Co-selling recommendations from an eCommerce dataset.

Pawan Rewari – SUID 01421653 – dadrewari@gmail.com
CS224W – Project Milestone Update

1.0 Introduction/Motivation/Problem Definition

A very simple customer use-case of eCommerce is as follows. Lets say Jane wants to buy a specific product (a Kielty 2 person backpacking tent) or one of a category of items (a tent). Typically she will go to the web or an eCommerce site and do a search. Then she may research the product and after several visits either the same day or over several days or weeks eventually converge to a purchasing decision. Each such interaction is an opportunity for the seller to showcase the specific product she is looking for and show other products that she is likely to be interested in. From a retailer perspective they want to keep the products showcased to the customer relevant to their purchase interest, and they want to provide an interesting variety of products to showcase.

The overall problem we set out to research is how can one provide a new and novel approach to selling more products at an eCommerce site. The real world application is a data science team, as consultants, who are approached to provide strategies to sell more products and given a slice of historical data – and asked to work with what they are given.

While this project is focused on an eCommerce dataset from Amazon (co-selling products) we would like to extrapolate the findings to inform any eCommerce retailer on the value of such network and data analysis. Specifically, if we are given a partial dataset to work with, as is the case with our dataset and no customer information beyond this aggregated data.

We have been provided with a dataset that includes:

- 548552 products (including 5868 discontinued products)
- for each product relevant information includes:
    - Id: nodes or vertices numbered from 0 to 548551
    - ASIN: Amazon Standard Identification Number for a product
    - Group: Book, Video, DVD, Music (the primary groups)
    - SalesRank: computed by Amazon as a function of historical sales, based on number of products sold; a lower rank signifies a higher selling product
    - Similar: products that have co-sold (Similar == Co-Selling)
    - Categories: the categories this product belongs to, can be 10 deep

The initial exploratory analysis of the dataset revealed that there were at most 5 co-selling recommendations for any product (the "Similar" products) and this was only for 60% of the products in the dataset, for the rest none or fewer recommendations (see Figure 1 next page).

Now as stated in the introductory use case, if you look at a typical customer's behavior, they are likely to visit an eCommerce website several times before they consummate a purchase. Each such visit is an opportunity to sell the product and recommend similar ones and you want to have a variety of products to recommend, one to five is not enough we need more, many more, Therein lies the core problem statement that we address in this paper. How do you to provide an added set of recommendations that are useful, relevant and on average take the retailer to a better place than they started out with.

So if we look at the basic distribution of recommended/similar products provided by the initial dataset we have:

Discontinued	5868	1%
0 Similar	163591	30%
1 Similar	11071	2%
2 Similar	10808	2%
3 Similar	10275	2%
4 Similar	9482	2%
5 Similar	337457	62%
Total	548552	100%

So the problem statement became very clear with three objectives:

1. provide a recommendation for a larger number of "Similar" products so you can provide fresh data on multiple visits by the shopper (metric – more recommended products)
2. increase the "relevance" of the added similar products with stronger "categorical similarity" – explained in detail in Section 3, briefly it is the depth of similar categories for 2 products (metric – raise the avg categorical similarity of recommended products)
3. improve the average SalesRank of the recommended set (metric – lower the avg SalesRank of recommended products)

This translates to 3 measures that we call the 3 **CoSellingMetrics.** We create a baseline with the initial dataset and try to improve upon each of these measures.

In the real world we would run actual experiments to see if these recommendations actually improve sales and then iteratively improve the recommendations, in our case since we can't do real trials we are measuring performance relative to the baseline established by the 3 **CoSellingMetrics**. Going back to the simple use-case, we would like to recommend 5 products for CoSale at every visit and then recommend other products on new visits, but we want to do this from a limited set of Similar products that we draw from, so while the user sees different products recommended, they also see the same ones recycle back on subsequent visits – ones they may be interested in. So we set a reasonable target for the recommended products to be ~15, this is relatively arbitrary but makes sense for the use case we have here.

2.0 Related Work

As stated in the References we did a literature review of several papers on Collaborative Filtering, Clustering Methods and Item-to-Item Collaborative Filtering. Here is a summary.

