Feeding the Machine: Developing A Content Based Filtering System for Food Blogs

Molly Domino

March 2, 2014

Contents

1 Introduction **3**

2 Terms **3**

3 Existing Algorithms **5**

4 The Domino Recommender System **5**

4.1 System Specific Terms 6

4.2 Recommending a Blog: Best of 10 Function 6

4.2.1 Evaluation Function 6

4.2.2 Proofs 7

4.3 Receiving Feedback From the User 8

4.3.1 Proofs 8

5 Process **9**

5.1 Scoping the Project: Food Blogging 9

5.2 The Categories 10

5.3 Creating the System 13

5.4 Testing the System 14

6 A Simple Run Through **14**

7 System Analysis **15**

7.1 The Issue Of Generating Diversity 15

7.2 The Cold Start Problem: 15

7.3 The Issue Of Grey Sheep: 15

7.4 The Issue Of Correctness 15

7.5 The Issue Of Speed: 15

7.6 Restrictions 16

8 Significance of The Domino Recommender System **16**

9 Future Work **17**

10 Reflection **17**

11 Conclusion **18**

A Annotations On Blogs In Database **21**

B Quality Scores In Each Blog **21**

1 Introduction

When we as humans like something, we often want to see more of it. Many web based companies have become exceptionally profitable off of software that can suggest new information based on a users preference. In this paper, I will start out by exploring Recommender Systems as well as some of the general differences in theory. In Section 2, I will discuss the benefits and drawbacks to existing algorithms for suggestions, followed by the introduction of my own linear time and space system in Sections 3 and 4. My specific implementation of this system involved looking closely at food writing on the internet, specifically blogging. Thus, section 5 details my thought process and defends the database I created for my system to work with. This serves as a defense of my implementation, showing that the database I was working with was adequately detailed enough to draw conclusions off of. Next, to further defend my system, I demonstrate the results of my system in action. In Section 7 I weigh the strengths and weaknesses of my own system, and in section 8 discuss why this is an important development to the world of Recommender Systems. In the final two sections of this piece, I consider where I would like to take this research in the future and reflect on my past experiences.

2 Terms

At its most abstract, this course of study has been considering the general field of Similarity Searches. The term "Similarity Search" is applied to any problem where the only relationship any two items in a (typically vast) data set have is how similar they are. Such data is often "unlabeled data" (CITE). However, this problem is only concerned with the subproblem of Similarity Searches: Information Filtering. This field focuses on creating a system that removes redundant or unwanted information in an automated fashion for presentation to a user. Even more specifically, the style of algorithms examined in this paper are concerned with presenting a user with information they have not experienced before. Recommender Systems model exactly this process. Ultimately, these systems are working to return data the user will find appealing based off of their preferences (CITE).

In any Recommender System, there are two components that intermingle. Users, as their name implies, are entities that use a system. Their preferences are inputted and some results are returned to them based off the data known. These results are in turn called items or objects. Based on these preferences, it is the job of such a system to draw conclusions about the user and return an item that the user will appreciate.

These systems have two major forms: content based filtering and collaborative filtering (CITE). Each approach can be useful, but each also has significant problems to consider.
Some of the major qualities all Recommender System algorithms must consider are:

1. **Creating Diversity:**
   One of the major expectations of most Recommender Systems is that they will return not only content that a user will like, but that is new to the user. Logically, this makes sense. Users do not need an algorithm to tell them things they already know: they are looking for something new. Pandora, especially, has gained notoriety for this feature. Therefore, these algorithms are not necessarily striving for a "best possible" match in their search for related content. Instead, in this case a "good enough" match is in fact superior. Thus, a strong algorithm (be it collaborative or collective) needs to ensure it isn't returning "stale" information (CITE).
2. **Gray Sheep:**
   Items that do not consistently agree or disagree with anything else in the dataset, are called "gray sheep". This is because they are neither a strong match or a definitively poor match.

Gray sheep are difficult to relate to other items in the dataset, thus increasing space and time without offering anything beneficial (CITE).

3. **Reasoning Behind Suggestions:**
   The problem of determining what aspects of a suggestion were appealing (or unappealing) to a user can be immensely difficult. Once a user has offered some feedback about an algorithm's suggestions, automated systems need a way to determine what they enjoyed and disliked about a given suggestion. These programs need to adapt to match not only a user's initial preferences, but any other preferential data entered (CITE).

4. **The Cold Start Problem:**
   Recommender Systems function by drawing conclusions off of data. However, if there is too little data, the conclusions even a functional algorithm can draw will be skewed (CITE).

Collaborative filtering bases its structure on the idea that if a user A and another user B share a similar opinion on idea x, it is likely they will also share an opinion idea y (CITE). Amazon.com, though certainly not the only site to do this, has a feature on any product page that notably uses a collaborative filtering algorithm. Below the description of a product, Amazon notes what other users bought along with a given item as well as what else other customers looked at in a more general sense. Both the Slope one algorithm and the (conveniently named) Collaborative Search Engine model employ this style of filtering. On one side, suggestions based off of other user's preferences are more or less guaranteed to have some diversity to them based off the fact that is unlikely for two people to have the exact same preferences on everything. Thus, there is always a force of entropy working to ensure diversity of information (CITE). Logically, it is also easy to prove that this concept will always return favorable results as it provides a clear metric to measure happiness by. At the same time, these algorithms struggle with the cold start problem, and gray sheep. Also, as their conclusions are drawn off of the opinions of others, past data can be skewed for unimportant reasons, called shilling attacks. For example, if some users arbitrarily disliked some items and liked others, potential suggestions will be less accurate than they could be.

Content based filtering (also sometimes referred to as model based filtering) relies only on the data set to draw conclusions rather than user's opinions of that data set (CITE). In contrast to collaborative filtering, they require only a small amount of information to return accurate results. This is due to the fact that such systems relate potential items to suggest by how closely they match a user's preference. Thus, this category of algorithm can be run correctly with only one user's initial opinions. However, such algorithms are limited in that every item it could possibly return must be categorized and related to all other items in the data set (CITE).

