

Collaborative Quality Filtering: Establishing Consensus or Recovering Ground Truth?

Jonathan Traupman and Robert Wilensky*

Computer Science Division
University of California, Berkeley
387 Soda Hall
Berkeley, CA 94702
{jont, wilensky}@cs.berkeley.edu

Abstract

We present a algorithm based on factor analysis for performing collaborative quality filtering (CQF). Unlike previous approaches to CQF, which estimate the consensus opinion of a group of reviewers, our algorithm uses a generative model of the review process to estimate the latent intrinsic quality of the items under reviews. We run several tests that demonstrate that consensus and intrinsic quality are, in fact different and unrelated aspects of quality. These results suggest that asymptotic consensus, which purports to model peer review, is, in fact, not recovering the ground truth quality of reviewed items.

Key Words: Collaborative Quality Filtering, Factor Analysis, Recommender Systems

1. Introduction

Despite the vast number of reviews of products, stores, media, and articles on the Internet, little has been done to separate worthwhile expertise from biased or uninformed opinions. When assigning a total score to an item, nearly all review sites simply average the scores given by each reviewer. A few, such as the Internet Movie Database [8], use a Bayesian mean, which takes into account both the number of reviews written as well as the scores given to a particular film. None of these systems make any attempt to determine the ability of the reviewers to estimate the quality of items they review. Such a method would be necessary to emulate peer review, which weights the opinions of experts most heavily.

Collaborative Quality Filtering (CQF) [16, 19] attempts to improve these estimates for item quality by giving more accurate reviewers more weight than less accurate ones. Unlike standard collaborative filtering, CQF systems do not create personal recommendations for users. Instead, they use individual reviews to estimate the underlying quality of the item being reviewed. Clearly, collaborative quality filtering is less useful for items where people's tastes tend to differ, such as movies, books, and music. However, for items where there is agreement about what constitutes quality, it can be a more accurate method of estimating quality than the simple or Bayesian means in common use today. Indeed, one goal of CQF research is to rationalize subjective processes, such as reviewing academic papers or ranking journals and institutions.

In [16] and [19], a CQF algorithm is described that works by what we will call "asymptotic consensus": an iterative method that gives greater weights to reviewers who give scores closer to the weighted mean for each item.

Here, we present a new algorithm for CQF based on factor analysis. In comparing these two algorithms, we have discovered what we believe to be a fundamental division in approaches to CQF. While both approaches use a set of reviews to compute estimates of item quality, they differ in what they consider to be a "high quality" item.

The first approach, exemplified by the asymptotic consensus algorithm, takes the view that quality is not an a priori property of an item. Instead, the quality estimates assigned by these algorithms attempt to reflect the consensus opinion of the reviewers. Algorithms in this family try to maximize the agreement

*This research was supported by the Digital Libraries Initiative under grant NSF CA98-17353.

among reviewers about the quality of an item by giving greater weight to reviewers whose opinions best reflect the consensus.

The second approach assumes that quality is an intrinsic property of an item that can be judged by reviewers but is not defined by them. Algorithms such as factor analysis estimate this intrinsic quality by assuming a generative model of reviews and then estimating the intrinsic qualities that best explain the observed data. The notion that quality is an intrinsic property of an item raises interesting philosophical issues, which we cannot do justice to here. However, we note that this view posits that quality is not a completely arbitrary social construction (even if the ideals with respect to which quality is judged are socially construed). That is, reviewers are making a determination about properties of an artifact, but their collective consensus doesn't establish the underlying reality. We view it as a positive if the reviewing process is in fact uncovering some underlying truth, rather than fabricating an arbitrary social agreement. Ideally, the social process and the uncovering of intrinsic properties would converge.

We believe that both of these approaches will find uses in different applications, but the vastly different results we have seen suggest that users of CQF systems need to consider carefully just what type of "quality" they hope to measure. For example, while the asymptotic consensus algorithm purports to model academic review, these results suggest it is measuring consensus rather than recovering an objective notion of quality. For a system intended to retrieve the highest quality articles, a CQF algorithm that accurately estimates intrinsic quality may in fact be more useful.