A traditional **collaborative filtering** algorithm represents a customer as an N-dimensional vector of items, where N is the number of distinct catalog items. The components of the vector are positive for purchased or positively rated items and negative for negatively rated items. To compensate for best-selling items, the algorithm typically multiplies the vector components by the inverse frequency (the inverse of the number of customers who have purchased or rated the item), making less well-known items more relevant. For most customers, this vector is extremely sparse. The algorithm generates recommendations based on a few customers who are most similar to the user. It can measure the similarity of two customers, A and B, in various ways; a common method is to measure the cosine of the angle between the two vectors. The algorithm can select recommendations from the similar customers' items using various methods as well, a common technique is to rank each item according to how many similar customers purchased it.

In **cluster models** to find customers who are similar to the user, divide the customer base into many segments and treat the task as a classification problem. The algorithm's goal is to assign the user to the segment containing the most similar customers. It then uses the purchases and ratings of the customers in the segment to generate recommendations. Because optimal clustering over large data sets is impractical, most applications use various forms of greedy cluster generation. These algorithms typically start with an initial set of segments, which often contain one randomly selected customer each. They then repeatedly match customers to the existing segments, usually with some provision for creating new or merging existing segments. For very large data sets — especially those with high dimensionality — sampling or dimensionality reduction is also necessary. Cluster models have better online scalability and performance than collaborative filtering because they compare the user to a controlled number of segments rather than the entire customer base. However, recommendation quality is low.

Rather than matching the user to similar customers, Amazon's **item-to-item collaborative filtering** matches each of the user's purchased and rated items to similar items, then combines those similar items into a recommendation list. To determine the most-similar match for a given item, the algorithm builds a similar-items table by finding items that customers tend to purchase together. They build a product-to-product matrix by iterating through all item pairs and computing a similarity metric for each pair based on the cosine of the two vectors. Given a similar-items table, the algorithm finds items similar to each of the user's purchases and ratings, aggregates those items, and then recommends the most popular or correlated items. This computation depends only on the number of items the user purchased or rated.

As stated in Section 1, with our dataset we have information on products, co-selling relations & categories, but not on customers. This may very well be the case in a real world situation where you are consulting for a customer who provides you with a limited dataset. Since we are not provided with customer purchase data, only aggregate data, we could learn from but not directly apply the above ideas. We came up with an approach that allowed us to make good recommendations by improving on the three **CoSellingMetrics** stated earlier.

3.1 Approach

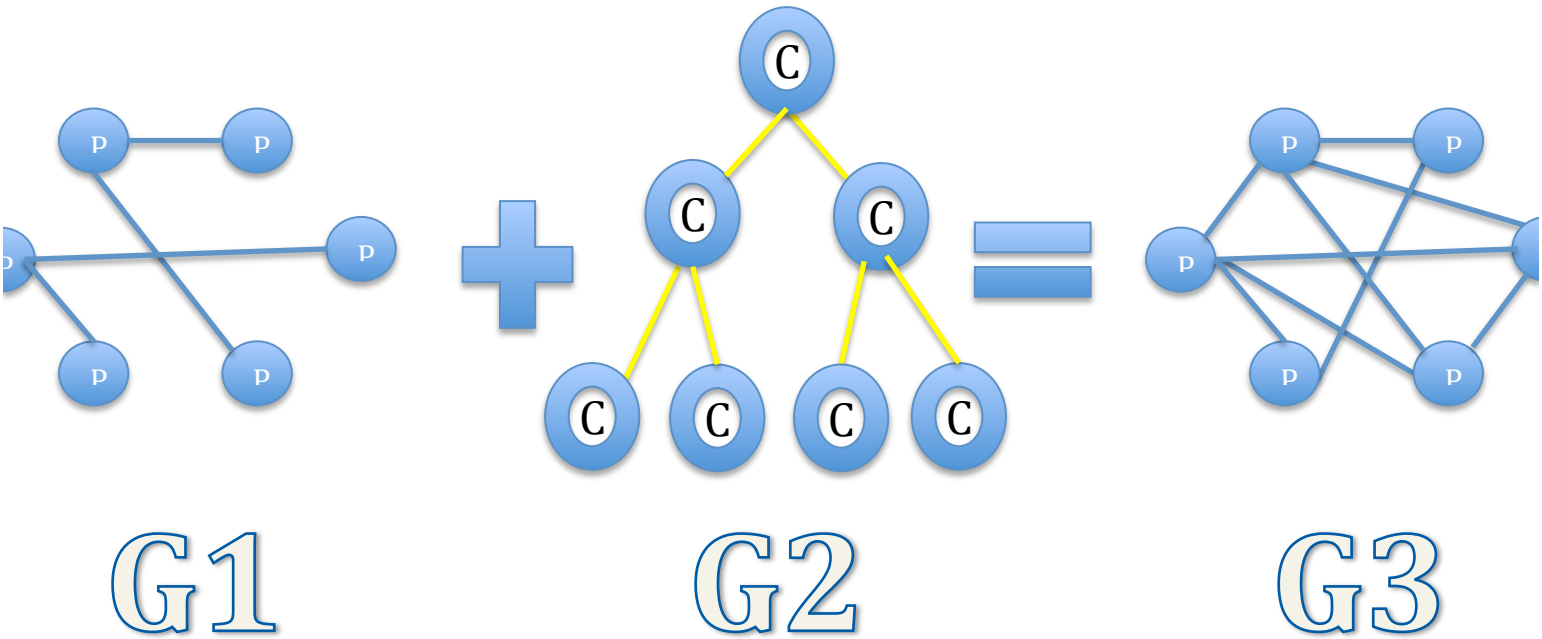
We start with an initial sparse set of recommendations (represented in G1), a tree of categories and sub-categories (G2) and use these to come up with a denser set of recommendations (G3) as shown.