Pandora's Music Genome Project is a strong example of the problems involved in content filtering. Every song in their extensive library is modeled as a vector with between 150 and 500 musical genes scored on a scale of 1-5 (CITE). It then uses a distance function to determine how similar two vectors are. While results from the Music Genome Project are notably accurate, the process of adding a single song to the database is impossibly slow. Without even considering the fact that a song's genes are all scored by between one and two employees of Pandora, this algorithm takes 20 minutes to relate two songs (CITE)! With over 900,000 songs in their database as of 2011 (CITE), adding music has become a mammoth operation. Furthermore, there is a chance for diversity to be limited in such systems as "best fits" are typically more likely to be returned over "good enough" fits. On a more abstracted level, both Bayesian Networks and Clustering Models both base their functionality off of content based filtering (CITE).

As a note, it is not out of the question to create a hybrid of these two forms of algorithms either, in an effort to optimize the strengths of each choice.

3 Existing Algorithms

While there are several published papers on different pattern finding and suggestion algorithms, it is important to note how it is difficult to determine exactly how many different systems for recommendation there are. Outside of Pandora's Music Genome Project, several major web corporations like Rotton Tomatoes, Netflix, Internet Movie Database (better known as IMDB), among others treat their systems and suggestion algorithms as trade secrets. Thus, a truly comprehensive discussion of existing recommendation systems is not possible. What is included in this section is a discussion of some of the more widely used and publicly available algorithms (CITE). For more information, one might look to one of the many open source Recommender Systems like Milk, Crab, Waffles, Shogun, as well as many others.

One form of collaborative filtering bases its assumptions on the statistical correlation coefficient between two user's responses. *Correlation Coefficients* offer a measure of confidence in the relationship between two items in the form of a number between 1 and -1. A correlation coefficient of 1 implies there is a positive linear relationship while -1 implies a negative linear relationship, and 0 implies no relationship. (CITE). In the case of collaborative filtering, this value provides a method of determining the relationship between two users where the more positive the relationship the better. Given two users responses to various items, this value can determine if their response to some new item will be the same (CITE).

A handful of algorithms fall under the umbrella term of *Clustering Algorithms*. Typically this concept is more often seen in content based filtering, but could prove to be a useful tool in collaborative filtering. Such algorithms are said to "find the general structure in a set of unlabeled data" (CITE). Typically the clusters considered these are distance based clusters. Assuming every item was plotted on a two-dimensional plane, items within a certain distance of each other would be assigned to the same cluster. In the case of some more complex systems, items are considered to be vectors and their distance calculated through Euclid's algorithm (CITE).

Bayesian Belief Networks, a system common to content based filtering, are often represented as directed trees of items to denote relationships between unlabeled items (CITE). These graphs are based off of propositional logic or the idea that $p \implies q$ and the logical rules surrounding it and act in a similar manner to neural networks. Any node in a Bayesian Belief Network has an associated boolean function and takes in a set of values from its parent nodes (CITE). Content based filtering in this regard is based off of the idea that if a user enjoy an item i, it is likely that they will enjoy an item in $\mathrm{N}(i)$. Once such a graph is constructed, there are several algorithms to traverse and perform inferences on such a structure. When these graphs have an edges without direction, they are more often called *Markov Networks*.

At the moment, the bulk of research of in new methods of recommendation is being done in collaborative based filtering as the assumptions it bases its assumptions off of clear and objective data. However, as part of my studies was in the critical examination of food writing (not to mention concerns about gathering enough data to prove the Cold Start Problem was not an issue) I chose to create a system with its filtering based on content, rather than popular opinion.

4 The Domino Recommender System

This Recommender System is not a single algorithm, but three processes that work together with a database of blogs. Thus, each algorithm will be presented and analyses on an individual level. Beyond these algorithms, my program interfaces with a database of Blog objects. However, this database could be abstracted to any kind of object and is in no way Blog specific.

4.1 System Specific Terms

- **Blog:** refers to an object. Each food blog that could be potentially suggested in this system is represented as a Blog object. Each contains information on the specific blog (the title, author, url, etc) as well as its scores in every category.
- **Q** : A dictionary of qualities mapped to a Blog's specific scores. Every Blog contains a dictionary Q, and $|Q|$ is the same across all Blogs.
- $\mathbf{q_n}$: refers to a specific score of a quality in Q.
- **Template Blog:** A term used to refer to the Blog (and associated preferences) the user enters. Template.Q becomes the standard all potential suggestions are compared against.
- **SuggestedBlog:** A term used to refer to the Blog (and associated preferences) the Domino Recommender System suggests.

4.2 Recommending a Blog: Best of 10 Function

Once a user has entered a Template, the first responsibility of this program is to suggest a SuggestedBlog. This algorithm starts out by drawing a Blog object from the database at random. This Blog is now considered the SuggestedBlog and SuggestedBlog.Q is compared against Template.Q through an evaluation function. This algorithm then draws out another blog and the process begins anew. In total, each running of the algorithm draws out 10 Blogs at random, and the lowest scoring (and therefore best) of these 10 iterations is suggested to the user. In pseudocode, this process functions as follows:

Overall Recommendation Algorithm

```
Blog SuggestedBlog = None
int BestScore = 100

for i in range (10):
        Blog NewBlog = generateRandomBlog()
        int tmp = evaluate (TemplateBlog, NewBlog)
        if (BestScore > tmp):
                BestScore = tmp
                SuggestedBlog = NewBlog
return SuggestedBlog
```

4.2.1 Evaluation Function

The evaluation function mentioned above iterates through Q, comparing any element q_n in Template and SuggestedBlog. This function calculates x where $x = \sum_{i=1}^{|Q|} |\text{Template}.q_i - \text{SuggestedBlog}.q_i|$. It should be noted that in this function, any time when $\text{Template}.q_n$ and $\text{SuggestedBlog}.q_n$ are the same number, no summation is calculated as the difference will always be 0. In pseudocode, this method translates to:

The Evaluation Function

```
total = 0

for quality in Template.$Q$.keys():
        if (Template[quality] != SuggestedBlog[quality]):
                x = |Template[quality] - SuggestedBlog[quality]|
                total += x
```

```
return total
```

From this code, it is clear that this function calculates the positive difference between the scores for a given quality. This is important as, if Template.q_n=9 and SuggestedBlog.q_n =0 we see a score of 9. However if in that same blog Template.q_m = 0 and SuggestedBlog.q_m = 9, the overall score becomes 9 + -9 ($\Delta q_n + \Delta q_m$) or 0, which is the same score one could achieve if $\Delta q_n = 0 and \Delta q_m = 0$. In the unlikely situation of a perfect match (meaning that for every quality in Q, Template.q_n equals SuggestedBlog.q_n) the evaluation function would return a fitness score of 0. Thus it is clear that the more alike two blogs are, the closer to 0 the fitness score returned is.