2. Related Work

Riggs and Wilensky [16, 19] coined the term *Collaborative Quality Filtering* to refer to the process of estimating the quality of an item using reviewer preference data, in contrast to standard collaborative filtering, which suggests items users may like based on their preferences. They presented a CQF system that works by asymptotic consensus, which we compare to our method.

Most existing systems for ranking item quality use either the mean of the reviews for an item or a Bayesian mean [8] of reviews as a proxy for quality. In the world of publishing, citation analysis [11, 1] has been used to measure the influence and thus the quality of articles and journals. However, citations are, at best, an indirect measure of quality and provide little information for more recently published works.

Collaborative Filtering makes recommendations rather than estimates item quality, but uses techniques that can be applied to CQF. Early, "memory-based," systems work by finding users with similar tastes and then making recommendations based on these preferences. These systems filter items as diverse as email [4], NetNews [14], movies [6] and music [18]. Herlocker [5] compares the performance of a number of these techniques.

More recently, model-based approaches, including latent semantic analysis [7] and factor analysis [2], have been applied to collaborative quality filtering with good results. The March 1997 issue of CACM [15] presents good summaries of early work on collaborative filtering and the recent special issue of ACM Transactions on Information Systems [12] highlights some contemporary results in this field.

While not a CQF system, [13] applies a hybrid approach, combining both collaborative and content-based filtering, to academic publications. However, like most collaborative filtering systems, it provides recommendations based on similar taste rather than on estimates of intrinsic quality.

3. Factor Analysis

Our CQF system is based on the widely used factor analysis (FA) method, a dimensional reduction technique that takes a high dimensional data vector as input and returns a vector of factors of typically much lower dimension. These factors form a basis for a subspace that captures as much of the variance in the original data space as possible. Our implementation uses an iterative EM algorithm, an approach widely discussed in available literature [17, 10], so we skip its derivation and simply present a summary of our implementation.

3.1 Factor Analysis of Dense Datasets

The data sets used in collaborative quality filtering are typically very sparse, since most reviewers will only rate a small fraction of the available items. However, it is helpful to understand factor analysis on dense datasets before examining the modifications necessary to make it work with sparse data.

The factor analysis model consists of a set of M independent, identically distributed data points, one for each item being reviewed. Each data point is associated with two random variables: R , a length N vector of observed reviews, and Q , the latent intrinsic quality that the FA algorithm estimates from the reviews. In the general case, Q may be a vector of length P , with $P < N$, but in our application, we choose to have $P = 1$, representing the single factor of “intrinsic quality.”

The entire dataset consists of the $M \times N$ matrix R . Each row, R_m , is a vector of N reviews for item m . Each element, $R_{m,n}$, is the review given to item m by reviewer n . The output of our algorithm is a $M \times 1$ matrix, Q , where each scalar row, Q_m , is the estimated quality of item m .

We assume that each Q_m is independent and distributed normally with a mean of zero and unit variance. The distribution of R_m conditioned on Q_m is also normal with mean $\mu + \Lambda Q_m$ and diagonal covariance matrix Ψ . Λ and μ are N element vectors and Ψ is a $N \times N$ diagonal matrix. We use a maximum likelihood approach to estimate the values of these three parameters.

The μ parameter can be estimated simply as the sample mean:

$$\mu = \frac{1}{M} \sum_{m=1}^M R_m \quad (1)$$

The other parameters, Λ , and Ψ , are estimated using the EM algorithm.

For the E-step, we calculate the conditional expectations, $E(Q_m|R_m)$, and the estimated variances, $\text{Var}(Q_m|R_m)$. [10] shows that:

$$\begin{aligned} E(Q_m|R_m) &= (I + \Lambda^T \Psi^{-1} \Lambda)^{-1} \Lambda^T \Psi^{-1} (R_m - \mu) \\ \text{Var}(Q_m|R_m) &= (I + \Lambda^T \Psi^{-1} \Lambda)^{-1} \end{aligned} \quad (2) \quad (3)$$

We can then use these two expected values to obtain the necessary estimates of the sufficient statistics used in the M-step:

$$\langle Q_m \rangle = E(Q_m|R_m) \quad (4)$$

$$\langle Q_m^2 \rangle = \text{Var}(Q_m|R_m) + E(Q_m|R_m)^2 \quad (5)$$

Note that some of these equations are slightly different than classical derivations of factor analysis because our Q_m are scalars, not vectors.

