venga having just received a \$1 million round of Series A VC funding in January 2014, although OpenTable dates back to 1998.

Machine Learning or Preference Handling?

Diners' restaurant meal preferences have some features that make them particularly interesting and challenging as a research topic. Recommender systems of the sort Amazon and Netflix have developed can 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 different individuals. Similarly, the online advertising ecosystem receives vast amounts of data such as linger time and clickthrough rate on an ad by ad basis, and needs only to predict average user behavior. As Marani points out, an outstanding restaurant knows *exactly* what *you* like based exclusively on your behavior (Mariani 2011).

Machine learning (ML) for recommender systems uses a variety of sophisticated techniques, most famously, variants of collaborative filtering 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. The fundamental idea is to predict objects that users will rate highly (Bennett and Lanning 2007). This type of recommendation is important, but good service goes beyond this—it anticipates needs, not suggests alternatives.

Other techniques in ML often used for recommendations include learning (and suggesting) items or patterns that frequently occur together or in sequence (Agrawal and Srikant 1994; Han, Kamber, and Pei 2006). Noting that beer sales pick up during happy hour or understanding frequent criticisms in customer reviews are important factors in understanding how customers behave in the aggregate, these techniques are used in a multitude of industries and targeted marketing campaigns throughout the world. While these techniques are useful in the restaurant setting for suggesting sides for everyone, they do not leverage the intimate personal knowledge of a great head-waiter.

The preference learning in ML community is the most closely related to MPREF. It seeks to predict “more complex objects such as weak or partial orders, rather than single values” (Fürnkranz and Hüllermeier 2010). This contrasts with the more score based objectives found in traditional machine learning and recommendation systems tasks (Ricci et al. 2011). We are not interested in big data problems—we want to personalize, not draw generalizations. Based on past action, we want to know what one person wants, not predict a suite of possible actions or relations.

Other work in ML of interest to the MPREF community is the marriage of latent factor analysis with automated interpretation of the free text that often accompanies numerical reviews on sites like TripAdvisor and BeerAdvocate (McAuley and Leskovec 2013). McAuley and Leskovec's method matches the overall score of an item with key words and phrases in the text. The techniques are able to provide better recommendations—the written reviews give justifica-

tion for the numerical scores, and thus can identify aspects or preferences of items that are important.

An intriguing question posed in the RecSys community, and necessary for implementing the suggestions of our reasoning system, is the *effectiveness* of different kinds of arguments for a recommendation (Sharma and 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: Several commercial software packages support restaurant recommendations by making it easy for customers to push “Like” to their Facebook accounts.

Now we turn to the subject of preferences and dining, and we illustrate some of our ideas with a story.

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 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. However, our occasional overall rating of a restaurant on Yelp or the like does not provide the sort of information about our meal preferences that we are discussing here. An interesting question from the RecSys and psychology literature is, “does showing a suggested score alter a user rating of an object?” (Cosley et al. 2003). The answer appears to be yes, and one thing that many restaurants may benefit from is aggressively suggesting positive reviews for their establishment. This brings up an interesting question: is the waiter attempting to influence our preferences, and hence our scores, by recommending items that are better (more profitable) for the house? Note that Yelp and TripAdvisor use voting schemes, and, as we know, voting schemes are subject to manipulation; some manipulators of such social choice sites are reasonably sophisticated in their manipulation (Garcin, Xia, and Faltings 2013).

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.

We are interested in succinct, feature-based preference representations, rather than explicit listings of all possible menu offerings,² because the number of possible meals is combinatorial in the number of standard offerings, or unbounded if the menu changes repeatedly. Thus, we ignore options such as a ranking of alternatives, or a pairwise representation (see PrefLib.org (Mattei and Walsh 2013)), explicitly showing preferences between each pair of meals. Broadly speaking, the MPREF community's focus on qualitative preference representations fall into two categories: graphical models, and logic-programming based representations. We begin by looking at the graphical models.

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.

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. Consider the example in Mariani (Mariani 2011), in which a client's 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) provide a data structure for representing such conditional preferences, and Li, et al., provides engines for reasoning about them (Li, Vo, and Kowalczyk 2011b; 2011a). A graphical example of a conditional preference structure is shown in in Figure 1.

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.

Conditional lexicographic preference trees form another

²Restaurant preferences, as the software websites remind us, can include non-food features such as table location, special occasion status, size of party, etc.

