A report on C. Domshlak’s ”Unstructuring User Preferences” seminar

Miguel Ramirez Javega

June 22, 2006

Abstract

One of the hot topics on automatic decision making research are ”Recommender Systems”. These systems, given a description of user preferences about items of a certain kind, offer advice for making decisions regarding other items of that same kind. One of the most popular applications of these systems is in electronic commerce, where user preferences are the history of past purchases on an e-store.

1 Helping people to make decisions

One of the aspects of human cognition that has attracted more interest from the Artificial Intelligence research community is that of making decisions. When a human makes a decision two elements are taken into account. First, the knowledge the person has about the subject of the decision. And second, her ”preferences”. Preferences are not facts, but a certain predisposition to judge certain situations or objects in a concrete, sometimes very particular way. So preferences can be seens as ”rules of thumb”, which do not need to be necessarily self-consistent, nor have a simple structure or being easy to describe.

AI work regarding decision making can be classified according the following dimensions:

- Is the intelligent system supposed to substitute a human decision maker? Or just to support her?
- Do we care about a concrete individual preferences? Are we rather looking at some sort of idealized set of preferences, that we identify with ”rationality” or ”common sense”? Or we just care about the utility of the outcome of the decisions made?

Mr. Domshlak talk was about systems that *support* a *concrete* human decision maker. The task of these systems is to produce *suggestions* about action taking into account the preferences of the supported decision maker. These suggestions are helpful when the system is able to reduce the effort invested by the user in reviewing possible options. That essentially amounts to prioritize options for reviewing, so the user can make a decision faster.

This idea of *focusing user attention* is somewhat similar to that of *search engines*. In this setting though, the system can produce a ranking of options without the user having to make explicit his needs or preferences. While search engines focusing on attending short term needs of information, decision support systems focus is on attending *long term* needs. Decision support systems are harder endeavours than search engines, since besides needing an efficient procedure to compute rankings for options, it is also necessary to model those rules of thumb that constitute user preferences.

This ”ranking” function is usually implemented by a *linear utility function*:

$$U(\mathbf{x}) = \sum_i w_i x_i \tag{1}$$

where $\mathbf{x}$ is the representation of an option as a vector (set) of attributes. The task can be then posed formally as that of computing the ”optimal” set of coefficients $\mathbf{x}$. And optimality means that these coefficients guarantee the maximum degree of agreement between user expectations and system suggestions.

2 User preferences are not necessarily simple

User preference representation and choice ranking computation are tightly coupled: representation mostly determines how ranks are computed. This representation has some built-in assumptions about the nature of users' preference rules. Since each rule will be affecting the ranking of a choice, past research has mostly limited itself to modeling these rules as independent, since this guarantees that choice ranking will have the nice properties of additivity. Frequently this assumption is not sound as the attributes used to represent options are not independent themselves.

Models based on these premises haven't show particularly exciting performance. While the model is simple enough to ensure efficient computation, it usually turns out to be too simple to faithfully depict user preferences.

Mr. Domshlak's proposal goes one step further from these simple models. He proposes that:

- The system should be free of any explicit assumption on user preferences structure.
- The user should be able to provide arbitrary qualitative preference expressions, and the system should be able to extract a reasonable interpretation of these expressions.

3 Modeling objects

Before describing Mr. Domshlak we should introduce the concept of an item's *surrogate*. A surrogate is the representation, or model, of the real thing about users will be expressing preferences about. The surrogate is composed by a set of *attributes* which describe qualitatively those item features that are find to be relevant. For instance, the surrogate for a car in an on-line car store would be composed by the attributes: maker brand, color, gas consumption per mile (high, low), top speed, etc. The set of attributes a surrogate can have is the backbone of the language the user will be expressing his preference statements.

Why is that? Preference statements, in this setting, are partial orderings supplied by the user on some conjunction of attributes' values. For instance, she can say that she prefers sport fast cars over non-sport, low gas consuming cars. The preference criteria implicit in the previous statement should be also encoded according those attributes.

