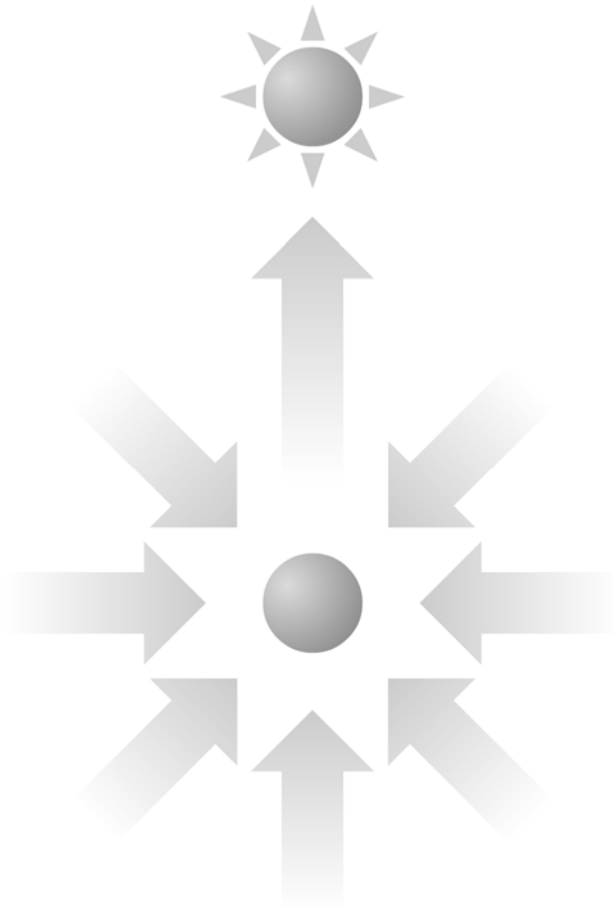


# **ATTRIBUTE PROFILING:**

## **A complete walk-through**



# TABLE OF CONTENTS

<b>Table of contents .....</b>	<b>2</b>
<b>Overview .....</b>	<b>3</b>
Terminology .....	3
<b>Attribute Profiling: A step by step walk-through .....</b>	<b>4</b>
Defining “Attribute” .....	4
The “what” and “why” of Attributes .....	4
Defining your Attributes .....	5
A profile of Attributes .....	5
Creating an Attribute Profile .....	5
Collecting your Attributes .....	5
Example Profile submission data .....	6
The different Attribute Profile types .....	11
Value AP .....	11
Submission AP .....	11
Positive AP .....	11
Preference AP .....	11
Attribute Profiles for a Product .....	12
Attribute Profiles for a User .....	12
Profile type to products and users summary .....	14
Comparing Attribute Profiles .....	14
Sub-Profiles .....	15
<b>Summary .....</b>	<b>17</b>
<b>Afterthoughts .....</b>	<b>18</b>
Sub-Attributes .....	18
Star-rated Attribute Profiles .....	18
User APs that evolve as the user changes .....	19
Wide-open algorithms .....	19
<b>Appendix .....</b>	<b>20</b>
Appendix A: Attribute Profiling pseudo database schema .....	21
Appendix B: Sample Attribute Profile data .....	22
<b>Glossary .....</b>	<b>23</b>

## OVERVIEW

Attribute Profiling is a method to improve collaborative filtering. The ability to capture “why” a user likes (or dislikes) a product, and then relate those reasons to other products, is the unique advantage that Attribute Profiling provides.

Although it isn't required, if you have not read the former white paper, *Attribute Profiling: The evolution of collaborative filtering*, it is highly recommended you do so in order to fully understand the business context for Attribute Profiling.

In summary, it has been recognized that the collaborative filtering industry has reached a plateau of growth and a new data model is necessary to attain higher levels of product recommendation success.

***The collaborative filtering industry has reached a plateau of growth.***

Attribute Profiling is one proposal of such a data model.

### Terminology

Before we begin, please note that I use the term “product” to refer to any type of item in which you can apply Attribute Profiling to (e.g. a DVD). A “product category”, then, is a collection of the same type of products such as “movies”, “CDs” or “books”. Would you like to be put in contact with someone who shares the same interests and characteristics as you? It is interesting to note that “product” may refer to people, as well, in the case of social-networking or dating services.

For the sake of simplicity the primary product category of discussion will be movies.

Finally, to help with new terminology introduced within this white paper, there is a Glossary at the end if necessary.

# ATTRIBUTE PROFILING: A STEP BY STEP WALK-THROUGH

Attribute Profiling, in short, collects a pre-defined set of Attributes of a product, a user's preferences towards those Attributes, and allows a head-to-head comparison of both for unprecedented levels of analysis.

To fully understand what that means we'll be walking through an example to explain what Attributes are, how they build Attribute Profiles and the comparison of Attribute Profiles for product recommendation purposes.

## Defining "Attribute"

The American Heritage dictionary defines "attribute" as:

"A quality or characteristic inherent in or ascribed to someone or something."

An Attribute, in the context of Attribute Profiling, is the same thing except that it is what binds a product's description (*what* the product is like) to a user's preferences (*why* a user likes a product).

Attribute Profiling, then, collects and organizes the necessary data to capture the "what" and "why" for products and users while, simultaneously, providing a seamless way to compare and match the two together.

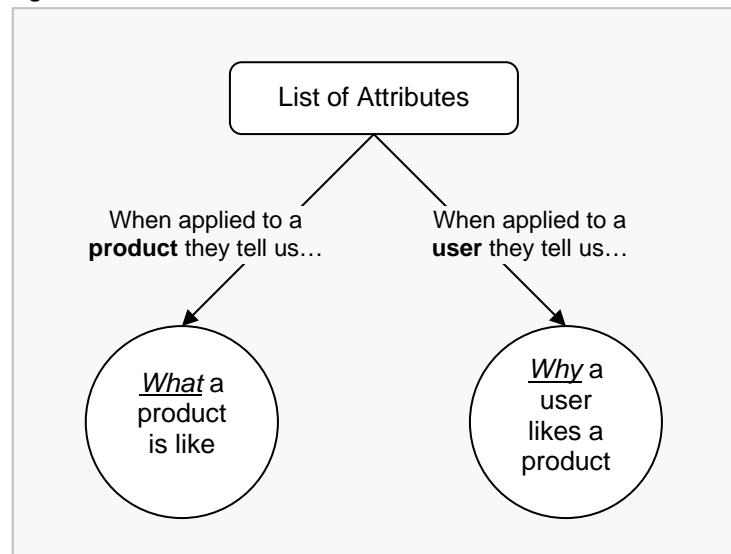
Let's take a look at Attributes applied to products and users.

### The "what" and "why" of Attributes

The Attributes applied to a product tells us *what* the product is like. They give us a weighted description of a product that goes beyond the black and white categorization of genres. For example, Attributes allow us to describe a movie as "full of action", "has a bit of romance", "a great ending" or "has mild gore". The values are not black and white, such as "has romance" or "does not have romance", but contain a weighted value relative to all of a product's Attributes.

Likewise, Attributes applied to a user tell us the *why* she likes or dislikes a product – it tells us her tastes of a particular product. For example, collecting user Attributes on movies is like having a user tell us she, "really likes action/adventure stories that are full of intrigue. My favorite, however, is a deep, romantic movie that has a happy ending – that is the best!" See Figure 1.