4.2.2 Proofs

Lemma 1 *The evaluation function will create a score that allows the system compare the fitness of two blogs*

Proof: Assume three blogs and some large integer value M

$\forall q_n$ such that $q_n \in Template.Q; q_n = M$
$\forall q_n$ such that $q_n \in Blog1.Q; q_n = 0$
$\forall q_n$ such that $q_n \in Blog2.Q; q_n = M - 2$

When evaluating the fitness score between Template and Blog1, we see that for every quality in Q the score increases by M-0 (or just M). Thus the fitness score of Template and Blog1 is $|Q| \times M$. However, the score Template and Blog2 evaluate to is much smaller. For every quality in Q the score increases by $M - M - 2$, or just 2. Thus the score here is $|Q| \times 2$. With a much smaller overall difference, Blog2 and Template are clearly more similar. □

Lemma 2 *The evaluation function notes exactly how different two blogs are in comparable qualities.*

Proof: Assume Template and SuggestedBlog are comparing quality q_n. Template.q_n = 9 and SuggestedBlog.q_n = 8. From this, the overall fitness score is only increased by 1 (or 9-8). However, if SuggestedBlog = 2, the overall increases by 7 (or 9-2). Thus, the evaluation function not only determines if two blogs are different, but offers a measurable value into how different they are with precision. □

Lemma 3 *Assume the $|Q|$ is n where n is $\in \mathbb{N}$. Then the evaluation function will evaluate in $O(n)$ time.*

Proof: The purpose of this algorithm is to iterate through all qualities q_n in Q. Since the database is formatted in such a way that all items have a set Q of the same size with the same keys, this program only needs to iterate through Template.Q as the keys will match in both SuggestedBlog and Template. Thus, at its most complex this function will only need to iterate through each item in Q once, making the time complexity O($|Q|$), otherwise written as O(n), or linear time. □

Lemma 4 *Assuming the evaluation function runs in $O(n)$ time, it is clear that the system then the system as a whole still runs in linear time.*

Proof: From Lemma 3 it is clear that the evaluation function runs in O(n) time. The entire system of recommending a blog runs the evaluation function ten times. From this, the complexity of the algorithm runs in O($10n$) time. As n increases, 10 becomes less and less significant, and thus one can represent this time complexity as O(n) time as well. □

Theorem 1 *It will take O(n) time for the Domino Recommender System to offer a suggestion to the user.*

Proof: This should be obvious from Lemma 4. □

4.3 Receiving Feedback From the User

Once the Recommender System has presented the user with a blog, it has the opportunity to draw conclusions as to what the user values in a blog. This is done through continual explicit feedback from the user. In the user interface of this program, there are two buttons the user can press to send feedback to the system. The Like and Dislike functions change the scores to qualities in Template.Q. The Like function increases the score (and therefore the importance of) qualities where the Template and SuggestedBlog matched. Dislike, follows almost the same algorithm, with the exception that instead of increasing the score of certain qualities it decreases scores where Template and SuggestedBlog did not match. It is notable that the calculations here are adaptations of *Rocchio's algorithm* for relevance feedback (CITE). Refining Template.Q to stand for an approximation of the user's preferences, rather than simply the qualities of a blog, allows for more accurate results to be presented to the user when it comes time to suggest another Blog. The pseudocode for both the Like and Dislike functions are included below:

The Like functionality

```
for quality in Template.$Q$.keys():
        if (Template[quality] == SuggestedBlog[quality]):
                Template[quality] += 1
return Template
```

The Dislike Functionality

```
for quality in Template.$Q$.keys():
        if (Template[quality] == SuggestedBlog[quality]):
                Template[quality] -= 1
return Template
```

4.3.1 Proofs

Lemma 5 *The Like functionality will run in O(n) time.*

Proof: Like the evaluation functionality, this process iterates through all qualities q in Q, while running some trivially fast functionalities in each iteration. Thus, the most time consuming aspect of this algorithm is the linear iteration, making the overall complexity O(n). □

Lemma 6 *The Dislike function runs in O(n) time.*

Proof: It is notable that these two functionalities both iterate over the set Q with the only difference between them being why perform addition or subtraction. Thus, if the Like functionality runs in O(n) time, so will the Dislike functionality. □

Lemma 7 *There is no way for the like and dislike function to run concurrently*

Proof: It is clear from the pseudocode in listings 3 and 4 that neither functionality calls the other. Thus the system itself cannot run these two functions in anything but a serial manner. Users can only run these functions by interacting with buttons on the user interface. As each button causes exactly 1 of these two to run, there is no way for the user to cause the Like and Dislike functionalities to run at the same time in a single running of the algorithm. Thus drawing on the conclusions in Lemmas 5 and 6, there is physically no way for these two processes to run in anything greater than O($|Q|$) or linear time. □

Theorem 2 *Drawing conclusions off of a user's further feedback will take no longer than O(n) time.*

Proof: This should be obvious from lemma 7. □

5 Process

5.1 Scoping the Project: Food Blogging

In order to implement this algorithm, it became necessary to create a database of items, and populate each item with a set of qualities for this system to iterate over. Specifically I chose to examine blogs discussing food. This decision was made for several reasons. As the ultimate goal of this project was to make a functional Recommender System, it was a priority to make a relatively lightweight (yet still functional) database with which to test it it on. Considering blogging as a whole would involve developing an immensely in depth list of qualities over a large number of blogs in order to produce accurate results. Instead, I focused on a specific subgenre of blogging: food blogs. I chose to look critically at online food writing (over any other possible topic) as there has been little to no research in the subject, not to mention that it seemed like an enjoyable topic to work with.