The M-step uses these estimates to update the parameter estimates:

$$\Lambda^{(t+1)} = \left(\sum_{m=1}^M R_m \langle Q_m \rangle \right) \left(\sum_{m=1}^M \langle Q_m^2 \rangle \right)^{-1} \quad (6)$$

$$\Psi^{(t+1)} = \frac{1}{M} \text{diag} \left(\sum_{m=1}^M R_m R_m^T - \Lambda^{(t+1)} \sum_{m=1}^M \langle Q_m \rangle R_m^T \right) \quad (7)$$

These new estimates are then used by the next iteration’s E-step. EM is proven to converge to a local maximum and in practice converges fairly rapidly.

Once the EM algorithm converges, the estimate of $E(Q_m|R_m)$ is our estimate of the quality of the item m . The first term in the equation for $E(Q_m|R_m)$:

$$(I + \Lambda^T \Psi^{-1} \Lambda)^{-1} \Lambda^T \Psi^{-1} \quad (8)$$

can be interpreted as a length N vector of weights, each of which represent the amount of confidence we have in a reviewer.

3.2 Factor Analysis of Sparse Datasets

Because the data sets we will be using for CQF are typically very sparse, we need to make a few modifications to the standard FA algorithm. These modifications are derived from the sparse FA algorithm described by [2], which in turn is based on [3]. The main difference between our derivation and [2] is our use of a diagonal covariance matrix Ψ , which allows individual variance values for each reviewer, instead of a single scalar variance that applies to all reviewers.

While our approach is very similar algorithmically to [2], it is important to note that we are applying it to a different problem. The collaborative filtering system in [2] finds factors across users to make personalized recommendations. We use factor analysis to find factors across items for estimating item quality.

We introduce a set of $N \times N$ diagonal *trimming matrices*, T_m , one for each item m in the dataset. Each diagonal element, $T_{m,n}$, is one if and only if there is a review of item m by reviewer n in the dataset and zero otherwise. All non-diagonal elements of T_m are zero.

The sparse estimate for μ now becomes:

$$\mu = \left(\sum_{m=1}^M T_m \right)^{-1} \left(\sum_{m=1}^M R_m \right) \quad (9)$$

Similarly, we have updated equations for the E-step:

$$\bar{\Lambda}_m = T_m \Lambda \quad (10)$$

$$E(Q_m | R_m) = (I + \bar{\Lambda}_m^T \Psi^{-1} \bar{\Lambda}_m)^{-1} \bar{\Lambda}_m^T \Psi^{-1} (R_m - \mu) \quad (11)$$

$$\text{Var}(Q_m | R_m) = (I + \bar{\Lambda}_m^T \Psi^{-1} \bar{\Lambda}_m)^{-1} \quad (12)$$

Note that because the rows of the trimming matrix are unique, the terms used to calculate $\text{Var}(Q_m | R_m)$ and $E(Q_m | R_m)$ are different for each row of the data matrix. The unfortunate result is that the E-step becomes much more computationally expensive.

These estimated can then be used to compute the sufficient statistics $\langle Q_m \rangle$ and $\langle Q_m^2 \rangle$ used in the M-step in the same fashion as the dense case.

The update equations for the M-step are also similar to their dense counterparts:

$$\Lambda^{(t+1)} = \left(\sum_{m=1}^M T_m \langle Q_m^2 \rangle \right)^{-1} \left(\sum_{m=1}^M R_m \langle Q_m \rangle \right) \quad (13)$$

$$\Psi^{(t+1)} = \left(\sum_{m=1}^M T_m \right)^{-1} \text{diag} \left(\sum_{m=1}^M R_m R_m^T - T_m \Lambda^{(t+1)} \langle Q_m \rangle R_m^T \right) \quad (14)$$

Canny [2] presents a more complicated form of 13 that can handle the general case of a vector Q_m . He also includes normalization terms necessary for his collaborative filtering application, which significantly degrade the performance in our application.