Figure 1



## Defining your Attributes

In order to create an Attribute Profile the first thing we must do is to define a set of Attributes for a given product. For example, for movie and music products the Attributes could be:

### Movie product Attributes:

Action / Adventure  
Animation / Computer Graphics  
Artistic / Surreal theme(s)  
Competition / Sports  
Drama  
Gore  
Humor  
Innocence / Youthfulness  
Insights / Education  
Intrigue / Suspense  
Music  
Philosophy / Spirituality  
Poignancy / Feeling  
Romance / Relationships  
Sci-Fi / Fantasy theme(s)  
Sexuality  
Story / Plot  
Violence

### Music product Attributes:

Experimental  
Hard  
Jazz  
Lounge  
Melancholy  
Pop  
Psychedelic  
Rap  
Rock  
Soft  
Soothing  
Sophisticated  
Strong beat  
Trance  
Up-beat

You may notice that for both product categories (movies and music) that some Attributes are genres as well. The key difference to note is that these “genres”, in the context of Attributes, are adjectives and not nouns. Attributes are concerned about adjectives and strive to describe a product by its various flavors, no matter how many there are. You could say that while genres are designed to categorize, and thus “lock down”, Attributes capture a product’s essence in a completely open-ended manner.

For example, in the music Attributes what genre does “soothing”, “up-beat” or “sophisticated” belong to? Any and all genres and that is exactly what we want – adjectives, not nouns.

## A profile of Attributes

Once we’ve defined our Attributes we can then begin to collect values for them relative to a product and/or user. From that data we can create a profile – a profile of Attributes. We’ll look at how this is done next.

## Creating an Attribute Profile

In order to clearly outline what an Attribute Profile (AP) is, and how to create one, we’ll walk through an example, step by step.

### Collecting your Attributes

Once your Attributes are defined the next step is to collect them. Not to collect the Attributes, themselves, but the values of each Attribute as they relate to the products (the “what”) and users (the “why”).

The simplest and most obvious way is to ask. At the same moment when a user would normally give a star-rating to a product you could simply have another section that asks the user to check off each Attribute of the product that they either liked or disliked. An Attribute rating GUI could look something like Figure 2.

The user is allowed to check off any Attributes that apply. Each check on a given Attribute is known as an Attribute submission. All Attribute submissions for a given product are known as a Profile submission. So what, exactly, does this do for us?

For each Profile submission (the Attributes a user applied to a product) we get two things: the “what” *and* the “why”.

### Example Profile submission data

To demonstrate, let’s take two users who have rated some movies they have recently seen. In this example the following is given.

- We have two users, John and Sue.
  - Both users complete Profile submissions on a small set of movies.
- John submits Attributes on:
  - *Lawrence of Arabia*,
  - *Groundhog Day* and
  - *The Fifth Element*.
- Sue submits Attributes on:
  - *Groundhog Day* and
  - *The Fifth Element*.

The Attributes that each user submitted, per movie, can be seen in Figure 3. Notice that John only checked “Liked” Attributes while Sue checked “Liked” and “Disliked” Attributes in her movies.

Figure 2

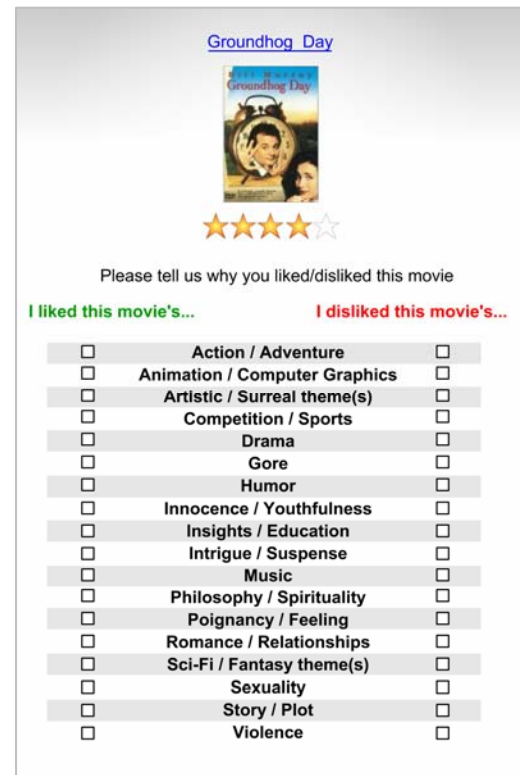


Figure 3

Name	Movie	Liked	Disliked
John	Lawrence of Arabia	Action / Adventure Story/Plot	
	Groundhog Day	Humor	
	The Fifth Element	Action / Adventure Humor Sci-Fi / Fantasy theme(s)	
Sue	Groundhog Day	Romance / Relationships Humor Philosophy / Spirituality	
		The Fifth Element	Action / Adventure Humor Sci-Fi / Fantasy theme(s)

(Note: A pseudo database schema can be found in Appendix A and corresponding sample data in Appendix B.)

Next, we will walk through how these Attribute submissions create the product and user Attribute Profiles at the same time.

### Movie Attribute Profiles based upon Figure 3 sample data

Figure 4

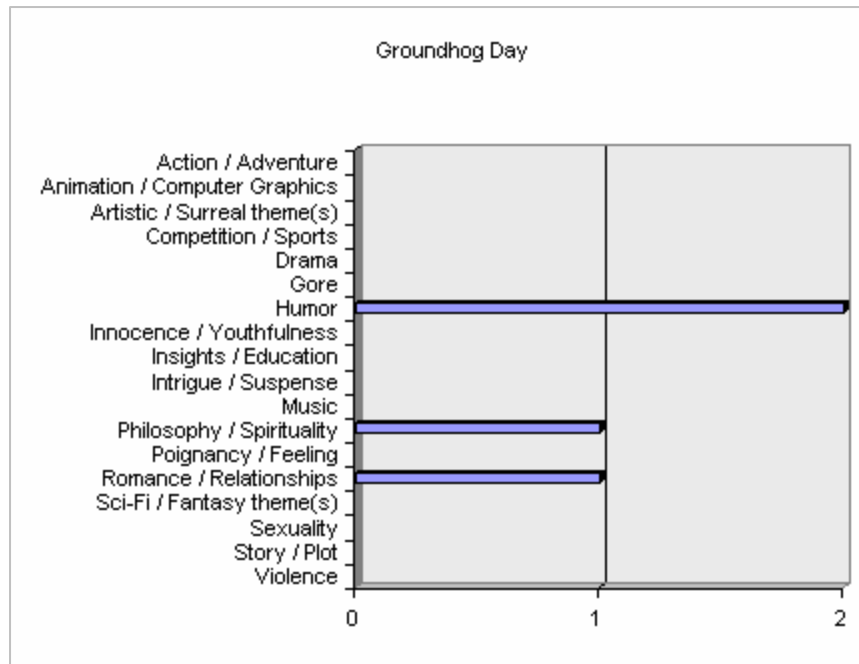
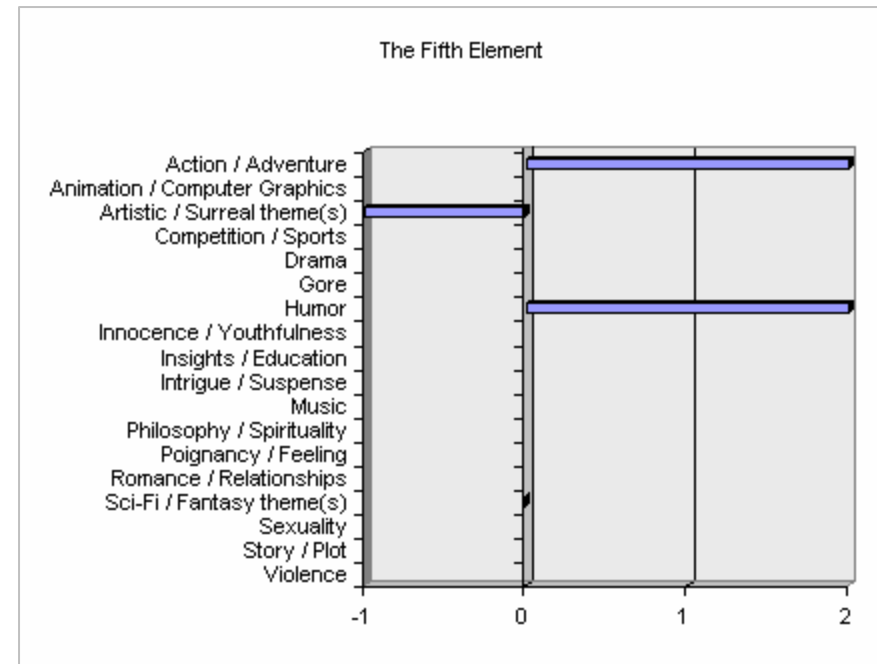


Figure 5



First let's look at the products. Every time a user applies an Attribute we add one to the total, cumulative value of that Attribute (for that product) if the user liked it, otherwise we subtract one. In other words, "Liked = + 1" while "Disliked = - 1" for each Attribute.