The 3 Graphs (Figure 1) relevant to our approach & analysis are:

- G1: G(V,E), each Vertex is a product and each Edge represents a co-selling relationship.
  This is an undirected graph (if I sell with you, you sell with me).
  Each Vertex has an ASIN, Group, SalesRank.
  Edges to all Similar/co-selling products.
- G2: (V, E) is the category tree.
  The Vertices are all the categories & sub-categories. The root node is product, next level is Books, Music, Video, DVD, their sub-categories at the next level and so on.
  The Edges indicates a category to sub-category relationship. Each product then belongs to multiple category paths as illustrated on the next page.
- G3: G(V,E) is built on top of G1 and uses the information from G2, it contains a significant number of added edges representing newly recommended co-selling relationships by our recommendation engine described in 3.2.

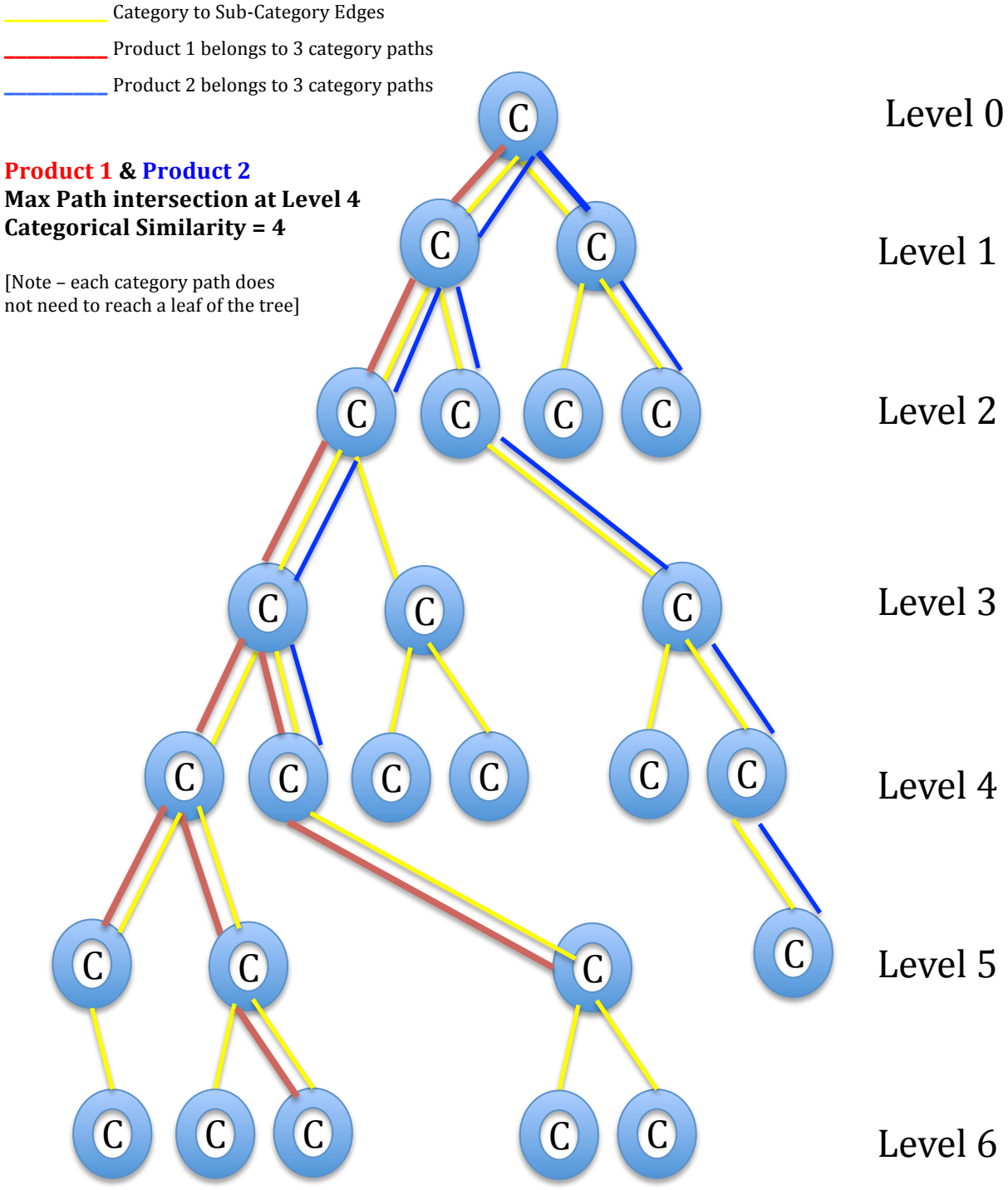
Metrics for initial Graph G1:
G1: Nodes 542684, Edges 987903
G1: Average Degree: 1.82
G1: MaxWcc: Nodes 334852, Edges 925823
G1: Diameter: 43
G1: Average Clustering Coefficient: 0.274496

Figure 1: G1, G2 & G3



Baseline and what we set out to measure. The three **CoSellingMetrics** are based on:
1: Avg Recommended Products – the number of Similar products (avg Degree)
This is simply the #edges (degree) of each node, averaged across all nodes
2: Avg Categorical Similarity – the Categorical Similarity average of the recommended products.
This is explained below
3: Avg SalesRank – average SalesRank of the similar products that are recommended
Each product (node) has a SalesRank, so we average the SalesRank of all it neighbors
The objective is to increase 1 (Co-Selling) & 2 (relevance) and decrease 3 (low SalesRank is better).

Figure 2: Partial tree of G2:



Average Categorical Similarity:

As the real example below shows each product (**ASIN: 0486220125**) belongs to a number of category paths, each product it is similar to (**ASIN: 0486401960** & **ASIN: 0486229076** & **ASIN: 0714840343**) also belongs to a number of category paths. We look for the highest Category Level that is common between the "Subject" and the products that is "Similar To" … we then average this across all "Similar products".