This raises an important question: what exactly is a food blog? The genre is not easy to define. For one thing, the designation is something awarded by the writer. A blog could, technically, mention food only a few times and still call itself a food blog. So where can one draw the line between a blog that has some food writing in it and a food blog? Since no official designation exists, I have done my best to pin down this amorphous topic. The posts on a blog primarily about food act like essays on a topic involving food. It should be noted that no two blogs seem to achieve this goal in the same way. Some discuss the cultural significance of a dish, some look at the chemical reactions that a recipe undergoes on its way to being food, some persuasively convince their readers to try something new, and still others discuss their personal relationship to a dish. Regardless of how exactly they do it, food, recipes, and the act of cooking need to be dominant themes in all posts.

The criteria I used in selecting blogs for my database is even more specific than this, yet largely outside the scope of this paper. For more information on this subject, a more description of the categories I created, and how they inform the reading of a blog, please consult my paper on the subject: *Feeding the Machine:Teaching a Computer to Examine the Stylistic Differences in Food Blogs*. For the purposes of this paper, I offer then that a working definition for the term "food blog" is an online collection of essays on food, where the writing is done by a small number of contributors.

5.2 The Categories

Once I had a working list of around 50 blogs to look at, I needed to develop qualities to populate each Blog object's set Q with a number of qualities. This involved closely reading and analyzing many of the blogs I had gathered together to look for similarities in style. The sites inside the scope of this project still have a massive range of styles, focuses, tones, and voices. I have divided them up into categories (four large categories, and eight subcategories) based on the ultimate goals of the writing and how different groups of writers achieve such ideas. Again, the system as a whole can handle any kind of item with any number of categories. Should one want to look at the kinds of food each blog discusses, rather than the style of writing, it would only require changing the qualities (and corresponding scores) of each item in the database. Ultimately, Q in my system was populated with the following qualities based on writing style. Any given blog could receive a score between 0 and 9 on any quality (where 0 implied the quality was nonexistent and 9 implied it was a major component of the writer's style)

Fact Focused Writing

Many food blogs argue that they are not about food at all. Instead they use food as a lens to engage with the world on a larger scale. These bloggers tend to base their writing in facts regarding food in some respect, and derive a greater message from this information.

1. **Food News:**

   One common focus for this style of fact based food writing is to use food as a lens for looking at ethical, political and health related issues. In print, one might look to Eric Schlossers *Fast Food Nation*, or Michael Pollans *The Omnivores Dilemma* to see this style of writing. These writers look at what we choose to eat as context for tackling greater issues such as workers rights or the laws defining how our food is labeled. Online, Alton Browns Edible Examiner, or The Ethicurean (which has several contributing writers) are known for using this style. Brown writes in one of his posts: You know something is wrong with the food industry when you can buy a salad that contains more sugar than a donut. That was one finding in a recent study conducted by Credit Suisse, which revealed the sugar content of some of Americas most popular foods and drinks: (CITE). Rather than speaking directly about a dish or recipe in this passage, Brown examines a larger issue of processed foods. Brown not only tackles issues of health in America, but begins to question the complexity of the phrase good for you." This larger topic could have been discussed in several different ways, but Brown chooses to contextualize these issues through a lens of food. This is the trait common to the blogs that have a high score in the Food News subcategory in my database. They do their best to engage with the world on a political level through the discussion of food.

2. **Cultural Perspectives:**

   Another way food is used as a way to engage with the outside is through the exploration of other cultures. This subcategory, titled Cultural Perspectives in my work, focuses on bloggers that zero in on a specific regional style of food as a way to introduce a culture. Specifically, these writers gear their content towards American readers and focus on distant destinations. Leela Punyaratabandhu and Jun Belens respective blogs both employ this style to great effect, as does Anthony Bourdains writing both online and in print. A good example of this style comes from another blogger in this category, Rachel Roddy. Roddy, the writer behind *RachelEats*, another blog of this style, discusses a recent architecture tour she took in Rome: To understand something of Testaccio and the Ex-mattatoio is to understand something of Roman food or one aspect of it at least and therefore part of the story of Rome. Food as story or story as food or something akin to that. The area has been associated with food trading since ancient Roman times when it was a port and sprawl of warehouses." (CITE) Here, Roddy writes of how an understanding of Roman architecture

leads to an understanding of Roman food, and vice versa. Thus, she encourages engaging with food (specifically artichokes and lamb) as a way to interact with a culture, even if one is not currently in the same country.

3. **Culinary Education:**

   The final common avenue for fact based writing in food blogs comes in the form of culinary education. In print these writers are common names to the kitchen. Irma Rombauer of The Joy of Cooking and James Beard are perhaps two of the more known, although far from the only writers in this category. Online, blogger Adam Roberts rises above the rest with his site: The Amateur Gourmet. One of the things that sets this subcategory of blogs apart from the others is its focus on explaining technique over actual discussion of recipes. In a post discussing cranberry sauce Roberts writes: Heres all you need to know: cranberries + sugar = cranberry sauce. If you use less sugar, your sauce will be tarter. If you use more sugar your sauceDUH!will be sweeter. Everything else just adds flavor.." (CITE) Here, rather than offering a recipe, Roberts does his best to explain the basic concept behind the dish in general. More than that, he writes in a way that makes cranberry sauce seem very easy to make when he says: cranberries + sugar = cranberry sauce, arguably encouraging readers to try out their own ideas in the kitchen. In short, Roberts does something closer to the reverse of what one might be doing in either other style under the fact based category of food blogging: he recognizes that his audience is mainly comprised of those uncomfortable with cooking, as well as the idea that cooking allows for engagement beyond the plate or the self. Thus he does his best to bring readers closer to food so that they may further extend themselves.

Persuasive Writing Qualities

Many bloggers have a more specific goal in mind when presenting facts and statistics. Rather than simply introducing a new way to think about food and the greater challenges it introduces, this category of writers do their best to lead the reader to a specific conclusion and act on it.The unifying quality in this category is that writers want some sort of action, and use their blogs to convince readers why this is a good idea.