The estimated quality of the items being reviewed is, as in the dense case, simply the expected values $E(Q_m | R_m)$. Calculating the reviewer weight vector requires a little more effort than the dense case:

$$\left(\sum_{m=1}^M T_m \right)^{-1} \left(\sum_{m=1}^M (I + \bar{\Lambda}_m^T \Psi^{-1} \bar{\Lambda}_m)^{-1} \bar{\Lambda}_m^T \Psi^{-1} \right) \quad (15)$$

where $\bar{\Lambda}_m$ is defined as in equation 11. Without this summation and normalization, reviewers with few reviews receive unrealistically high weights.

3.3 Smoothing

To further improve performance with sparse data, we implemented a simple smoothing method loosely based on the technique of deleted interpolation [9], commonly used to smooth Hidden Markov Models.

We smooth Λ and Ψ using a linear combination of the estimates produced by the EM algorithm and prior estimates of these parameters:

$$\Lambda_n = (1 - w_n)\Lambda_{EM,n} + w_n\Lambda_{prior,n} \quad (16)$$

The equation for Ψ is identical. Each weight, w_n , is the reciprocal of the number of reviews written by reviewer n , which gives more weight to the EM estimates for reviewers with many reviews and more weight to the prior for reviewers with few reviews.

We determine the prior estimates of Λ and Ψ by running the EM factor analysis algorithm on a subset of the dataset restricted to reviewers with 100 or more reviews, then take the mean of the resulting Λ and Ψ vectors.

3.4 Implementation and Performance

We implement factor analysis in C++ as a Matlab plug-in. An 867MHz Pentium III with 512MB of RAM requires approximately 111 minutes to process the full dataset. Smoothing reduces the number of iterations needed for convergence and thus decreases the time by about two thirds. For extremely large datasets, factor analysis can easily be parallelized.

For processing large datasets that change and grow over time, a previous run's Λ and Ψ values can be used to bootstrap subsequent runs, drastically reducing the number of EM iterations required. A large review website, for example, would only need to run the full EM algorithm once. As new items, reviewers, and reviews are added to the site, the existing parameter values can be used to bootstrap EM on the new dataset. As long as new information is only a small fraction of the total data set, only one or two iterations should be necessary to re-estimate the parameter values.

4. Methodology

Testing CQF algorithms present several challenges in addition to the implementation problems related to dataset size and sparseness. Existing datasets, such as the data from epinions.com that we used, do not come with ground truth, making it impossible to measure how well an algorithm recovers intrinsic quality. Likewise, it is easy to create synthetically generated datasets with known ground truth, but not obvious how to model agreement among reviewers and thus consensus.

4.1 Testing Consensus

To test an algorithm's ability to measure consensus, we use a dataset gathered by crawling the epinions.com site in 2001. Since the factor analysis algorithm will fail with a singular Ψ matrix if there are any reviewers with fewer than two reviews, we trim out reviewers with only one review, leaving us with 104,889 items and 53,796 reviewers.

In addition to the individual reviews of items, the epinions.com dataset also includes readers' evaluations of the reviewers, which we use as a measure of consensus. Users of the site are allowed to indicate reviewers that they trust. The number of users that trust a reviewer thus becomes a metric for the community's opinion of a reviewer's reviews. Since the reviews of a prolific reviewer who has rated many items are more likely to be seen and therefore trusted, we divide the raw number of users trusting each reviewer by the number of reviews the reviewer has written to determine an average trust score.

It must be noted that average trust is not necessarily a good indicator of a reviewer's ability to accurately judge the quality of an item. A reader's decision to trust a reviewer is also influenced by subjective factors, such as the length, thoroughness, and quality of the written reviews. However, we believe one important

influence on trust scores is how closely a reviewer’s opinions match those of the site’s readers. Reviewers who consistently write reviews that agree with the consensus opinion will likely be trusted by large numbers of readers.

4.2 Testing Intrinsic Quality

The epinions.com dataset does not contain any information about the ground truth intrinsic quality of the items it contains, nor do any sets of user reviews that we know of. Therefore, we use synthetically generated datasets to test each algorithm’s ability to recover the ground truth quality.