For example, both John and Sue liked the Humor Attribute of *Groundhog Day* therefore *Groundhog Day* has a two (2) in its Humor column (Figure 4).

However, for *The Fifth Element*, John liked the Sci-Fi / Fantasy theme(s) Attribute while Sue did not. As such we add one (1) for John's submission and we subtract one (-1) for Sue's. The two submissions cancel each other out and the end result is zero (0) for *The Fifth Element's* Sci-Fi / Fantasy theme(s) Attribute (Figure 5).

Then, if you take the cumulative values of all the Attribute submissions of thousands of users, as a whole, a profile emerges of the Collective's opinion. The Collective, then, defines the "what" of a movie (or product) – what Attributes are good, which are bad, etc... This is perfect because it doesn't matter whether or not the Collective's opinion is correct (if it were possible to determine so). Since they (the Collective) are our target audience it's the only opinion we care about.

### User Attribute Profiles based upon Figure 3 sample data

Figure 6

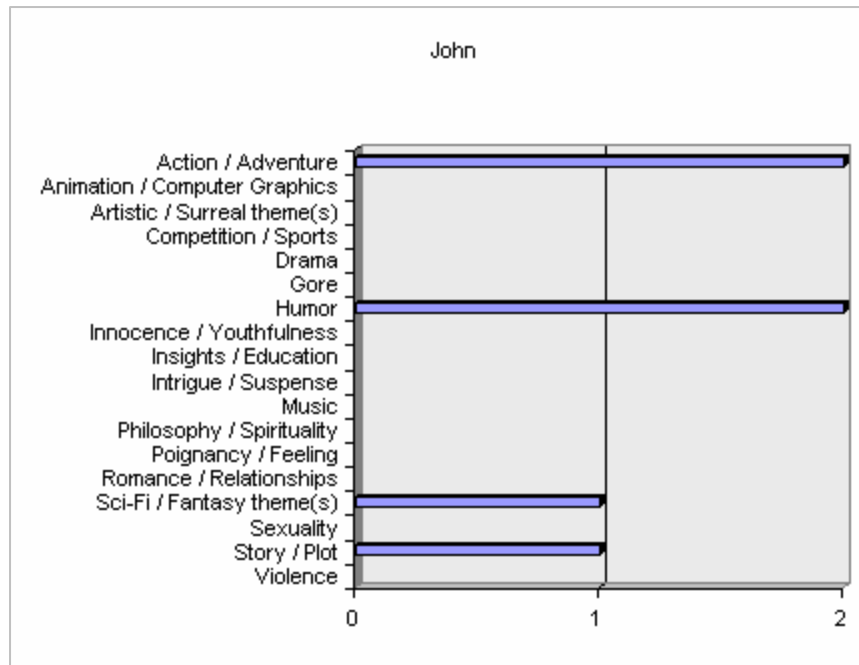
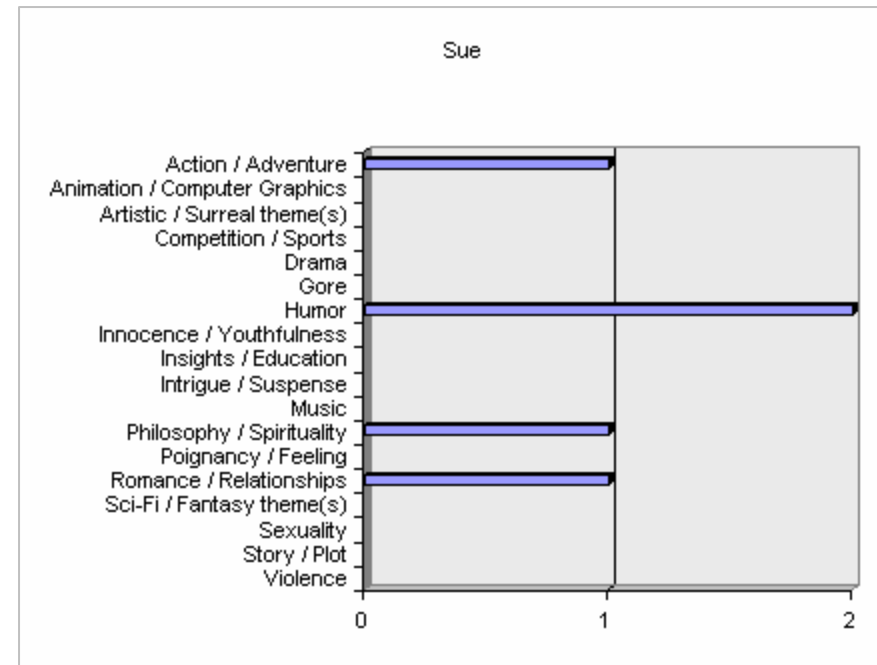


Figure 7



A User's AP is calculated a bit differently than the product's. For a user we add one to the total, cumulative value of a given Attribute only if the user *liked* it. In short, we only focus on the positive – all “disliked” Attributes can be ignored (for now).

The reason for this is because if a user dislikes an Attribute we don't know if she dislikes the movie's implementation of that Attribute or the Attribute inherently. In other words, let's say Sue disliked a movie's gore. Did she just not like that particular movie's gore or does she not like gore inherently? There is no way to know. This ambiguity on the disliked, or negative, Attributes is known as negative ambiguity.

If there is negative ambiguity then is there positive ambiguity? Nope. Again, this is perfect as every “liked” Attribute submission a user makes unquestionably defines exactly what she likes simply because she will only select “liked” on those Attributes that she likes. How else could it possibly be?

So, after many Attribute submissions we get the “why” of a user because, over time, a profile of why a user likes a product emerges.

Figures 6 and 7 show John's and Sue's Attribute Profiles respectively.