1. **Dietary Persuasion:**

   Blogs are often used to encourage a change in the eating ways of the reader. Books like *Food Matters: A Guide to Conscious Eating* or *Eat Vegan Before 6:00* by Mark Bittman exemplify this kind of writing in print, while online one might look to Hank Shaws *Hunter Angler Gardener Cook* or *Post Punk Kitchen* by Isa Chandra. On such sites recipes, statistics, and personal experiences are all shared in equal measure, but they all work toward encouraging the reader to make a change in the way they eat. Chandra sells her recipe for Vegan Thanksgiving Burgers by saying Youre basically eating a beautiful Bob Ross landscape of the prettiest autumn foliage you ever did see. But in burger form. And believe me, you will give thanks with every bite!" (CITE). Through her description, the reader is given the impression that these burgers taste even better than they look. Given that they are compared to a beautiful landscape, one can draw conclusions about their flavor. Chandra goes further and caters to a non vegan audience by offering burgers as well. It is a reminder that even if one is eating vegan, they do not need to give up the foods they love, such as burgers. Recognizing that many readers will resist changing their diet, writing in this style of post does its best to sell its recipes, and this passage does just that. This style of writing does everything it can to convince: from developing a strong narrative voice that readers would want to follow, to proving to the reader that a dish is easy to make (regardless of the actual complexity of the recipe).

2. **Political Persuasion:**

   Sometimes, persuasive writing takes a less direct focus over the message eat this not that." Often, such writing takes a political slant and often concerns itself with looking at the laws

that govern what and how we eat. While this may sound very much like the category of news on food already discussed, this style is separated by its persuasive call to action, over a simple presentation of the facts. Bob DelGrosso writes on his blog *A Hunger Artist* on a new farm bill in progress: This news should give supporters of the abolition of foie gras farming pause to reconsider the strength of their political and moral arguments. I'm not thrilled by much of what I have read of the new Farm Bill...Don't like the way a particular food is produced? Then encourage a free and vigilant press to keep people informed about what the farming industry is doing and let the people decide for themselves what they want to eat.." (CITE) Unlike the highly democratic style of food news, delGrosso is actively taking a news story and telling the reader how to respond. He argues to readers to to take a specific stance in the debate of factory farming. Yet, exactly what he is asking of the reader is slightly more complicated than what to put in their mouths. What encouraging a free and vigilant press entails isnt discussed in his writing. The exact course of action is left to the part of the reader. In short, the changes discussed in political persuasive writing tend to be much larger than a personal diet change, more abstract and focused on debates of policy

Memoir Writing Qualities

Without doubt, however, the most commonly seen of food blogs revolves around the personal. Writers like Deb Pearlman of *Smitten Kitchen*, and Julie Powell of *The Julie and Julia Project* have had wild success with this form, as demonstrated by the fact that both started as bloggers and turned into published authors. As the name suggests, the bulk of this writing in such posts centers on personal experience and thought.

1. **Diary Like Writing:**

   Pearlmans Smitten Kitchen is far from alone in the diary category. Such blogs direct much of their writing to discussing the day to day life of the writer. They report, in a fashion common to Christmas card newsletters. Molly Wizenberg of Orangette writes: I've been wanting to tell you about this soup for more than a week now, but a certain crazy-haired dancing maniac of a young person is getting a molar, or something, and has been waking up veeeerrrrry early and then spending a large portion of the day crawl-running around the house/park/bathtub/Delancey" (CITE). As Wizenberg displays in this passage, this style of writing is much less focused on telling stories, and relies on the creation and development of the authorial persona through voice, tone, and humor. Anecdotes may be used to introduce a recipe, but these are are mainly focused on the household, childcare, and recent local experiences. In short, these blogs are largely a diary. A record of the day to day life of a specific writer, that just happens to involve a great deal of food.

2. **Long Term Reflective Writing:**

   Very closely related to the diary category is another subgenre labeled Long Term Reflective." In content, these two sub genres are very similar. Both relying on the writer re-telling experiences as a way to set the stage for both a dish and a narrative. Yet, these categories are also arguably opposites. This style of writing is not limited to recent events in the way bloggers of the former category are. Any stories shared in this style are selected for how they tie back to a larger idea, rather than any reflection coming out of a desire to share a story. Kate Christensen exemplifies the memoir style in her self titled blog when she writes about butternut squash soup: While we ate big bowls of it, I reminisced about how Connie used to sit with some of us in the student lounge before workshop, the scariest, most nerve-jangling time of every week, and how comforting her presence always was for me...I remember myself as intimidated, lovelorn, uncertain of my literary aims, and painfully shy." (CITE) Like the large fact based category, food is not the ultimate focus of the piece. It is used as springboard for considering deeper topics. It is set apart, however, as instead of using food to to further introspection, rather than about things external to the self.

3. **Philosophical Writing:**

The final subgenre in this category starts and ends with personal writing, but uses personal experience to consider their lives through a larger philosophical context. Emily Ludolf of *Edible Ink* and Karen Coats of *The Rambling Spoon* explore this style to great effect in their online writing. Not quite a connection to something external, and not quite introspective writing, this genre largely looks at living, rather than life. To illustrate this idea more clearly consider one of Coats recent posts, which considers what it means to be grateful for the small things and ties it together with the story of a friends relationship with olive jelly. Ludolf explores the definition of soul food and examines where a country's soul comes from in her post titled: Less Criminal Forms of Sin: Sweet Potato and Coca Cola: the places where every glass surface is pock-marked with bullet holes but there's a din inside, not coming from the TV. These are the homes where Soul Food is cooked and shared; comfort without a shiny price tag and a mind-numbing jingle...golden hush puppies and waffles & chicken...The soul food was deep-fried, salty and mouth-wateringly rich but a hell of a lot less sinful than a Macdonalds burger and Starbucks frappe with a side of exploitation." (CITE) Here, food is the focal point of the piece, but also allows Ludolf to reflect and draw conclusions on her perceptions of the world and recent vacation.

