NTNU
Norwegian University of
Science and Technology

Adaptive Aggregation of Recommender Systems

Olav Bjørkøy

Department of Computer and Information Science

October 3rd, 2011

Terminology

u	user
i	item (article, website, movie, product...)
r	rating, relevance, utility (domain specific)
m	a method for predicting ratings.
$p(m, u, i)$	predicted rating of a method for (u, i).

2006: The Netflix Challenge

- USD 1MM prize for a 10% accuracy improvement.
- Breakthrough: Combining methods from many teams.
- Team BellKor finally achieved a 10.06% improvement by combining **107** different recommender algorithms.

Today: Web Search

Google

"Today we use more than 200 signals, including PageRank, to order websites, and we update these algorithms on a weekly basis." (google.com/corporate/tech.html)

Bing

"We use over 1,000 different signals and features in our ranking algorithm." (bing.com/community/site_blogs/b/search/archive/2011/02/01/thoughts-on-search-quality.aspx)

Why multiple algorithms?

- Use more data.
- Capture more predictive aspects.
- Disjoint predictors.

Bell, R., Koren, Y., and Volinsky, C. (2007) (Netflix)

"Quite frequently we have found that the more accurate predictors are less useful within the full blend."

The Problem: Latent Subjectivity

$$\hat{r}_{u,i} = \sum_{m \in M} w_m \times p_r(m, u, i) \tag{1}$$

- Generalized optimal weights.
- Treats all users and items the same.
- Varying accuracy across users and items.
- Methods are chosen by the system, not the users or items.

The Problem: Latent Subjectivity

Systems that insist on being adaptive in a certain way are not really adaptive at all.

Adaptive Recommenders

$$\hat{r}_{u,i} = \sum_{m \in M} p_w(m, u, i) \times p_r(m, u, i) \tag{2}$$

— p_r: predicted rating from method m for (u, i).
— p_w: predicted optimal weight for method m for (u, i).
— We can use standard recommenders for both p_r and p_w.

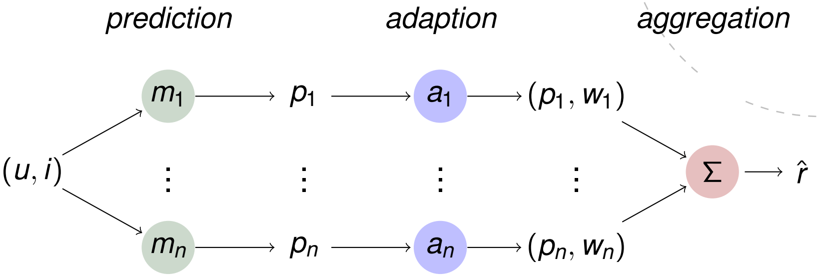

Figure: Layers of recommenders.

Training phase

- Split into two sets using bootstrap aggregation: (d_m, d_e).
- Use d_m to train the basic recommenders.
- Create an error matrix using the basic recommenders and d_e.
- The error matrix values are predicted errors for (u, i, m).
- Train adaptive recommenders on the error matrix.

$$\forall (u, i, r) \in (d_e - d_m) : E(m)_{u,i} = |r - p(m, u, i)| \tag{3}$$

$$R_{u,i} = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,i} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ r_{u,1} & r_{u,2} & \cdots & r_{u,i} \end{pmatrix}$$

$$E_{u,i} = \begin{pmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,i} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ e_{u,1} & e_{u,2} & \cdots & e_{u,i} \end{pmatrix}$$

Train p_r with R and p_w with E.

Prediction phase

- Calculate each prediction $\hat{r}_{(u,i,m)}$.
- Calculate each predicted error $\hat{e}_{(u,i,m)}$.
- The adaptive weights are the inverses of the normalized error.
- Sum the weighted predictions to get the final $\hat{r}$.

$$\hat{r}_{u,i} = \sum_{(m_e,m_r)\in M} \left(1 - \frac{p(m_e,u,i)}{error(u,i)}\right) \times p(m_r,u,i) \tag{4}$$

$$error(u,i) = \sum_{m_e \in M} p(m_e,u,i) \tag{5}$$

Results

- Calculate RMSE values for basic recommenders, simple aggregations and adaptive aggregation.
- Used the Movielens movie rating dataset (see paper for more details).

$$\mathrm{RMSE}(\hat{R}, R) = \sqrt{\frac{\sum_{i=1}^{n}(\hat{R}_i - R_i)^2}{n}} \tag{6}$$