### Total user Attribute submissions vs. total movie Attribute submissions

Figure 8

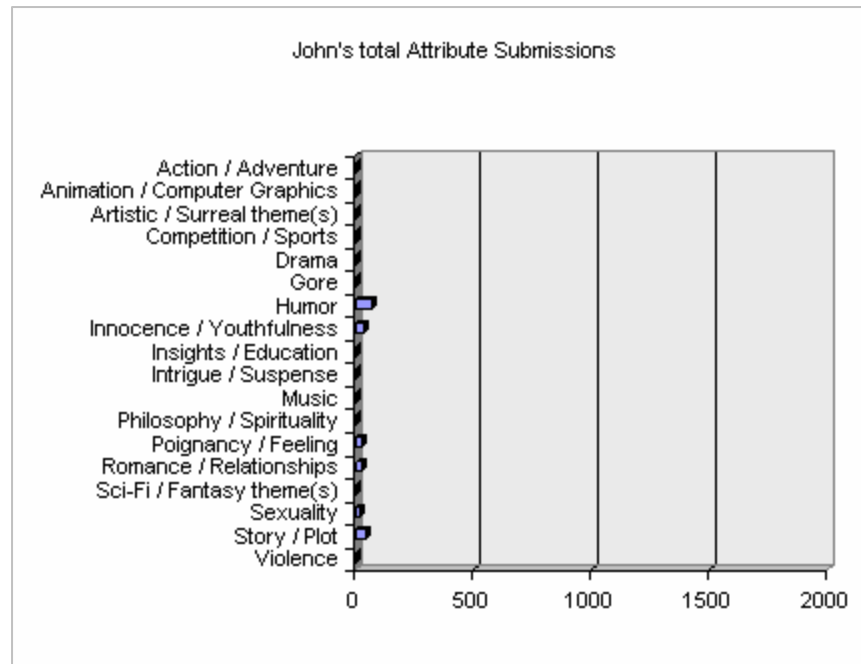
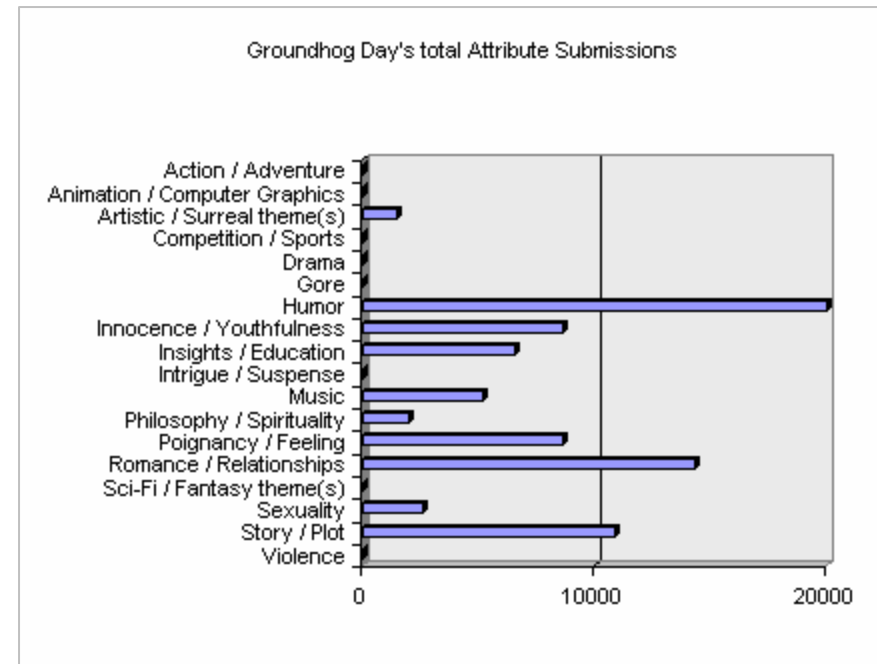


Figure 9



Now, after some period of time, let's look at each profile after John has made submissions on his movies (Figure 8) and thousands of users have made submissions on *Groundhog Day* (Figure 9).

As you can see, if we scale each graph to the same value (i.e. 2,000), the two do not compare easily to one another. The problem is, then, how do we compare apples to apples? The obvious thing to do is to normalize the data of each graph. By "normalize" I mean to reevaluate each Attribute as its percentage relative to the total all of the Attributes for a given user or product.<sup>1</sup>

By definition, to Profile a user or product is to normalize its Attribute data.

The normalized data then becomes that user's or product's Attribute Profile (AP). We'll take a look at John's and *Groundhog Day's* Attribute Profiles next.

<sup>1</sup> For example, if a movie has 1,000 individual Attribute submissions, and the humor Attribute was 300 of those, then its normalized humor value would be 0.3 (300 divided by 1000).

### User Attribute Profile vs. movie Attribute Profile

Figure 10

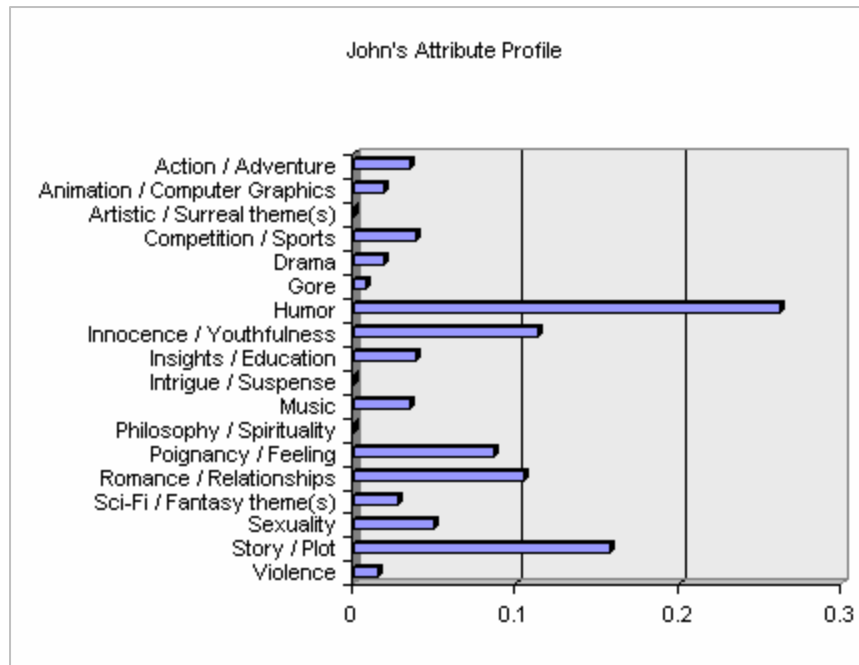
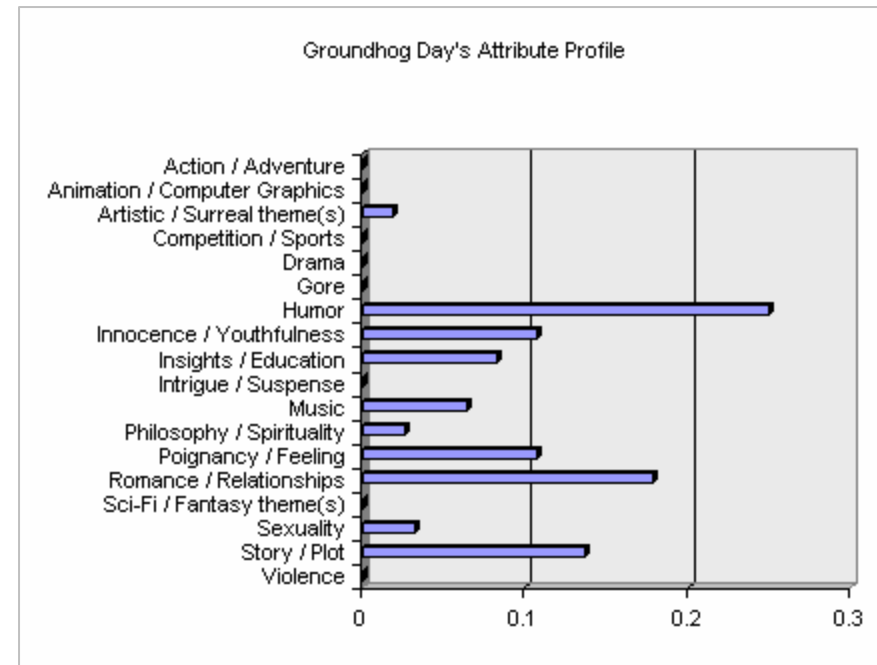


Figure 11



Now that each graph has been Profiled we can clearly see the two relative to one another. A user's AP describes why he/she likes a product (the user's "why") and a product's AP describes what it is like (the product's "what"). An AP gives a complete and unique description of a user and/or product – very much acting like a "fingerprint" – which then can be used to compare against one another.