Lyric Writing

Lynn Z. Bloom might put it best when she classifies food writing as belletristic nonfiction either devoted to food or containing significant food-related scenes." (Bloom 348) In print, this descriptive and often lyric style of food centered writing can be seen in the works of Michelle Morano and M.F.K. Fisher. Online, it is seen through bloggers like Matt Armendariz and Kate Christensen. As this style typically focuses on the experience of eating, it almost always employs sensory based writing. Five big shrimp came with their heads on, eyes broiled red, tiny arms and all (CITE), Christensen writes in one of her posts, and Armendariz explains his favorite variety of anchovies by writing that they are: slightly milder and fresher in flavor than the salt-packed variety, they always lend a sweet, tangy taste to dishes and salads." (CITE) These passages do their best to directly explain what their respective dishes look and taste like. Texture and smell are also often discussed. Beyond simple telling, however, this style often focuses much more on the universal experience of enjoying a meal. Christensen writes much more on her personal feelings while eating when she writes: We grunted like Dingos as we ate them. Five minutes later the plate was practically licked clean." (CITE) Stephanie Stiavetti, of the blog *The Culinary Life*, employs alliteration in her writing, in passages like: Topped with chunky cubed sourdough for crunch, this casserole is more than deliciousits sinful." (CITE) More than flavor, which can only be described in a limited amount of ways mainly ending in ly, these writers create a larger picture with their writing. Christensen captures the experience of eating as a primal force through her image of grunting dingos that licked the plate clean. Stiavettis lyric description use the percussive sound of a hard c in the sentence: chunky cubed sourdough for crunch, this casserole is more than delicious to generate the sense of a crisp, crunchy texture. Thus these writers are then using elevated literary styles to convey their experience of eating.

5.3 Creating the System

Once I had a list of 45 items and categories detailed enough to draw conclusions off of, the next step was to start developing an algorithm. After doing some research into existing algorithms, I remembered an assignment from my Operations Research class in 2011. That assignment was to create an optimal matching of 200 reservations to hotel rooms based on a list of 40 room preferences . Many other teams of students implemented simulated annealing algorithms or genetic algorithms to accomplish this, but my team developed our own algorithm. Each reservation was initially assigned a random room, then each room's fitness level was calculated. If a fitness score failed a certain threshold, it would look for another reservation to swap with. If the fitness level for such a swap is better than the current fitness level, a swap is made. Fitness scores would be

calculated again, and more reservations continually swapped as necessary.

I adapted this idea to create the suggestion algorithm. I calculated a fitness score in a similar manner, and used it to not so much determine "good" matches from "bad", but quantify how good a match was based on an initial set of preferences. However, instead of looking for an optimal solution (which would decrease the diversity of results) I merely looked for the best of a small, randomly selected subset of the database.

After that, I needed a way to draw conclusions about what the user was really looking for in a blog. Drawing from the ideas of many other Recommender Systems, I decided to include "Like" and "Dislike" buttons for the user to offer feedback about a given suggestion. One can assume that the qualities that are more key to the blog are more important to the reader as well. This assumption is based on the idea that one will only ever read something that interests them. For example, it would be foolish to read a blog focused on persuading readers to eat a vegan diet if one could not tolerate reading about veganism. Thus the Like function should increase the score (and therefore the importance) of matching qualities, and the Dislike decrease their score. Over time, this would allow the scores corresponding to the template blog to mirror the user's preferences, rather than simply the qualities of a blog.

5.4 Testing the System

Once I had a system I *believed* to function, it became time to put it to the test. I would enter a Blog and track how different suggested Blogs were based on quality. In later tests I would input a blog and attemt to teach the system my preferences until the blogs it suggested did not match the initially entered blog's qualities at all. Thus, I was able to see that it would both suggest Blogs that were similar to the initial user preferences, but was capable of learning and changing beyond those scores.

6 A Simple Run Through

Ideally this will be done in TeX, but TeX is being stubborn. Therefore, know that there will be some A Blog's set Q might look like this if graphed:
GRAPH
Here, we see each score and corresponding quality plotted as a point and connected to form a line. One of the first tests I ran just returned 10 blogs, running the LIKE function after every return.
GRAPH HERE
Another more complex test involved disliking any blog with a score less than 7 in the International Culture section
GRAPH
Clearly from these images, peaks in the graphs (the high scoring qualities in each Blog) are roughly matched, and if anything emphasized. (ALAN: Ideally this will be done in TeX, but TeX is being stubborn. Therefore, they're attached, but in a separate document)

7 System Analysis

7.1 The Issue Of Generating Diversity

In this system when running the algorithm to return suggest a blog, 10 blogs are chosen at random (or at least pseudo-random) and compared via the evaluation function to determine which is the best fit when compared with the Template blog. It is this random aspect that ensures diversity. Even with the same initial Template blog, the system is in no way guaranteed to produce the same 10 blogs to compare. Instead, the system will always do the best it can with the subset available to it, producing different results every time. As the user likes and dislikes what is returned, the scores of Template will alter accordingly. Thus, a blog that might initially be a good fit may never be suggested based on the Like and Dislike functions running. Even with a database of only 45 items, no two test runs have returned the exact same list of blogs.

7.2 The Cold Start Problem:

Overall, the system has no infrastructure in place to *ensure* that the Cold Start Problem does not become an issue. How detailed the database of items, qualities of each item, and range of score all determine how useful the information The Domino System returns is. Yet, the system has no way of ensuring the data provided is sufficient to be useful. From this, it's difficult to say whether the Cold Start Problem would always be an issue when working with this system. However based off the information from section 5.2, one can assume that for the purposes of this discussion, then the the depth of detail (seen as the number of qualities and corresponding scores in Q) is adequate enough to make informed decisions.

7.3 The Issue Of Grey Sheep:

Through this system, grey sheep should never be a concern. As the scores in Template.Q are always changing, there is always a chance certain blogs (which may initially have had mediocre scores) can be reached and even be optimal. Furthermore, given that the system draws 10 blogs out of the database at random, even a grey sheep has a chance to be the most optimal out of all 10 drawn.

7.4 The Issue Of Correctness

From Lemmas 1 and 2, it is clear that the evaluation function returns an accurate measure ΔTemplate.Q and SuggestedBlog.Q. Thus, the best blog (meaning the blog with a score closest to 0) over 10 iterations of random blogs is the most like the Template blog out of 10. Furthermore, as the system runs, the quality scores in Template.Q (which initially are only an approximation of a user's preferences) become a more and more accurate mapping of the user's preferences. Thus, with the knowledge that these scores are as as accurate as they can be at any given time, it will always return as good a fit as it can at any point.

7.5 The Issue Of Speed:

As we have seen from the proofs in previous sections, no component of this system will run in longer than O($|Q|$) time. To demonstrate what an improvement this is, consider Pandora's Music Genome Project. The Music Genome Project, using the system outlined in Section 3 takes an average of 23 minutes to draw a relationship between 2 songs (CITE). Granted the items Pandora's system works with have between 150 and 500 qualities (CITE), which is significantly more complex than small database I've implemented I've described in this paper. However, assuming each iteration of the evaluation system I've presented takes 1 second, comparing two songs with my system would take around 8.3 minutes, or nearly a third of the time!