ASIN: 0486220125
title: How the Other Half Lives: Studies Among the Tenements of New York
Similar to: = Co-Purchasing Relationship **ASIN: 0486401960** **ASIN: 0486229076** **ASIN: 0714840343**

Category Levels	Categorical Similarity	Category Paths the product (ASIN) is listed in 1	2	3	4	5	6	7
Subject		Books[283155]	Subjects[1000]	Arts & Photography[1]	Photography[2020]	Photo Essays[2082]		
ASIN: 0486220125		Books[283155]	Subjects[1000]	History[9]	Americas[4808]	United States[4853]	General[4870]	
		Books[283155]	Subjects[1000]	History[9]	Jewish[4992]	General[4993]		
		Books[283155]	Subjects[1000]	Nonfiction[53]	Social Sciences[11232]	Sociology[11288]	Urban[11296]	
		[172282]	Categories[493964]	Camera & Photo[502394]	Photography Books[733540]	Photo Essays[733676]		
Similar to Subject	6	Books[283155]	Subjects[1000]	History[9]	Americas[4808]	United States[4853]	General[4870]	
ASIN: 0486401960	3	Books[283155]	Subjects[1000]	Nonfiction[53]	Current Events[10546]	Poverty[10576]	General[10578]	
	6	Books[283155]	Subjects[1000]	Nonfiction[53]	Social Sciences[11232]	Sociology[11288]	Urban[11296]	
	4	Books[283155]	Subjects[1000]	Arts & Photography[1]	Photography[2020]|	History[2032]		
Similar to Subject	4	Books[283155]	Subjects[1000]	Arts & Photography[1]	Photography[2020]	Travel[2087]	United States[2099]	General[21
ASIN: 0486229076	4	Books[283155]	Subjects[1000]	Arts & Photography[1]	Photography[2020]	Travel[2087]	United States[2099]	Mid Atlantic[21
	5	Books[283155]	Subjects[1000]	History[9]	Americas[4808]	United States[4853]	State & Local[4872]	
	4	[172282]	Categories[493964]	Camera & Photo[502394]	Photography Books[733540]	Travel[733684]	United States[733708]	General[73
	4	[172282]	Categories[493964]	Camera & Photo[502394]	Photography Books[733540]	Travel[733684]	United States[733708]	Mid Atlantic[73
Similar to Subject	4	Books[283155]	Subjects[1000]	Arts & Photography[1]	Photography[2020]	Photographers, A-Z[2033]|	General[2050]	
ASIN: 0714840343	5	Books[283155]	Subjects[1000]	Arts & Photography[1]	Photography[2020]	Photo Essays[2082]		
	4	[172282]	Categories[493964]	Camera & Photo[502394]	Photography Books[733540]	Photographers, A-Z[733566]	General[733568]	
	5	[172282]	Categories[493964]	Camera & Photo[502394]	Photography Books[733540]	Photo Essays[733676]		

To illustrate this in detail. The Subject (ASIN: … 0125) is listed in 5 category Paths. It has a co-selling relationship (Similar to) with ASIN: … 1960 which is listed in 3 Category Paths, with ASIN: … 9076 which is listed in 6 Category Paths, with ASIN: … 0343 which is listed in 4 Category Paths. If I examine the first Category Path of each of these, they intersect with the Category Paths of the Subject at Levels 6, 4 and 4 respectively. We continue this process and compute all the path intersections (in Yellow) and take the Max Categorical Similarity in each case (in Green).

Average Categorical Similarity [for node with ASIN: … 0125]
= Average of (max) Similarity with ASINs: 1960 9076 0343 = (6 + 5 + 5) / 3 = 5.33

BASELINE Metrics for G1:

	Original Dataset
Whole Graph	
Avg Recommended Products	1.82
Avg Categorical Similarity	3.25
Avg Sales Rank	81729
Subset - All Products with 1 or more co-selling products	
Avg Recommended Products	2.69
Avg Categorical Similarity	4.81
Avg Sales Rank	67662
Books	
Avg Recommended Products	2.75
Avg Categorical Similarity	4.85
Avg Sales Rank	84674
Music	
Avg Recommended Products	2.43
Avg Categorical Similarity	4.92
Avg Sales Rank	26739
Video	
Avg Recommended Products	1.95
Avg Categorical Similarity	3.56
Avg Sales Rank	7196
DVD	
Avg Recommended Products	3.43
Avg Categorical Similarity	5.03
Avg Sales Rank	8430

3.2 Model/Algorithm/Method

- formal description of any algorithms used

Algorithm followed:

Create G1(V,E): Read the initial dataset and
each Vertex is a product and each Edge represents a co-selling relationship
for each Vertex – ASIN, Group, SalesRank ; Edges to Similar co-selling products

Create G2(V, E): a category tree with sub-categories at each sub-level.
Map each product to the category paths it belongs to (typically 3-7 paths).
The paths are provided by the dataset.