At a quick glance it seems John (Figure 10) would really like *Groundhog Day* (Figure 11). They both share the same, prominent Attribute (Humor) as well as many others.

As another example, what about the mother who, because of her children, is concerned about PG-13 movies with sexual themes in them? Attribute Profiling can help her. We simply remove all PG-13 movies that have a Sexuality Attribute of any significant value from her list of possible recommendations.

With these kind of user specific insights Attribute Profiling offers a much richer and deeper comparison mechanism that simply was not possible before.

## The different Attribute Profile types

As we saw above there are different ways to calculate an Attribute Profile (AP). In fact, the way you calculate an Attribute Profile defines what it means. Although it is conceivably possible to create thousands of different Attribute Profiles I would like to highlight four, in particular:

- Value AP
- Submission AP
- Positive AP
- Preference AP

*No longer is a product bound to a scale rating between 1 and 10.*

Once I've explained each Profile I'll then show you how each has a unique meaning in the context of a product and user. A table summarizing the Profile to product/user breakdown is available at the end of this section.

Now, to help explain the different Profile types let's take the given: After 13 Attribute submissions 10 users liked Groundhog Day's Humor Attribute while 3 did not.

### Value AP

A Value AP is the most intuitive. Given that every "liked" Attribute receives a one (1) and every "disliked" Attribute receives negative one (-1) then the value of that Attribute would be all the "positives" minus all of the "negatives".

Given the example data above the value of Humor for *Groundhog Day* would be 7 (10 like – 3 dislike).

A Value AP, then, is when every Attribute within the profile is calculated this way.

### Submission AP

A Submission AP is when every Attribute's value takes on the total number of submissions for that Attribute regardless if they were positive or negative (i.e. "liked" or "disliked").

Given the example data above the absolute value for Humor on *Groundhog Day* would be 13 (10 like + 3 dislike).

### Positive AP

A Positive AP is when only the positive, or "liked", submissions are applied to an Attribute while the negative, or "disliked", submissions are ignored.

Given the example data above the positive value for Humor on *Groundhog Day* would be 10 (10 like).

### Preference AP

A Preference AP, by definition, is any Profile that has one or more Attributes that are calculated differently from the others. For example, you can build a Profile that has some Attributes calculated as a Value AP while others are calculated as a Submission AP.

In short, if there is any business logic used on one Attribute that is differently from the others then that makes it a Preference AP.

It is called a Preference AP because these Profiles will invoke business logic to more accurately capture a user's preferences for the purpose of product Profile comparisons (and, thus, recommendations).

## Attribute Profiles for a Product

The Value AP of a product tells us what the movie is like – or how good it is.

As the Collective's opinion builds on a product the corresponding Value AP reflects that opinion precisely.

No longer is a movie bound to a scale rating between 1 and 10. With Attribute Profiles not only can you derive the movie rating but, in addition, each Attribute gets a rating as well.

A movie that receives a 7.2 value overall may have a Humor Attribute value of 8.9 and a Plot Attribute value of 4.6. If Plot isn't as important to you as Humor (as shown within your AP) then this movie may be more like an 8.0 for you.

The Submission AP of a product tells us the contents of a product.

Let's say there is a horror movie that has a Gore Attribute value of -2.0 (all the "likes" minus the "dislikes") and a Gore submission value of 9.0 (i.e. 9 gore submissions were made) from a total of 10 Profile submissions (10 users submitted Attributes against this movie). This tells us the vast majority of all user submissions (9 out of 10) checked that they either "liked" or "disliked" the Gore Attribute.

***Attribute Profiling allows for movie content descriptions that go beyond "Strong sexual content" or "Mild language".***

If a user likes a movie's gore then, of course, that movie must have gore in it in order for the user to like it. If a user dislikes a movie's gore then the same thing is true. Whether or not a user likes or dislikes a movie's gore simply means that movie must have gore in it.

While the -2 tells us slightly more users disliked the gore rather than liked it, the 9.0 (out of 10) tells us the movie has a lot of gore regardless!

This allows a content description of movies that goes beyond "Strong sexual content" or "Mild language". Now we can apply a rating to it. For example, a Violence Attribute value of 2.0 or lower may be okay for children but anything above a 5.0 could be intended only for adults.

A Positive or Preference AP has no meaningful context for a product.

## Attribute Profiles for a User

The Value AP of a user tells us his/her experience so far.

To explain this let's take a user who loves comedy. This user rents 10 comedy movies and they all turn out to be horrible. He completes 10 Attribute submissions on those movies and gives each movie a negative (or "dislike") for the Humor Attribute. If we look at his Value AP that user will have a negative ten (-10) Humor Attribute value. But we just agreed that he loves comedy! So that is incorrect. What it does tell us, however, is that he has watched 10 movies that he doesn't like.

By applying a filter on the data we can use a user's Value AP to get feedback against the recommendations made to that user.

In short, a user's Value AP only tells us what that user's experience has been so far for each Attribute but it doesn't tell us what he likes.

The Submission AP of a user tells us what is important to the user (not necessarily what she likes).

Again, using the mother who is concerned about PG-13 movies with sexual themes because of her children. If she rates nine (9) movies with a "dislike" for the Sexuality Attribute then her Submission AP will have a nine (9) value for Sexuality. So, by combining her Submission AP value (9) with her Value AP

value (-9) we can derive two things: 1) her Submission AP tells us sexual themes are important to her, and 2) her Value AP tells us she doesn't like them for those given movies.

The Positive AP of a user tells us explicitly what the user likes.

A Positive AP only captures positive (or "liked") submissions and so, by definition, this bypasses negative ambiguity and, at the same time, explicitly tells us exactly what the user likes.

Again, if a user likes an Attribute of a movie then it's a given that the user must like that Attribute inherently to some degree. Of all the Attributes a user likes she will naturally select her favorites over the others.

***It reflects exactly what a user likes and to what degree they like it.***

For example, if you like humor more than action then you will naturally be attracted towards the humorous qualities of a movie over its action and, thus, will submit more "liked" humor Attributes than action.

In this manner a user's Positive AP will reflect exactly what a user likes *and to what degree they like it*.

The Preference AP for a user is the most accurate Profile of what a user likes (or dislikes) as derived from applied business logic.

Anytime you take a user's Profile (of any type) and apply business logic to change the calculations for a better user representation then that Profile, by definition, becomes the user's Preference AP.

For example, let's say you currently use a user's Positive AP to compare against other products' Value AP to find recommendations. That makes sense; you take what the user likes and find a product that closely matches.

However, you know you can get better matches if you were able to adjust the user's Positive AP to more accurately reflect what they like. Sure, that makes sense as well. But how would you do that? One way would be to give a weight to the submissions he made when calculating his Positive AP. Meaning, of all the "liked" submission a user makes across many products you can give a higher weight to those products that he rated higher via the five-star rating value.

For example, if he gave movie A five stars and movie B only two stars then the "liked" Attributes on movie A will have a larger impact on his Positive AP than the "liked" Attributes of movie B.

This additional business logic converts a user's Positive AP into his Preference AP.

As another example, a user could specify which Attributes he prefers within his online account profile. You can then use those user preferences to adjust the values of his Positive AP as well. Again, this extra business logic makes it his Preference AP.

So, in short, all the other "pure" Profiles above (and any others you can come up with) could be considered inputs into your business algorithm that will output a user's Preference AP (which, in turn, is used to find product recommendations).

A Preference AP, then, is just a means to put a name to this custom, open-ended profile.