7.6 Restrictions

As discussed above, the Cold Start Problem is not guaranteed to be resolved by this system. Thus the system is very much restricted to having an adequately detailed database of items to draw from. If the user does not enter their preferences on blogs (meaning they do not run the Like and Dislike functions) the algorithm is limited as to how accurately it can meet the user's desires.

Also, there is no inherent algorithm in place to automate changes in this database. Every item and corresponding set Q in my implementation is stored in an XML file and manipulated thusly. This issue is not new in the world of Recommender Systems. Indeed some companies like Pandora and their competitor Songza both employ scores of people to classify the music that is entered into their databases. Songza even takes pride in the fact that their service does not automate their suggestions. As one article puts it, "the company started with the small objective of recruiting 25 experts to make playlists. They did that to see how the product felt with real content in the system. The expert playlists became so popular that Songza quickly made that the company's mission: Build the greatest collection of expertly-curated playlists the world has ever seen" (CITE). While this business model clearly has worked for these successful companies, it seems feasible that a system should be able to automate the entry of new items into a database. Thus while my suggestion algorithm is fast, the work involved in creating a usable database is still significant.

8 Significance of The Domino Recommender System

Perhaps one of the most significant points of my research into Recommender Systems is that it appears new to the world of content based filtering. While the fitness scores my evaluation algorithm calculates approximate a Correlation Coefficient, they are also not the same thing. Correlation Coefficients are used to compare users based on their preferences. When a similar user is found, the issue of determining a suggestion is relatively straight forward: determine what User1 enjoyed that User2 has not viewed or rated. However, as these metrics can only be calculated between two users at a time, clustering algorithms are typically employed in such systems as well. Not only does this increase time and complexity, ensuring diverse information becomes quite an issue. In my system, neither time nor diversity have been shown to be problematic.

It is also notable that this system is quite fast. Rather than needing to iterate over a very large set of users or items for the purposes of clustering like objects for fast comparison, this system iterates over only a random subset of constant size. Instead, time will increase as $|Q|$ increases. In most environments where Recommendation Systems are useful, $|Q|$ is not just less than but dwarfed by the size of items or users as a whole. In my implementation, a sample size of less than a quarter of the database is taken with every request for new data. While 10 items might be adequate for me, however, a larger percentage would need to be taken from Pandora's database of 900,000 songs. However, as this sampling will always be a subset of the data as a whole, it should remain constant with respect to every implementation of the database, increasing only as the database grows in some significant order of magnitude.

As for the relevance of Recommender Systems as a whole in the world of Computer Science, it is useful to look at a recent example. In 2006, Netflix opened up a challenge to the world, offering $1,000,000.00 to anyone who could improve their Recommender System for movies (called CineMatch) by 10% (CITE). A 2007 article in the *The New York Times* quoted the CEO of Netflix, Reed Hastings, saying "getting to 10 percent would certainly be worth well in excess of $1 million to the company. The competition was announced in October 2006, and...30,000 hackers worldwide are hard at work on the problem" (CITE). It took this immense following of "hackers" 5 years to increase CineMatch's accuracy by 10%, with the award finally being given in 2011 (CITE). The

importance of an accurate and fast Recommender System is clear from this interview. Accurate systems, beyond the complex problems they introduce to both the fields of optimization as well as artificial intelligence, are worth millions of dollars. Furthermore, the question of further optimization will always be up for debate.

9 Future Work

As I have mentioned in this paper already, it seems entirely feasible that items could be added to the database without human classification. Many existing content based systems take a great deal of pride in the fact that they do not automate the classification of their databases, yet to me this seems odd. If an algorithm could achieve this with equal accuracy, why pay workers to classify at most 100 items a day (CITE)? In future, I'd like to look more into Latent Dirichlet Allocation and Autotagging to achieve this result while maintaining accuracy in classifications. Furthermore, I would like to explore the idea of creating a hybrid filtering system, rather than simply a content based one without reducing the speed of my system.

10 Reflection

In 8th grade my family and I moved to Sydney Australia for three and a half months. When we came back people bunched around me, faces bright and excited, and they'd start to ask questions.

How was your trip?
Did you have fun?
What was it like there?

Later, when I came home from college the first few times, I'd again be asked for the cliff notes of how I'd been and what I had been up to for the past few months. I still have no idea how to answer these questions. In conversation, I'll pick one word and pretend like that sums up the whole thing. "It was great", I say and share a funny anecdote. In truth it was great and it was nothing special, it was heaps of fun and at times I just wanted to go home. Three months of doing things can't really be summed up in a sentence. There's just too much to say. So how do I start this section off? There were days when this project would get me out of bed in the morning, and days when I would have rather just stayed under the covers. Suffice it to say that I learned. I learned and learned and learned.

As it might be expected, I learned in the academic sense. I knew nothing about Recommender Systems before starting out on this project, and only a little about food writing. The 45 total pages of work I've produced only discuss most of the relevant information I've found. I've read posts in something in the neighborhood of 100 food blogs too, and now am eternally ready a game of trivial pursuit. I now know how to bone and butcher squirrels, how to make a truly great cup of Chai, and the sexism seen in Japanese sushi restaurants.

I learned how to think critically and speak confidently about my project. Many don't see food blogging as a subject with enough academic heft, when there's plenty of evidence out there to show how untrue that is. Still, when your elevator pitch for a project is: "I'd like to make a Pandora–type application for food blogs" you learn to accept the dubious looks. More than once I've been told my project doesn't make much sense. At first, I would deflate and change the subject, but through practice I learned how to take it in stride and prove that this work is important.

I learned how to take a red pen to something I'd called "my best work"; that the phrase "I'll get up early and finish it" is hardly ever true. This project taught me how to work for my own approval, rather than for a grade or the approval of my professors. I had to teach myself how to stick to a schedule, and how to own up to my mistakes when I missed deadlines. I learned how still and how dark the air is at 4 in the morning, the variety of surfaces I can fall asleep on, and the unspoken understanding of your peers when you say "I can't, I need to go work on my SMP".