One of the biggest risks when using synthetic datasets is that the choice of models for generating the data may bias the set to favor one algorithm over another. For example, if we created a synthetic dataset with a zero mean, unit variance Gaussian distributed ground truth and reviews generated by a simple linear transformation of this ground truth, we would imagine that factor analysis would perform superbly, since that is exactly the model that factor analysis assumes.

To more completely and fairly evaluate these algorithms, we create two synthetic datasets, both considerably different than the simplistic approach described above. These two datasets use different distributions of ground truth quality, which allows us to measure how well our algorithm performs when presented with data whose distribution differs from what its model assumes.

When creating the synthetic datasets, we were primarily concerned with making it as similar as possible to the real epinions.com dataset. The sparseness of reviews in the real and synthetic datasets are identical: an item, m , has a review by reviewer n in the synthetic datasets if and only if n reviewed m in the real dataset.

The distribution of reviews in the epinions.com set is skewed (the average review is 1 on a scale of -2 to 2), so we added similar skew to our synthetic sets. However, the way we accounted for this skew is different in our two synthetic datasets. For the first set, we assumed that the observed skew is a result of the review process and not indicative of the intrinsic quality. In this set, the ground truth is normally distributed with a mean of zero and variance of one, and thus has no skew. Because the asymptotic consensus algorithm and the mean both assume a bounded quality scale, we truncate the ground truth values to lie between -2 and 2.¹

An alternative explanation for the observed skew in the epinions.com data is that it is a reflection of skew in the intrinsic quality of the items being reviewed. Our second synthetic dataset explores this possibility. The ground truth of this dataset is generated so that the frequency of each score is the same as the frequency of that score in the observed data.

We created the reviews by adding randomly generated noise term to the ground truth of the item being reviewed. This noise was distributed normally from mean and variance parameters associated with each reviewer. For the first dataset, the mean parameters are normally distributed with mean and variance one, which gives us the overall bias toward higher valued reviews observed in the epinions.com data. In the second dataset this bias is already present in the ground truth, so the distribution of the mean parameter has mean zero and variance one. For both datasets, the noise variance parameters were generated from the square of a zero mean, unit variance normal distribution.

¹The question of whether to model intrinsic quality by a bounded or an unbounded distribution is another philosophical difference between our factor analysis approach and earlier CQF systems. We believe that allowing both intrinsic quality and reviewer scores to be unbounded more closely matches our intuitive understanding of quality. After all, at any time it is possible to encounter an item of higher quality than any previous seen item, though the probability of such an encounter becomes smaller as more items are seen. However, we chose to bound both the ground truth intrinsic quality and the reviewer scores for this experiment, in order to level the field for all three methods and more closely model the epinions.com data. Experiments we conducted using unbounded intrinsic quality but bounded reviews showed no significant differences from results we present here.