Let's consider that our items are described by just two binary (yes/no) attributes, X_1 and X_2, where x_1 denotes the "yes" value for attribute X_1. All the possible "words" of the language the user will be using are $x_1, \bar{x_1}, x_2, \bar{x_2}, x_1x_2, x_1\bar{x_2}, \bar{x_1}x_2, \bar{x_1}\bar{x_2}$. Note that is not required that preference statements refer explicitly to *all* attributes. The size of this language is exponential on the number of attributes, more precisely 2^{2^n}.

So every object in our world then can be projected to an algebraic space whose axis are precisely all the possible conjunctions of attribute values $x_1, \ldots, \bar{x_1}\bar{x_n}$. More formally we can define the following mapping $\Phi : \mathcal{X} \longmapsto \mathcal{F} = \mathbb{R}^{2^{2^n}}$:

$$\Phi(\mathbf{x})[i] = \begin{cases} 1 & val(f_i) \subseteq \mathbf{x} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $val(f_i)$ denotes the i-th element of the set:

$$x_1, \bar{x_1}, \ldots, x_n, \bar{x_n}, \ldots, x_1x_2 \cdots x_n, \ldots, \bar{x_1}\bar{x_2} \cdots \bar{x_n} \tag{3}$$

note that his space accounts for the relationships between different objects' surrogates – sharing an attribute value. In our 2 attribute objects world, the object $x_1, \bar{x_2}$ would be represented by the vector $\Phi(\mathbf{x}) = \{1, 0, 0, 1, 0, 1, 0, 0\}$.

This space has two major advantages with respect to previous approaches. First, axis are *meaningful*, in contrast with similar techniques like *Principal Component Analysis* or *Factor Analysis*. Second, a linear utility function definition comes forward:

$$\mathfrak{U}(\Phi(\mathbf{x})) = \sum_{i}^{2^{2^n}} w_i \Phi(\mathbf{x})[i] \tag{4}$$

Note that the linearity of the utility function, in this space, is guaranteed by construction.

4 Interpretation of preference statements

The question that is yet to be addressed is how one is supposed to compute the coefficients w_i. This computation implies that the system gives an interpretation to user's preference statements. Domshlak addresses interpretation from a non-monotonic logic perspective: to reason about preferred models.

Given an statement $\gamma > \psi$, that is that the objects implied by logical sentence γ are preferred over those implied by sentence ψ, we consider the constraints in the w coefficients associated. For instance, if we have the preference statement:

$$\gamma > \psi \tag{5}$$

$$X_1 \vee X_2 > \bar{X_3} \tag{6}$$

the models (the attribute values conjunction implying those sentences) of γ and ψ are:

$$Models(\gamma) = \{x_1x_2, x_1\bar{x_2}, \bar{x_1}x_2\} \tag{7}$$

and

$$Models(\psi) = \{\bar{x_3}\} \tag{8}$$

so the user is specifying implicitly the w_i coefficients, since she's saying that the sum of the utilities of the models of γ should be always greater than the sum of the utilities of the models ψ:

$$w_{x_1} + w_{x_2} + w_{x_1 x_2} > w_{\bar{x_3}} \tag{9}$$

$$w_{x_1} + w_{\bar{x_2}} + w_{x_1 \bar{x_2}} > w_{\bar{x_3}} \tag{10}$$

$$w_{\bar{x_1}} + w_{x_2} + w_{\bar{x_1} x_2} > w_{\bar{x_3}} \tag{11}$$

Formally, the process of deriving the set of constraints $\mathfrak{C}$ is as follows for a single preference statement is:

$$\mathfrak{U}(\Phi(x)) > \mathfrak{U}(\Phi(x)) \Leftrightarrow \sum_{i=1}^{2^{2^n}} w_i \Phi(x)[i] > \sum_{i=1}^{2^{2^n}} w_i \Phi(x')[i] \tag{12}$$

$$\Leftrightarrow \mathbf{w}\Phi(x) > \mathbf{w}\Phi(x') \tag{13}$$

For m preference statements we end up with the constraint set $\mathfrak{C}$:

$$\mathfrak{C} : \mathbf{w}\Phi(x_1) > \mathbf{w}\Phi(x'_1) \tag{14}$$

$$\dots \tag{15}$$

$$\mathbf{w}\Phi(x_m) > \mathbf{w}\Phi(x'_m) \tag{16}$$

Summing up, preference statements are used to build a set of linear equations (constraints) $\mathfrak{C}$, that constrain the parameters of our utility function $\mathfrak{U}(\Phi((x))$. Any solution to the set of constraints (equations) will yield a valid, that is consistent with the interpretation of user's statements, solution. Another boon of this interpretation is the fact that is *least commiting* - thus being able to handle inconsistent information provided by the user.