Looking back, I would love to say I would do things differently. Part of me would have been happier doing a strict Computer Science project. Overall, the days when I woke up and got straight to work were (more often than not) days I was working on the computer science half of my project. This is not to say the English section was not enjoyable. I had good days doing my close readings of food blogs too. In general, though, the bigger puzzles and challenges came from creating and testing this algorithm. This SMP has taught me where my interests are and what following those interests entails. I came into college firmly believing I'd never enjoy anything as much as studying literature, and now I spend my time filling out applications for jobs with titles like: "software developer" and "data scientist".

I'm immensely proud of the work I have done as a whole. I started with a vague idea and came out the other end with definitive work. Granted, in a perfect world I would have paced myself a little better. Due to timing and miscommunication issues, I ended up doing the bulk of my development work at the same time I was composing and defending my categories. Still, I feel like the research process has been immensely beneficial to my growth as a student, a reader, and a developer.

11 Conclusion

In conclusion, for my St. Mary's Project I examined the up and coming field of Recommender Systems and their significance today. In doing so I created my own system as well as a detailed database to accompany it. This new system draws thoughtful conclusions and does its best to recommend food blogs a user might want to read based off of their preferences. Moreover, the system I devised is not only fast, but appears to have gone unpublished until this time. I'm looking forward to pushing on with this work and seeing what new challenges there are to overcome with it.

References

[1] The inner workings of the music genome project. http://blogs.cornell.edu/info2040/2012/09/23/the-inner-workings-of-the-music-genome-project/, bibsource = http://www.interaction-design.org/references/.

[2] Music genome project. http://en.wikipedia.org/wiki/Music_Genome_Project, bibsource = http://www.interaction-design.org/references/.

[3] Recommender systems. http://recommender-systems.org/content-based-filtering/, bibsource = http://www.interaction-design.org/references/.

[4] The similarity search wiki homepage. http://sswiki.tierra-aoi.net/index.php?title=Main_Page, bibsource = http://www.interaction-design.org/references/.

[5] Matt Armendariz. Kale salad with ricotta salata, pine nuts and anchovies. http://mattbites.com/2013/11/12/kale-salad-with-ricotta-salata-pine-nuts-and-anchovies/, bibsource = http://www.interaction-design.org/references/.

[6] Dipartimento Di Elettronica Informazione E Bioingegneria. A tutorial on clustering algorithms. http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/index.htm.

[7] Isa Chandra. Stuffed thanksgiving burger. http://www.theppk.com/2013/11/stuffed-thanksgiving-burger/, bibsource = http://www.interaction-design.org/references/.

[8] Kate Christensen. But you can find me, when the light is changing, at that time of day when there's little day remaining. http://katechristensen.wordpress.com/2013/10/12/but-you-can-find-me-when-the-light-is-changing-at-that-time-of-day-when-theres-little-day-remai bibsource = http://www.interaction-design.org/references/.

[9] Kate Christensen. Oh, how we danced and we swallowed the night. https://katechristensen.wordpress.com/page/2/, bibsource = http://www.interaction-design.org/references/.

[10] Bob del Grosso. New farm bill may flip ca ban on foie gras. http://ahungerartist.bobdelgrosso.com/2012/07/new-farm-bill-may-flip-ca-ban-on-foie.html, bibsource = http://www.interaction-design.org/references/.

[11] Norman Fenton and Martin Neil. Risk assessment and decision analysis with bayesian networks. http://www.eecs.qmul.ac.uk/~norman/blog_articles/p_values.pdf.

[12] Erin Griffith. Songza's founders realized they weren't thinking radically enough here's how they changed that. http://pando.com/2012/08/15/songzas-founders-realized-they-werent-thinking-radically-enough-heres-how-they-changed-that/, bibsource = http://www.interaction-design.org/references/.

[13] Michael Howe. Pandora's music recommender. http://courses.cs.washington.edu/courses/csep521/07wi/prj/michael.pdf, bibsource = http://www.interaction-design.org/references/.

[14] M. Tim Jones. Recommender systems, part 2: Introducing open source engines. http://www.ibm.com/developerworks/library/os-recommender2/os-recommender2-pdf.pdf, bibsource = http://www.interaction-design.org/references/.

[15] Emily Ludolf. 'less criminal forms of sin' : Sweet potato and coca-cola. http://emilyludolf.blogspot.com/, bibsource = http://www.interaction-design.org/references/.

[16] Adam Roberts. Cranberry sauce 101. http://www.amateurgourmet.com/2013/11/cranberry-sauce-101.html, bibsource = http://www.interaction-design.org/references/.

[17] Rachel Roddy. the other quarter. http://racheleats.wordpress.com/2013/10/26/the-other-quarter/, bibsource = http://www.interaction-design.org/references/.

[18] Stephanie Stiavetti. Gruyre and emmentaler macaroni and cheese casserole with ham and cubed sourdough. http://www.theculinarylife.com/2013/gruyere-emmentaler-macaroni-cheese-casserole-ham-cubed-sourdough/, bibsource = http://www.interaction-design.org/references/.

[19] Clive Thompson. If you liked this, you're sure to love that. http://www.nytimes.com/2008/11/23/magazine/23Netflixt.html?_r=1&partner=permalink&exprod=permalink&_r=0.

[20] Eberhard Karls Universitat Tubigen. Information retrieval: Lecture 8 - relevance feedback and query expansion. http://www.sfs.uni-tuebingen.de/~parmenti/slides/slides9-1x4.pdf.

[21] Jeffrey D. Ullman and Anand Rajaraman. Mining of massive datasets, chapter 9: Recommendation systems. http://infolab.stanford.edu/~ullman/mmds/ch9.pdf, bibsource = http://www.interaction-design.org/references/.

[22] Molly Wizenberg. But the soup. http://orangette.blogspot.com/2013/11/but-soup.html, bibsource = http://www.interaction-design.org/references/.

[**?**] [5] [18] [12] [1] [9] [15] [8] [22] [10] [7] [16] [17] [**?**] [2] [13] [21] [4] [14] [3] [19] [20] [6] [11]

A Annotations On Blogs In Database

B Quality Scores In Each Blog

(ALAN: A few of these headings are outdated, but by and large my database is on the next page)