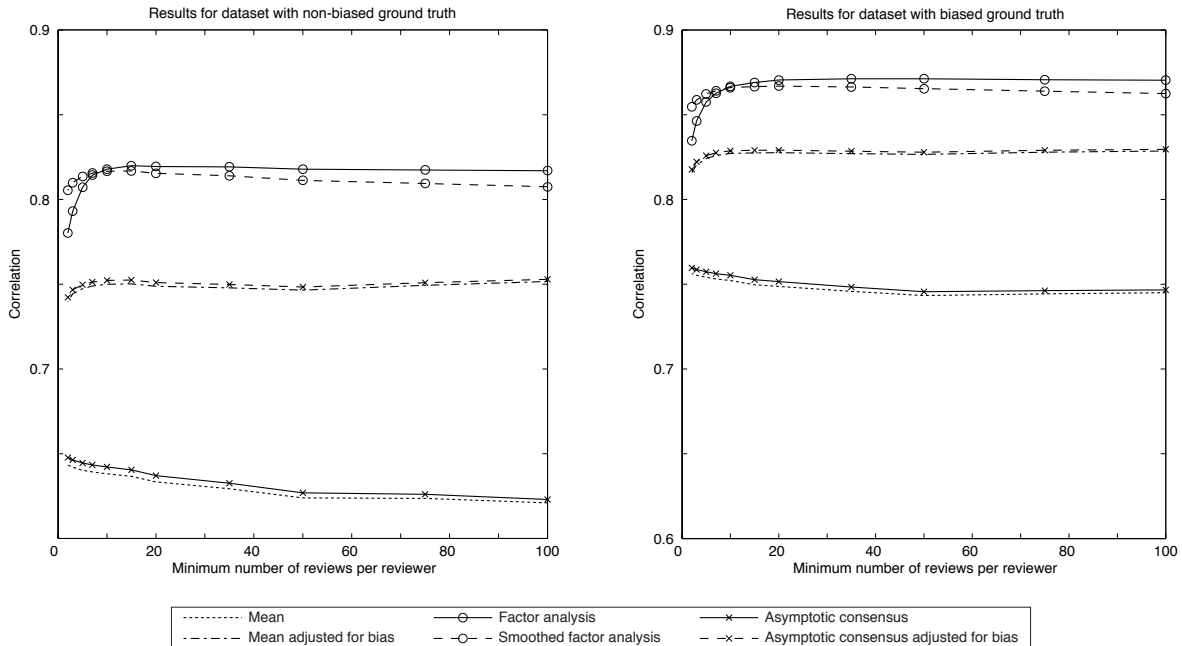


Figure 1. Comparison of CQF algorithms on a dataset with unbiased ground truth, left, and one with biased ground truth, right.

5. Results

We performed two sets of tests on several variants of both the factor analysis and asymptotic consensus CQF algorithms. The first set of experiments tests the abilities of the algorithms to recover the ground truth intrinsic quality of the items reviewed. For this test, we use our two synthetically generated datasets, which include ground truth information. The second experiment measures the algorithms’ ability to estimate the consensus opinion of the reviewers. This test uses the epinions.com review data and uses the average trust statistic as a measure of how closely a reviewer’s opinions are aligned with that of the consensus.

5.1 Recovering Ground Truth

To test a CQF algorithm’s ability to recover ground truth intrinsic quality, we compared its estimates of item quality to the known ground truth using our two synthetically generated datasets. Dataset 1 has no bias in its ground truth, while the ground truth for dataset 2 has a bias equal to the bias we observed in the epinions.com data. We compared our factor analysis CQF algorithm, the asymptotic consensus algorithm, and also simple mean of an item’s reviews, which serves as a baseline.

We compare the algorithms’ quality estimates to the ground truth by measuring their correlation. The use of correlation, as opposed to a simpler metric like mean absolute error, allows us to ignore differences in scale and normalization among the systems tested.

We tested factor analysis both with and without the smoothing technique described in Sect. 3.3. We found that the α parameter, a smoothing technique for asymptotic consensus that gives greater weight to reviewers with more reviews, had no effect on performance in this test, so these results are omitted for clarity. We did not test the β and γ factors because they had little effect according to [16].

We discovered that both the mean and the asymptotic consensus algorithm delivered better results when we adjusted for reviewer bias by normalizing each reviewer’s average score to zero, so we also tested versions of these algorithms with the observed data adjusted for bias. We did not adjust for the bias with factor

analysis because this calculation is already part of the factor analysis algorithm.

To show how sparseness affected these algorithms, we varied the minimum number of reviews necessary for a reviewer to be considered by the algorithms. The most sparse run used all reviewers with at least two reviews, while other runs only used reviewers with a minimum of 3, 5, 7, 10, 15, 20, 35, 50, 75, and 100 reviews.

The results of our tests can be seen in Fig. 1, which shows that factor analysis is significantly better than both the mean and the asymptotic consensus algorithm at recovering ground truth quality. Smoothing improves performance with sparse data, but slightly degrades performance with denser datasets. The asymptotic consensus algorithm provides little gain versus the mean on this test, but adjusting for reviewer bias provides a significant improvement to both the mean and asymptotic consensus algorithms.

The results are roughly the same on both of our datasets. The mean and the asymptotic consensus algorithm showed better performance on the second dataset, where the skew in the observed data is due to similar skew in the ground truth. This result is not surprising, since the distribution of observed data and ground truth data are much more similar in this dataset than in the first one.