(a) RMSE values for the five disjoint subsets:

	method	d_1	d_2	d_3	d_4	d_5
S	svd1	1.2389	1.1260	1.1327	1.1045	1.1184
S	svd2	1.2630	1.1416	1.1260	1.1458	1.1260
S	svd3	1.0061	0.9825	0.9830	0.9815	0.9797
S	svd4	1.0040	0.9830	0.9849	0.9850	0.9798
S	slope_one	1.1919	1.0540	1.0476	1.0454	1.0393
S	item_avg	1.0713	0.9692	0.9662	0.9683	0.9725
S	baseline	1.0698	0.9557	0.9527	0.9415	0.9492
S	cosine	1.1101	0.9463	0.9412	0.9413	0.9382
S	knn	1.4850	1.1435	1.1872	1.2156	1.2022
A	median	0.9869	0.8886	0.8857	0.8857	0.8855
A	average	0.9900	0.8536	0.8525	0.8525	0.8519
A	adaptive	**0.9324**	**0.8015**	**0.7993**	**0.8238**	**0.8192**

(b) Statistics for the methods:

	method	min	max	mean	σ	Δ
S	knn	1.1435	1.4850	1.2467	0.3487	-
S	svd2	1.1260	1.2630	1.1605	0.2277	6.9%
S	svd1	1.1045	1.2389	1.1441	0.2197	1.4%
S	slope_one	1.0393	1.1919	1.0756	0.2415	5.9%
S	item_avg	0.9662	1.0713	0.9895	0.2023	8.0%
S	svd4	0.9798	1.0040	0.9873	**0.0924**	2.2%
S	svd3	0.9797	1.0061	0.9865	0.0991	0.1%
S	cosine	0.9382	1.1101	0.9754	0.2595	1.1%
S	baseline	0.9415	1.0698	0.9738	0.2196	1.6%
A	median	0.8855	0.9865	0.9065	0.2005	6.9%
A	average	0.8519	0.9900	0.8801	0.2344	2.9%
A	adaptive	**0.7993**	**0.9324**	**0.8352**	0.2225	5.1%

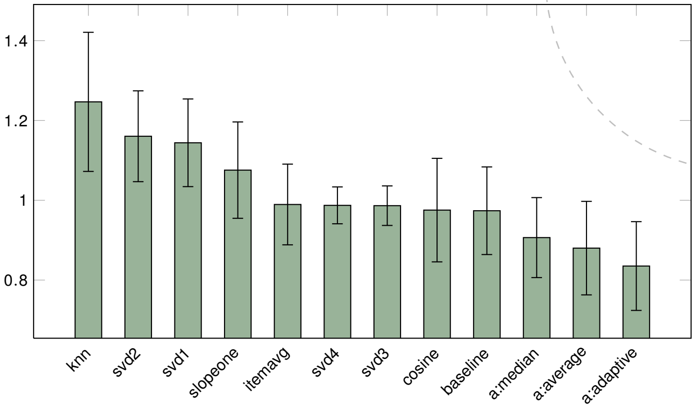

Figure: Average RMSE plot.

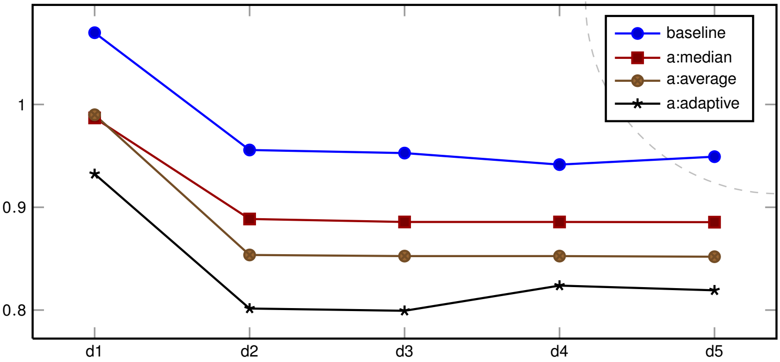

Figure: RMSE Variations: This plot shows that, while the standard deviation of each method may be high, this has more to do with the selected dataset than with their performance in comparison with each other.

Limitations

- Lots of added complexity for fairly unknown improvement.
- Only tested on a few datasets, no real world situations.
- Only compared to simple aggregation methods.
- Neither the aggregators nor the basic recommenders were heavily optimized to the domain of the dataset.

Adaptive Recommenders

- Combine disjoint algorithms
- Weight recommenders by predicted accuracy.
- Accuracy predictions are contextually dependent on (u, i, m).
- *Any* applicable recommender becomes a worthy addition.
- See paper for references and more results.