Compute:
Avg Recommended Products is the #edges (degree) of each node, averaged across all nodes
Avg SalesRank is the average of the SalesRank of all of its neighbors.
Categorical Similarity for any 2 products is the highest Category Level they have in common
The Avg Categorical Similarity is average Categorical Similarity across all the products that are in a co-selling relationship with a specific product.

Create G3 (V,E): Start with G1. For each product – develop a list of additional "similar" products by looking at ones with the highest Categorical Similarity with the product, and amongst the ones with the same Categorical Similarity have a preference for products with a lower SalesRank, stop after ~15 or so (a "reasonable" target we established in Section 1). Add these edges to G3.

To identify the additional products to create edges to:

(a) For products (nodes) which have one or more "similar" products we follow the path of the neighbors and their neighbors … and evaluate the ones to add based on higher Categorical Similarity and lower SalesRank.
(b) For all products in addition, and for ones with ZERO "similar" products this is the only way to add more recommendations, look for products that are in the same categories and add based on higher Categorical Similarity and lower SalesRank.

Re-Compute
Avg Recommended Products, Avg Categorical Similarity & Avg SalesRank and compare with initial results. Ideally we now should have a much denser graph of co-selling relationships with more relevance and lower SalesRank.

4.0 Results and Findings

So if we look at the example we started with, the new set of "similar" products are:

ASIN: 0486220125	title: How the Other Half Lives: Studies Among the Tenements of New York
	New set of "similar" products.
ASIN: 0486401960	title: The Battle with the Slum
ASIN: 0486229076	title: Old New York in Early Photographs
ASIN: 0714840343	title: Jacob Riis (55)
ASIN: 0451628810	title: The Federalist Papers
ASIN: 0395914787	title: American Vintage : The Rise of American Wine
ASIN: 1841762121	title: Matchlock Musketeer 1588-1688 (Warrior 43)
ASIN: 0609608363	title: Wondrous Contrivances : Technology at the Threshold
ASIN: 0807120820	title: Freedom's Lawmakers: A Directory of Black Officeholders During Reconstruction
ASIN: 0060004436	title: Power Plays: Win or Lose--How History's Great Political Leaders Play the Game
ASIN: 0807841625	title: The Enduring South: Subcultural Persistence in Mass Society
ASIN: 0805072845	title: Project Orion: The True Story of the Atomic Spaceship
ASIN: 0811717003	title: Street Without Joy
ASIN: 0813911311	title: The Religious Life of Thomas Jefferson
ASIN: 0801486955	title: Going Native: Indians in the American Cultural Imagination
ASIN: 0029024803	title: An Economic Interpretation of the Constitution of The United States

First, just taking a casual look at this list, it looks like a reasonable set of recommended products. Many are in the theme of the original product, and several take you to new, but related places. So every time a user visits the eCommerce site to purchase they will be shown a subset of "people who looked at this also looked at this", "people who purchased this, also purchased this".

As we look at the transformation from G1 to G3, for this ONE product we have the new metrics:

- starting with 3 (pg5) we now have 15 "similar" products (better)
- the average categorical similarity goes from 5.33 to 5.72 (better)
- the average SalesRank from 209144 to 299227 (worse)

For the overall graph of similar/co-selling products, we clearly have a transformation to a much denser graph with a significantly larger number of recommendations per product.

Metrics for initial Graph G1:
G1: Nodes 542684, Edges 987903
G1: Average Degree: 1.82
G1: MaxWcc: Nodes 334852, Edges 925823
G1: Diameter: 43
G1: Average Clustering Coefficient: 0.274

Metrics for new Graph G3:
G3: Nodes 542684, Edges 7352419
G3: Average Degree: 13.55
G3: MaxWcc: Nodes 524997, Edges 6426637
G3: Diameter: 11
G1: Average Clustering Coefficient: 0.569

Now if we look into the details, it clearly shows that we have significantly raised the number of recommended products while simultaneously improving average categorical similarity (relevance) and average SalesRank (rank based on historical sales) of the recommended products. This

involved significant programming with much new learning, the results are solid and we have confidence in the findings.