However, factor analysis also shows a slight increase in overall performance compared to the first dataset. We believe that this result demonstrates that factor analysis is a robust technique that excels at recovering intrinsic quality even when the distribution of that ground truth data is quite different than what the factor analysis model assumes.

As one might expect, all of the algorithms predict the quality of items with many reviews far more accurately than items with only a few reviews. However, because approximately 90% of items in these datasets have 10 or fewer reviews, an algorithm’s ability to accurately estimate the quality of items with very few reviews has significant impact on its total performance.

For example, when run on the first dataset with a minimum of 10 reviews per reviewer, the average correlation between ground truth and factor analysis’s quality estimates for items with only one review is 0.73 compared to 0.56 for the mean and 0.68 for the mean adjusted for bias. Other datasets and levels of sparseness show similar results: factor analysis’s greatest benefits are for items with few reviews. Asymptotic consensus, however, shows no benefit versus the mean for items with a single review because by definition, it and the mean return the same estimate for such items.

5.2 Estimating consensus

To measure a CQF system’s ability to estimate the consensus of a group of reviewers, we measured the correlation between the reviewer weights returned by an algorithm and the average trust score assigned to the reviewer in the epinions.com dataset.

For this test, we compared factor analysis, both with and without smoothing, to several variants of the asymptotic consensus algorithm. For asymptotic consensus, we used both the raw reviewer data and the data where we adjusted each reviewer’s bias to zero. We also compared using asymptotic consensus’s α parameter to not using it. Because the mean does not calculate weights for individual reviewers, we omitted it from this experiment. The results of comparing these six algorithm variants is shown in Fig. 2.

None of the algorithms did particularly well on this test, with a maximum correlation of about 0.18. We believe this has as much to do with average trust being a rather poor indicator of consensus as it does with any weakness in the particular algorithms.

As expected, asymptotic consensus is clearly best at estimating reviewer consensus. Using the α parameter—a smoothing parameter that discounts reviewers with few reviews—provides a significant boost with highly sparse datasets. Smoothing also provides a noticeable benefit for factor analysis, though even the smoothed version lags behind asymptotic consensus.

Interestingly, asymptotic consensus performs slightly worse on this test when we center the reviewer data to zero out any reviewer bias. While unbiased reviews allow better predictions of intrinsic quality, they actually hurt the algorithm’s ability to predict consensus. Upon reflection, this isn’t surprising: if the consensus opinion is also biased, we should expect the best prediction will be made from biased data.

In general, techniques that were most successful at estimating intrinsic quality—namely factor analysis and using unbiased data—had a detrimental impact when estimating consensus. Others which had no effect