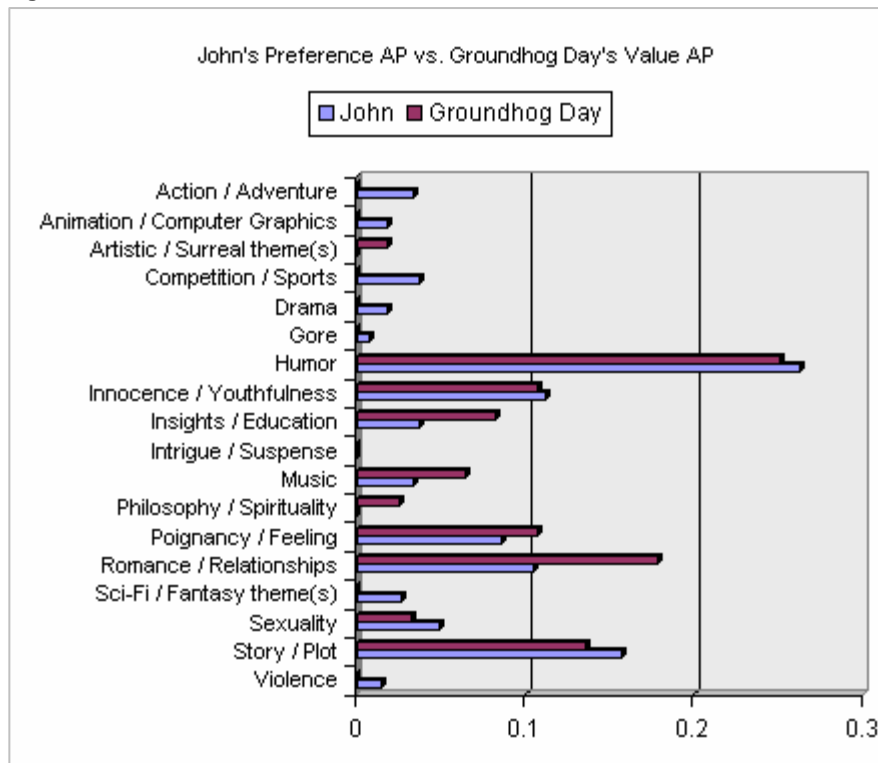
## Profile type to products and users summary

Profile Type	Product	User
Value AP	What the product is like (the "what")	The user's experience thus far
Submission AP	Contents of the product	What's important to the user
Positive AP	N/A	What the user likes (the "why")
Preference AP	N/A	Applied business logic to more accurately capture user's likes/dislikes/preferences (a more accurate "why")

## Comparing Attribute Profiles

Since the Attributes on a product and the user are the same set of Attributes we can easily compare them side-by-side to find the best match. Given the Profile types above, what we want to do when comparing a user to a product is to compare a user's Preference AP (what the user likes – however that is defined) to a product's Value AP (what that product is like). Figure 12 shows an example of a user's Preference AP compared against a product's Value AP. It seems John would like the movie Groundhog Day very much.

Figure 12

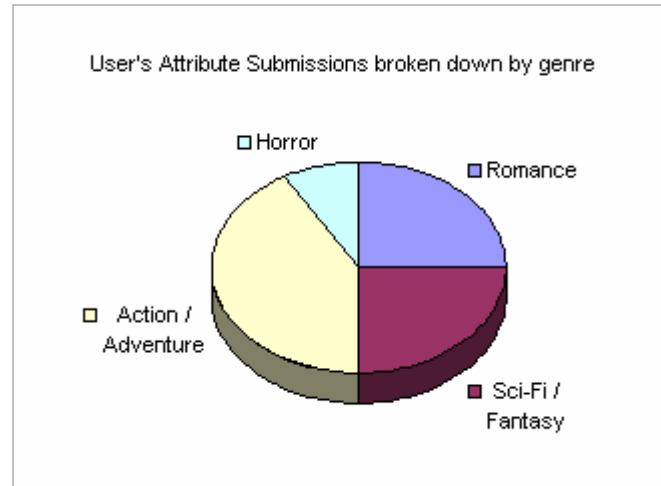


While this will give us good product recommendations we can leverage your existing data structures and do ever better.

## Sub-Profiles

As a user makes many Profile submissions they begin to develop a Preference AP. At the same time they also begin to develop sub-Profiles based around the metadata of a movie, such as genre. If we take all of the submissions a user makes we can also group those submission by genre. See Figure 13.

Figure 13



Looking at a user's (total) Attribute Profile tells us what the user liked/disliked across *all* genres.

If we isolate only those submissions that belong to a genre then we can derive much more information about what a user likes and provide even better recommendations.

A sub-Profile such as this would be considered a user's Preference-Genre AP. Therefore, when recommending a movie from a particular genre you would want to compare the user's Preference-Genre AP to the Value AP of all the movies within that genre.

Let's see how we can derive more information from a sub-Profile using Sue as an example next.

### User Preference AP vs. user Preference-Genre AP

Figure 14

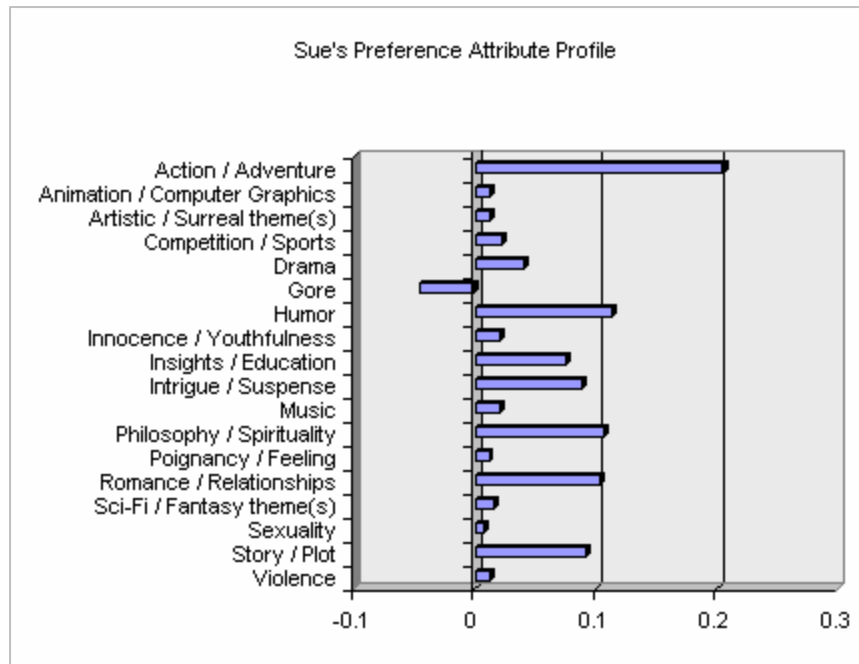
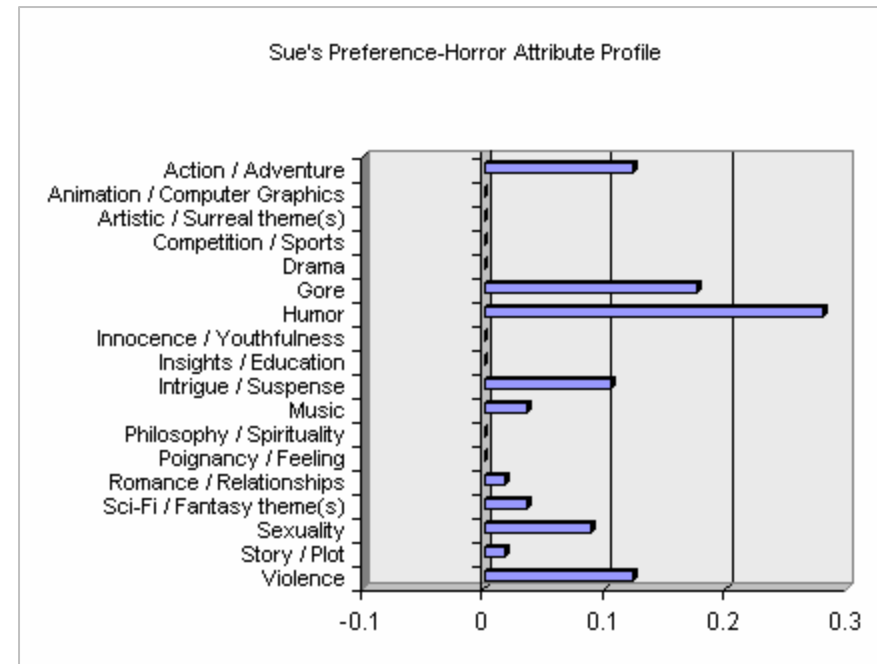


Figure 15



Looking at Sue's Preference AP across all genres tells us what her overall movie likes/dislikes are. Notice the only negative we have is the Gore Attribute (Figure 14).