NEW METRICS:

	Original Dataset	New Recommendations	Improvement	
Whole Graph				
Avg Recommended Products	1.82	13.55	7.4	X
Avg Categorical Similarity	3.25	5.60	72.0%	
Avg Sales Rank	81729	24343	70.2%	
Graph subset - All Products with 1 or more co-selling products				
Avg Recommended Products	2.69	13.97	5.2	X
Avg Categorical Similarity	4.81	5.77	19.9%	
Avg Sales Rank	67662	21654	68.0%	
Books				
Avg Recommended Products	2.75	14.00	5.1	X
Avg Categorical Similarity	4.85	5.78	19.0%	
Avg Sales Rank	84674	27450	67.6%	
Music				
Avg Recommended Products	2.43	13.96	5.7	X
Avg Categorical Similarity	4.92	5.67	15.2%	
Avg Sales Rank	26739	8217	69.3%	
Video				
Avg Recommended Products	1.95	12.73	6.5	X
Avg Categorical Similarity	3.56	5.83	63.7%	
Avg Sales Rank	7196	2821	60.8%	
DVD				
Avg Recommended Products	3.43	14.99	4.4	X
Avg Categorical Similarity	5.03	6.08	20.9%	
Avg Sales Rank	8430	2646	68.6%	

Since the example we have used is on Books let me explore the books category in more detail to explain this table. When we started, on average there were 2.75 recommendations for each book, not near enough to provide adequate choices, after the transformation, we now have 14 recommendations per book, that is much more useful for an eCommerce engine. This also ensures a larger set of choices provided to consumers. As a measure of relevance we have used Categorical Similarity and on this measure there is a 19% improvement in Categorical Similarity. The SalesRank for a product is computed by Amazon as a function of historical sales. A lower SalesRank signifies a higher selling product. There is a nice improvement in the average SalesRank by 67% increasing the probability of selling other products.

Now let me challenge some of the assumptions made in this project:

- increase average # of recommended products to ~15, is this a reasonable number, do we need more, do we need less; in this case with the same algorithm this can be changed
- the "weight" of each Level in the category tree is the same, we could weigh deeper sub-categories higher and see if that yields better results, this is worth experimenting with
- lowering the average PageRank of the co-selling products could lead to a rich get richer phenomenon, is this what we want or do we want to sprinkle the recommendations with some high sellers and some low sellers, this is certainly a strategy that can be investigated
- is Categorical Similarity a good measure, the results seem promising, but needs to be tested
- to the last point and everything stated so far the ultimate measure is will this lead to more ecommerce sales and the only way we determine that is by real world tests of the model
- finally the assumption made in this discussion about maximizing sales is to maximize the number of items sold, would we do anything different to maximize sales revenues, we suspect the approach is similar, but this would be another worthy path to investigate

5.0 Conclusion

We started with a Dataset that provided a sparse set of recommendations for co-selling products. This is a real world scenario where one needs to produce more and relevant recommendations given a limited set of data. We looked at cluster models and collaborative filtering, both of which would require more information on customers that was not available. So with the limited information available, we came up with a metric around Categorical Similarity and provided additional recommendations that improved upon both this measure and the average SalesRank. Now in a real world scenario the next step is to run a test for a limited period of time, so we can progressively refine it based on real results. That said our theoretical results show that we are headed in the right direction.

6.0 References

1. J. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Conf. Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1998, pp. 43-52.
2. B.M. Sarwar et al., "Item-Based Collaborative Filtering Recommendation Algorithms," 10th Int'l World Wide Web Conference, ACM Press, 2001, pp. 285-295.
3. M. Balabanovic and Y. Shoham, "Content-Based Collaborative Recommendation," Comm. ACM, Mar. 1997, pp. 66-72.
4. L. Ungar and D. Foster, "Clustering Methods for Collaborative Filtering," Proc. Workshop on Recommendation Systems, AAAI Press, 1998.
5. Linden G., Smith S. & York J, – Amazon.com Recommendations: Item to Item Collaborative Filtering, 2003,