Although Mr. Domshlak's proposal is sound, is not very tractable. There's a combinatoric explosion:

- in the number of constraints is linear in 2^{2^n}
- in the summation in each constraint for an statement $\gamma > \psi$ is exponential in the number of attributes appearing on γ and ψ.

so the utility function will be usually unfeasible to compute (and store somewhere).

5 Computing the utility function

As mentioned above, solving the constraint set $\mathfrak{C}$ for an interesting number of attributes n will be unfeasible for nave methods. Mr. Domshlak's approach, based on duality techniques and Mercer kernels solves such a system in time linear on n, the number of attributes used to describe items, and polynomial on m, the number of statements supplied by the user.

The first step consists in posing the optimization problem as an strictly convex quadratic program:

$$\text{Minimize w.r.t. } \mathbf{x} : \frac{1}{2}\mathbf{w} \cdot \mathbf{w} \tag{17}$$

$$\text{subject to : } \mathbf{w}\Phi(x_1) > \mathbf{w}\Phi(x'_1) + 1 \tag{18}$$

$$\dots \tag{19}$$

$$\mathbf{w}\Phi(x_m) > \mathbf{w}\Phi(x'_m) + 1 \tag{20}$$

This optimization problem is moved into its Wolfe's dual, where it becomes obvious the necessity of finding an effective way to compute inner products in $\mathcal{F}$. In order to being able to use Mercer kernels, and compute efficiently the inner product, the mapping is slightly changed, as described in the following theorem.

Theorem(Domshlak & Joachims, 2005).

For the mapping $\Phi_\lambda : \mathcal{X} \mapsto \mathcal{F} = \mathbb{R}^{2^{2^n}}$

$$\Phi_\lambda(x)[i] = \begin{cases} \sqrt{c_\lambda(|val(f_i)|)}, & val(fi) \subseteq \mathbf{x} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

where

$$c_\lambda(k) = \sum_{l=k}^{n} \lambda_l \sum_{\substack{l_1>1,\cdots,l_k\geq 1 \\ l_1+\cdots+l_k=l}} \frac{l!}{l_1!\cdots l_m!} \tag{22}$$

and any pair $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ and $\lambda_1, \cdots, \lambda_n \geq 0$, the kernel

$$K(\mathbf{x}, \mathbf{x}') = \sum_{l=1}^{n} \lambda_l (\mathbf{x} \cdot \mathbf{x}')^l \tag{23}$$

computes the inner product $\Phi_\lambda(\mathbf{x}) \cdot \Phi_\lambda(\mathbf{x}') = K(\mathbf{x}, \mathbf{x}')$.

To compute the value of $\mathfrak{U}$ for a given alternative x'', that is, an object that does not appear on those statements supplied by the user, we need to know only the dual solution and the kernel:

$$\mathfrak{U}(\Phi(x'')) = \mathbf{x}^* \cdot \Phi_\lambda(x'') = \sum_{i=1}^{m} \alpha_i (K(x_i, x'') - K(x_i', x'') \tag{24}$$

So by using the "kernel trick", computing $\mathfrak{U}$ or solving $\mathfrak{C}$ does not require computations to be performed on the intractable $\mathbb{R}^{2^{2^n}}$ space.

6 Biographical information about C. Domshlak

Carmel Domshlak is a senior lecturer at Technion, the Israel Institute of Technology. He holds a PhD on Computer Science from Ben Gurion University (2002). Mr. Domshlak's work is centered on research about algorithms and models for automatic decision making and reasoning, both in sequential situated domains (planning) and graphical models.