Now, if we build a Profile based upon only those submissions that are related to horror movies we can build Sue's Preference-Horror AP as well (Figure 15). This looks dramatically different from her overall Preference AP. While, overall, it may seem like she doesn't like gore, when we take a look at her Preference-Horror AP we see it's one of her favorite Attributes!

Now let's take it one step further. Not only can we see that Sue likes gore in her horror movies but we see that she likes humor even more. It is as if she is telling us directly, "I don't care too much for the serious horror flicks, but I love the fun, campy ones!"

How exciting to be able to accurately capture such detail about a user!

This is yet another way how Attribute Profiling allows you to gain insights into *why* someone likes a movie. These kinds of insights are simply not possible with the current state of collaborative filtering.

## SUMMARY

In summary, Attribute Profiling collects and organizes the necessary data to capture the “what” (what a product is like) and “why” (why a user likes/dislikes a product) while simultaneously providing a seamless way to compare and match the two together.

***A superior service is undoubtedly the most attractive beacon for any user.***

Focusing around adjectives, it captures a structured “fingerprint” for products and users’ preferences that allow easy, apples-to-apples comparison to identify accurate product recommendations.

Once implemented, the easy, intuitive and extended avenue in which a user can express their feelings will pull them to submit Attribute data for a given product. Also, as data increases, so will the product recommendation accuracy, increasing the user experience. As the user experience increases via Attribute Profiling it will incite a stronger pull to submit more Attribute data (knowing that it is used to improve the recommendations).

This upward, self-feeding spiral will reach a critical point and take off very much like Google’s search engine.

After all, a superior service is undoubtedly the most attractive beacon for any user.

## AFTERTHOUGHTS

Below are some thoughts relating to the topics above to help generate further discussions and/or insights.

### Sub-Attributes

The Attributes themselves are completely open to be defined. Which Attributes best captures a user's feelings on a product? How detailed should the Attributes be? To these questions the idea of sub-Attributes has come about. In the movie Attribute examples above one of the categories is "Humor". Is that enough? There are many types of humor and many users prefer some types over another. Wouldn't it be nice to be able to identify movies that have slapstick humor in them so you can recommend said movie to those users who like slapstick? Or avoid the ones that don't?

If "Humor" is an Attribute then some sub-Attributes could be: Slapstick, Dry, Prank, Campy, Standup, etc...

We can even extend this concept to include different types of endings. There could be an "Ending" Attribute with sub-Attributes as: Happy, Realistic, Hollywood, Heavy, etc... The options are completely wide open.

Beyond this there are GUI considerations as well. How do you display sub-Attributes easily without the user clicking on tree nodes to expose them? (It is of the author's opinion that the more clicks you force a user to endure the less likely he/she will complete the task at hand.) Of course with Flex, DHTML, Java, Ajax, etc... there is no reason to see this as a major problem but it's worth keeping in mind when defining your Attributes and designing your website.

### Star-rated Attribute Profiles

Throughout this paper Attributes are captured as either "liked" or "disliked" values. However, there is no reason why they have to be this way. Another possibility would be to have a star-rating per Attribute as shown in Figure 16.

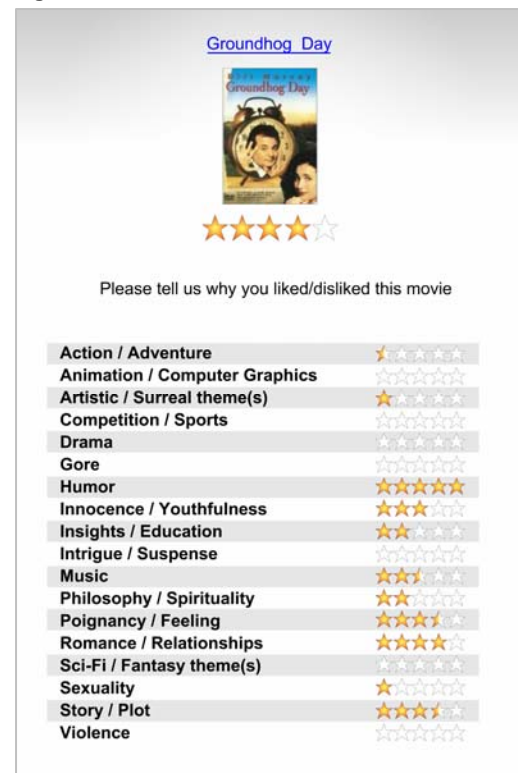
Just as a user rates the movie, as a whole, he can rate each relevant Attribute. Product and user Profiles can then be generated off of this data.

Naturally there are trade-offs for each type of implementation. In a star-rated implementation it may be more difficult to track which Attributes a user didn't like for a movie because star-ratings are, by definition, positive values. This means that in order of a user to mark something as negative he must apply some positive value to it. If, for example, a user rates an Attribute only one star does that mean he doesn't like the Attribute at all or that he only likes it a little bit? Negative ambiguity may have crept back into the picture.

While the discussion of these issues are beyond the scope of this paper, the point is there are many possible ways to capture Attributes and each implementation offers unique pros and cons that need to be evaluated.

In the end it will be a company's unique approach that will distinguish it from its competitors in its perpetual pursuit of better customer service.

Figure 16



## User APs that evolve as the user changes

A very interesting aspect of Attribute Profiling is that it may capture the user's tastes over time. People are not static. Their tastes, let alone their moods, are not static. While, one day, a user may be bored of documentaries they may crave them a year later (or visa versa). Nothing stays the same.

If, when we build a user's Preference AP, we weigh the current submissions more than the older ones we allow the profile to "bend" towards the user's current tastes. If the user decided that she really likes documentaries now, while before she didn't, this will be reflected within her Preference AP as she submits more "liked" Attributes on those films.

Another advantage to this – besides being able to capture a more accurate reflection of what the user likes – is that it means you no longer have to store these user preferences within their website profile (which are static).

*What seemed like a limitation suddenly becomes an open field of possibilities.*

In short, we get the option to implement an additional two "wins" for free:

- 1) A more accurate Preference AP and
- 2) Fewer options the user has to maintain within her website profile.