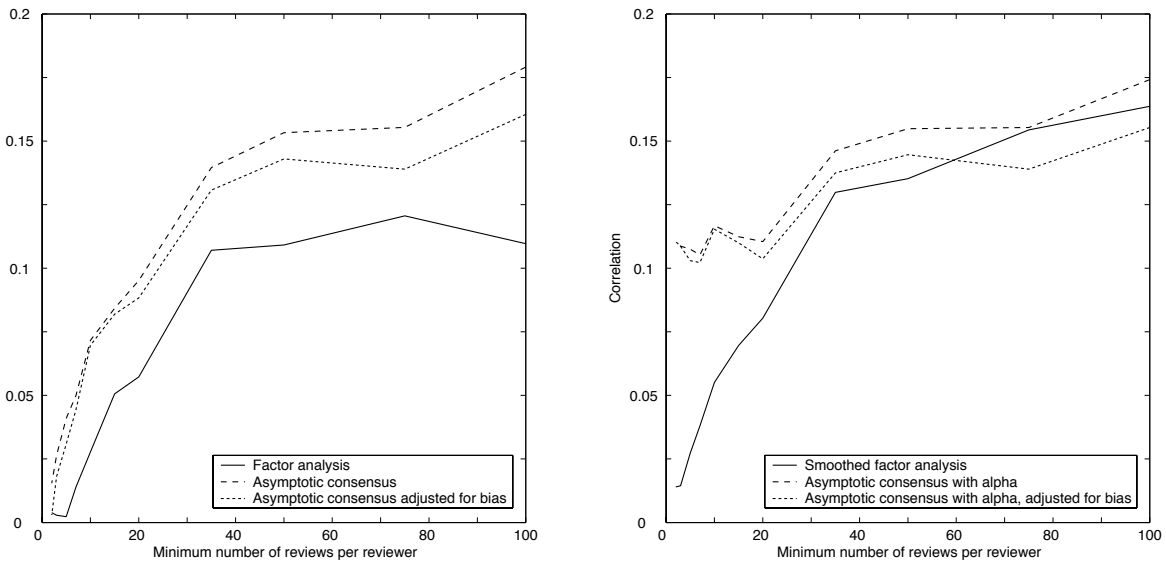


Figure 2. Comparison of algorithms' abilities estimating reviewer consensus.

on estimating quality, like the α parameter, had large effects on this test. We feel the simplest explanation for this behavior is that intrinsic quality and consensus are actually very different concepts of quality.

6. Conclusion

Our experiments suggest that while both asymptotic consensus and factor analysis are purported to be filtering on quality, they are in fact computing quite different things. The asymptotic consensus algorithm converges to an estimate of the consensus of a group of reviewers, while factor analysis tries to recover objective quality. While the lack of objective ground truth for either intrinsic quality or consensus makes evaluation difficult, the preponderance of evidence suggests that these two aspects of quality are actually distinct.

Neither approach is necessarily “correct”; each may be suitable for different types of applications. However, as the stated goal of asymptotic consensus is to serve as a model of scholarly review, one might hope that reviewer consensus and intrinsic quality would converge. Unfortunately, our results suggest that they are likely to be different. Should these results withstand further scrutiny, in particular, a test against real peer review data, they would raise interesting questions about what such processes are in fact measuring. We believe that for such applications, factor analysis, or another approach that recovers intrinsic quality, may be the better technique.

References

- [1] L. Brown and J. Gardner. Using citation analysis to assess the impact of journals and articles on contemporary accounting research. *Journal of Accounting Research*, 23(1):84–109, 1995.
- [2] John Canny. Collaborative filtering with privacy via factor analysis. In *Proc. of the 25th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 238–245, August 2002.

- [3] Zoubin Ghahramani and Michael I. Jordan. Learning from incomplete data. Technical Report AIM-1509, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1994.
- [4] David Goldberg et al. Using collaborative filtering to weave and information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.
- [5] Jonathan L. Herlocker et al. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proc. of the 22nd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 230–237. ACM, 1999.
- [6] Will Hill et al. Recommending and evaluating choices in a virtual community of use. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pages 194–201, 1995.
- [7] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, January 2004.
- [8] Internet Movie Database. Top 250 films. http://www.imdb.com/top_250_films.
- [9] F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice*, pages 381–397. North-Holland Publishing Company, 1980.
- [10] Michael I. Jordan. An introduction to probabilistic graphic models. Unpublished textbook manuscript.
- [11] Pairin Katerattanakui et al. Objective quality ranking of computing journals. *Communications of the ACM*, 46(10):111–114, October 2003.
- [12] Joseph A. Konstan. Introduction to recommender systems: Algorithms and evaluation. *ACM Transactions on Information Systems*, 22(1):1–4, January 2004.
- [13] Stuart E. Middleton et al. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22(1):54–88, January 2004.
- [14] P. Resnick et al. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proc. of ACM 1994 Conf. on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.
- [15] Paul Resnick and Hal R. Varian. Recommender systems. *Communications of the ACM*, March 1997.
- [16] Tracy Riggs. Collaborative quality filtering in open review systems. Master’s thesis, University of California, Berkeley, Computer Science Division, December 2001.
- [17] Donald B. Rubin and Dorothy T. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, March 1982.
- [18] Upendra Shardanand and Patti Maes. Social information filtering: Algorithms for automating “word of mouth”. In *Proc. of ACM CHI'95 Conf. on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.
- [19] Robert Wilensky and Tracy Riggs. An algorithm for automatically rating reviewers. In *Proc. of the First Joint Conf. on Digital Libraries*, Roanoke, Virginia, June 2001.