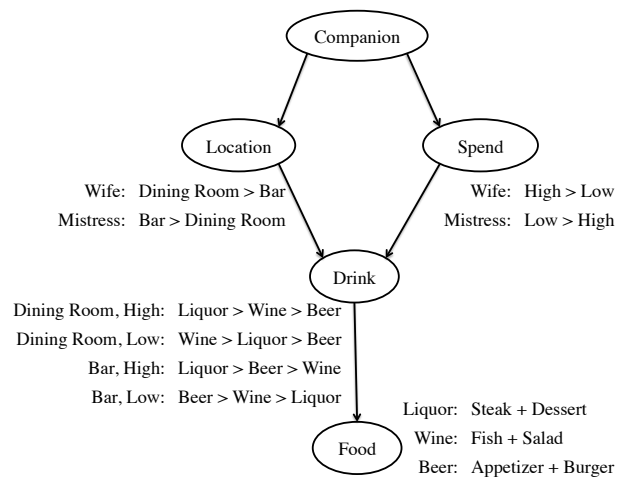


Figure 1: A CP-net for a simplified dining example inspired by wife/mistress example (Mariani 2011).

family of graphical model-based preferences (Booth et al. 2010). Unlike CP-nets, all models are tree-shaped, for easier reasoning, and variables may appear as multiple nodes in the tree, where the effect of drink choice on food choice depends on whether the drink choice is conditioned on the locale or on the decision to spend high or low.

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'.

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 mentioned 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. I wasn't going to take it home and dump it down my toilet, just in case someone had a warrant. Told them later it turned out it just wasn't to my taste. Took it to a city restroom, and in front of the cameras, took a swig. Made an awful series of faces as I worked my adams' apple, then dumped the rest of it down the toilet. Those toilets get flushed a lot.

Mostly, the reservations systems flag the mark, tell me where he'll be when. I know 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 because they are a human-readable representation with considerably more expressive power than straightforward feature-based preferences, and 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 and Popova 2011). Another is that each choice we make is probabilistic. We can model the latter with a probabilistic conditional preference network (PCP-net) (Bigot et al. 2013; Cornelio et al. 2013), and can consider the former as a collection of CP-nets—which can also be interpreted as a PCP-net (Cornelio et al. 2014). In a PCP-net, we give probabilities over lines in the conditional preference table, e.g., 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. (There's a 60% chance he'll order a burger.)

Table 1: An entry in a PCP-net CPT

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

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.

So far, we have discussed preference models based on individual preferences. However, many people dine with friends. In some cases, this has no effect on individual orders, but there are settings (such as pizza places or set menus for large groups) where preferences must be aggregated or choice sets must be minimized. There has been considerable work on aggregating CP-net preferences, including both theoretical analyses (Mattei et al. 2013) and actual implementations (Li, Vo, and Kowalczyk 2011c; 2011d), as well as for Conditional Lexicographic Preference Trees (Liu and Truszczyński 2013). The notion of budgeted social choice works to incorporate ideas of proportional representation with minimal elicitation requirements—attempting to find

a set of options that keep the most people happy (Lu and Boutilier 2011).

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 (in their plain form) always have a unique most-preferred outcome, at least when all items are on the menu. *Weighted logic* representations allow for explicit ties. For example, in *penalty logic* (De Saint-Cyr, Lang, and Schiex 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 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, Lang, and Prade 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 following answer set program (Zhu and Truszczyński 2013), where “ $1\{\cdot\}1$ ” means that exactly one element in the set is true.

Generator (hard constraints):

```
1{nibbles, salad, soup}1 %First Course
1{fish, pasta, meat}1 %Main Course
1{white wine, red wine, beer, cocktails}1 %Drink
1{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.
```

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 and Truszczyński 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 sure plated 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. 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.

The MPREF community has been slow to present time-aware representations. 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 than we believe is currently available might have shown him her birthday. He might have used her Facebook account to see that she was lonely, 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 packages. It's all a game of discouraging the would-be thief.

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 repeat 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.

Finally, I had had enough of Frank's unpredictability. I cleared my calendar and waited 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.

Conclusions

As a community, we have shown interest in food preferences. 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 so much to automating the profiling of regular customers and the personalization of menus or at least presentations of the menu by waiters and maître d's. We have also provided a fanciful example of how preference reasoning can be used to target the restaurant goer's experience.

Bon appétit!