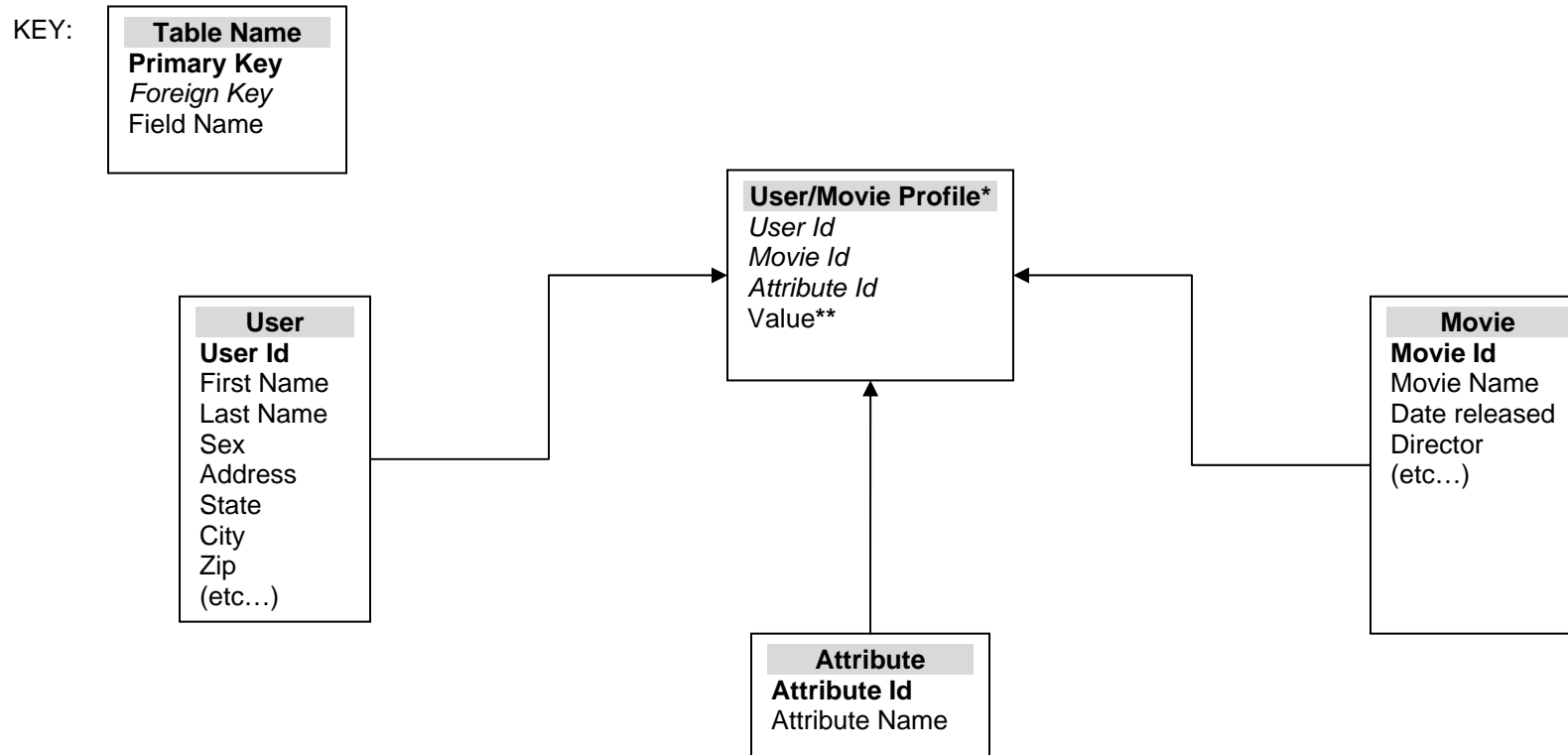
## Wide-open algorithms

Once Attributes have been defined, and data has been collected, the possibilities for various types Profiles, let alone comparisons, are vast. The richness of data that Attribute Profiling offers opens up a whole new world of possibilities that we've only begun to touch upon within this paper. Because Attribute Profiling fits inside of your current implementation, existing comparison algorithms may be applied to derive new levels of recommendation accuracy.

Like shifting gears in a red-lined car. What seemed like a limitation suddenly becomes an open field of possibilities.

## APPENDIX

## Appendix A: Attribute Profiling pseudo database schema



NOTE: This diagram is only intended to represent a design concept. Naturally the table names, field names, table normalization/de-normalization, etc. would need to be configured according to the target development environment/requirements.

\* The Primary Key of the User/Movie Profile table is the unique combination of the “User Id”, “Movie Id” and “Attribute Id” Foreign keys.

\*\* Here “Value” would be a simple integer value of three possibilities. For example, “-1” for negative and a “1” for positive (there is no need for a record if the value is “0”. Meaning it has no impact so there is no need to track it.)

## Appendix B: Sample Attribute Profile data

(Based upon the Appendix A pseudo database schema)

### User

User Id	First Name	Last Name	Sex	Address	State	City	Zip	...
10	John	Smith	M	123 Somewhere	CA	Somecity	94111	...
20	Sue	Chang	F	456 Thisway	NY	Another	10500	...
...	...	...	...	...	...	...	...	...

### Movie

Movie Id	Movie Name	Date Released	Director	...
100	Lawrence of Arabia	1962	David Lean	...
200	Groundhog Day	1993	Harold Ramis	...
300	The Fifth Element	1997	Luc Besson	...
...	...	...	...	...

### Attribute

Attribute Id	Attribute Name
1	Action / Adventure
2	Animation / Computer Graphics
3	Artistic / Surreal theme(s)
4	Romance / Relationships
5	Competition / Sports
6	Drama
7	Gore
8	Humor
9	Innocence / Youthfulness
10	Insights / Education
11	Music
12	Philosophy / Spirituality
13	Poignancy / Feeling
14	Sci-Fi / Fantasy theme(s)
15	Sexuality
16	Story / Plot
17	Intrigue / Suspense
18	Violence

### User/Movie Profile

User Id	Movie Id	Attribute Id	Value
10	100	1	1
10	100	16	1
10	200	8	1
10	300	1	1
10	300	8	1
10	300	14	1
20	200	4	1
20	200	8	1
20	200	12	1
20	300	1	1
20	300	3	-1
20	300	8	1
20	300	14	-1
...	...	...	...

## GLOSSARY

<b>Absolute AP</b>	<i>See Submission AP</i>
<b>AP</b>	<i>See Attribute Profile</i>
<b>Attribute</b>	An adjective that describes a product yet is shared across all the products within its product category.
<b>Attribute Profile</b>	The normalization all the values of each Attribute within a product profile. This allows the “apples to apples” comparison between two profiles that share the same Attributes.
<b>Attribute submission</b>	When a user submits a “liked” or “disliked” on a given Attribute for a given product.
<b>Negative ambiguity</b>	Negative ambiguity is when a user dislikes an Attribute on a given product and we are unable to tell if the user disliked the Attribute inherently or disliked the product’s implementation of that Attribute. For example, if a user disliked a movie’s violence do they dislike violence, in general, or did they only dislike that movie’s version of violence?
<b>Positive AP</b>	An Attribute Profile in which the weight of each Attributes value is only based upon the “liked”, or positive, Attribute submissions. All “disliked”, or negative, submissions are ignored. For example, if a user likes 3 movie’s humor but disliked 1 movie’s humor then the Humor Attribute would be weighted with a value of 3.
<b>Preference AP</b>	An Attribute Profile in which any business logic is used to more accurately reflect what the user likes or dislikes.
<b>Preference-Genre AP</b>	A subset of the Preference AP in which the input data is filtered by a particular genre.
<b>Product</b>	Any noun in which a user may apply an Attribute to. For example, a DVD or book.
<b>Product Category</b>	A collection of like products. For example, movies, books or CDs.
<b>Profile</b>	A list of Attributes for a given product that is shared across all products within a product category.
<b>Profile submission</b>	The total Attribute submissions a user submits for a given product.
<b>Submission</b>	<i>See Attribute submission</i>
<b>Submission AP</b>	A Submission AP is in which each Attribute’s value is based upon the total submission count regardless if those submissions were “liked” or “disliked”. For example, if a user likes 3 movie’s humor but disliked 1 movie’s humor then the Humor Attribute would be weighted with a value of 4. It is also known as an Absolute AP.
<b>Value AP</b>	An Attribute Profile in which the Attributes are weighed by the sum value of positive and negative submissions. For example, if a user likes 3 movie’s humor but disliked 1 movie’s humor then the Humor Attribute would be weighted with a value of 2.