Acknowledgments This work is partially supported by the National Science Foundation, under grants CCF-1215985 and CCF-0916708. We thank T.E. Allen, S. Saari-nen, and M. Spradling for discussions of temporal preferences. NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

References

- Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules. In *Proc. 20th International Conference on Very Large Databases (VLDB)*, volume 1215, 487–499.
- Benferhat, S.; Cayrol, C.; Dubois, D.; Lang, J.; and Prade, H. 1993. Inconsistency management and prioritized syntax-based entailment. In *Proc. IJCAI*, volume 93, 640–645.
- Bennett, J., and Lanning, S. 2007. The Netflix prize. In *Proc. KDD Cup and Workshop*.
- Bigot, D.; Zanuttini, B.; Fargier, H.; and Mengin, J. 2013. Probabilistic conditional preference networks. In *Proc. UAI*.
- Booth, R.; Chevaleyre, Y.; Lang, J.; Mengin, J.; and Som-batheera, C. 2010. Learning conditionally lexicographic preference relations. In *Proc. ECAI*, 269–274.
- Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A tool for representing and reasoning with conditional *ceteris paribus* preference statements. *J. Artif. Intell. Res. (JAIR)* 21:135–191.
- Chomicki, J. 2002. Querying with intrinsic preferences. In *Advances in Database Technology — EDBT 2002*. Springer. 34–51.
- Cornelio, C.; Goldsmith, J.; Mattei, N.; Rossi, F.; and Venable, K. B. 2013. Updates and uncertainty in CP-nets. In *AI 2013: Advances in Artificial Intelligence*. Springer. 301–312.
- Cornelio, C.; Goldsmith, J.; Grandi, U.; Mattei, N.; Rossi, F.; and Venable, K. B. 2014. Voting with CP-nets using a probabilistic preference structure. In *Proc. ComSoc*.
- Cosley, D.; Lam, S. K.; Albert, I.; Konstan, J. A.; and Riedl, J. 2003. Is seeing believing? How recommender system interfaces affect users’ opinions. In *Proc. SIGCHI Conference on Human Factors in Computing systems*, 585–592. ACM.
- 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.; and Schiex, T. 1994. Penalty logic and its link with Dempster-Shafer theory. In *Proc. UAI*, 204–211.
- Dubois, D.; Lang, J.; and Prade, H. 1991. A brief overview of possibilistic logic. In *Symbolic and Quantitative Approaches to Uncertainty*. Springer. 53–57.
- Fürnkranz, J., and Hüllermeier, E. 2010. *Preference Learning*. Springer.
- Garcin, F.; Xia, L.; and Faltings, B. 2013. How aggregators influence human rater behavior? In *Proc. Workshops @ 14th ACM Conference on Electronic Commerce*.
- Han, J.; Kamber, M.; and Pei, J. 2006. *Data Mining: Concepts and Techniques*. Morgan Kaufmann.
- Kaci, S. 2011. *Working with Preferences: Less Is More*. Springer.
- Li, M.; Vo, Q. B.; and Kowalczyk, R. 2011a. An efficient approach for ordering outcomes and making social choices with CP-nets. In *AI 2010: Advances in Artificial Intelligence*. Springer. 375–384.
- Li, M.; Vo, Q. B.; and Kowalczyk, R. 2011b. Efficient heuristic approach to dominance testing in CP-nets. In *Proc. AAMAS*, volume 1, 353–360.
- Li, M.; Vo, Q. B.; and Kowalczyk, R. 2011c. Efficient penalty scoring functions for group decision-making with TCP-nets. In *Proc. AAMAS*, volume 3, 1073–1074.
- Li, M.; Vo, Q. B.; and Kowalczyk, R. 2011d. Majority-rule-based preference aggregation on multi-attribute domains with CP-nets. In *Proc. AAMAS*, volume 2, 659–666.
- Liu, X., and Truszczynski, M. 2013. Aggregating conditionally lexicographic preferences using answer set programming solvers. In *Proc. ADT*. Springer. 244–258.
- Lu, T., and Boutilier, C. 2011. Budgeted social choice: From consensus to personalized decision making. In *Proc. IJCAI*, 280–286. AAAI Press.
- 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>.
- Mattei, N., and Walsh, T. 2013. Preflib: A library for preferences <http://www.preflib.org>. In *Proc. ADT*. Springer. 259–270.
- Mattei, N.; Pini, M. S.; Rossi, F.; and Venable, K. B. 2013. Bribery in voting with CP-nets. *Annals of Mathematics and Artificial Intelligence* 68(1-3):135–160.
- McAuley, J., and Leskovec, J. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proc. RecSys*, 165–172. ACM.
- 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/>.
- Regenwetter, M., and Popova, A. 2011. Consensus with oneself: Within-person choice aggregation in the laboratory. In *Consensual Processes*. Springer. 95–121.
- Ricci, F.; Rokach, L.; Shapira, B.; and Kantor, P. B., eds. 2011. *Recommender Systems Handbook*. Springer.
- Sharma, A., and Cosley, D. 2013. Do social explanations work? Studying and modeling the effects of social explanations in recommender systems. In *Proc. 22nd International World Wide Web Conference*, 1133–1144. International World Wide Web Conferences Steering Committee.
- Zhu, Y., and Truszczynski, M. 2013. On optimal solutions of answer set optimization problems. In *Logic Programming and Nonmonotonic Reasoning*. Springer. 556–568.