# Adaptive Aggregation of Recommender Systems

By Olav Bjørkøy

Supervised by Prof. Asbjørn Thomassen

Norwegian University of Science and Technology

Department of Computer and Information Science

India-Norway Workshop on

Web Concepts and Technologies

October 3rd, 2011

# Recommender Systems Terminology

| **term** | **description** |
|---|---|
| $u$ | a user |
| $i$ | an item (article, website, movie, email...) |
| $r$ | the rating (relevance, utility, other domain term) |
| $m$ | a method/algorithm for predicting ratings. |
| $p(m, u, i)$ | rating prediction from method $m$ 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

*"Today we use more than 200 signals, including PageRank, to order websites, and we update these algorithms on a weekly basis."*
*— Google*

*(google.com/corporate/tech.html)*

*"We use over 1,000 different signals and features in our ranking algorithm."*
*— Microsoft Bing*

*(bing.com/community/site_blogs/b/search/archive/2011/02/01/thoughts-on-search-quality.aspx)*

# Why multiple algorithms?

- The unreasonable effectiveness of data.
- Capture more predictive aspects of existing data.
- Specialized predictors for subsets of data.

*"Quite frequently we have found that the more accurate predictors are less useful within the full blend."*
*— Bell, R., Koren, Y., and Volinsky, C. (2007) (Netflix)*

# 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

The *amount* and *type* of personalization should be based on each user's preferences.

# Adaptive Recommenders

- Combining predictors by estimated contextual accuracy.

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

- $p_r$: predicted rating from method $m$ for $(u, i)$.
- $p_w$: predicted optimal weight for method $m$ for $(u, i)$.
- We can use recommender systems for **both** $p_r$ and $p_w$.

# Adaptive Recommenders

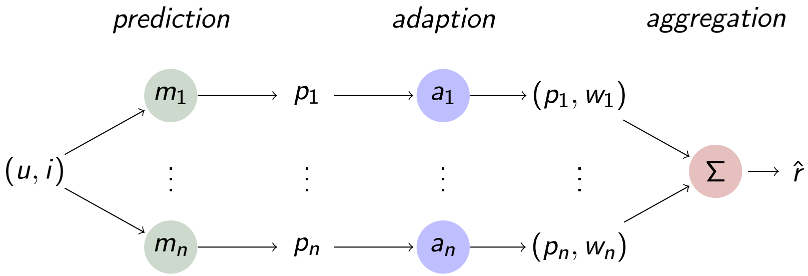


Figure: Layers of recommenders.

# Training phase

1. Split data into two sets: $(d_m, d_e)$.
2. Use $d_m$ to train the rating predictors.
3. Create *error matrices* for the rating predictors and $d_e$.
4. The error matrices now hold known errors for some $(m, u, i)$.
5. Train error predictors from the error matrices.

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

# Training phase

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

$$Error(m)_{u,i} = \begin{pmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,i} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ e_{u,1} & e_{u,2} & \cdots & e_{u,i} \end{pmatrix}$$

- $\mathrm{train(m_1, R)} \rightarrow$ rating predictor $m_1$
- $\mathrm{train(m_2, E(m_1))} \rightarrow$ error predictor for $m_1$

# Prediction phase

1. Predict ratings $\hat{r}_{(u,i,m)}$.
2. Predict errors $\hat{e}_{(u,i,m)}$.
3. Create adaptive weights by inversing the normalized errors.
4. Sum the weighted predictions to get the final $\hat{r}$.

# Prediction phase

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

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

# Experiment

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

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

# Results

(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 | $\sigma$ | $\Delta$ |
|---|---|---|---|---|---|---|
| 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% |

# Results

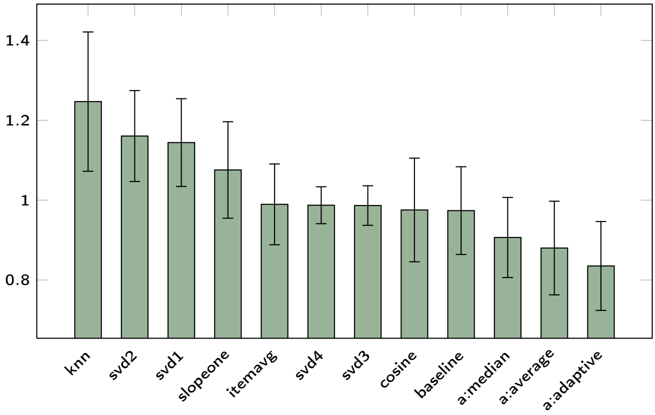


Figure: Average RMSE plot.

# Results

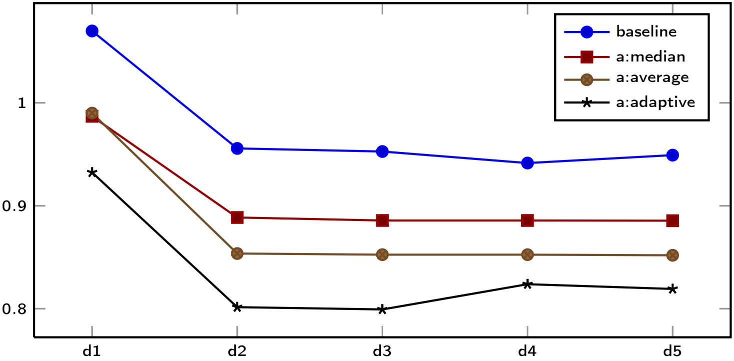


Figure: RMSE Standard deviation caused by dataset $d_1$.

# 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

- Combines disjoint rating prediction algorithms.
- Weigh algorithms by their predicted accuracy.
- Accuracy predictions are contextually dependent on $(u, i, m)$.
- *Any* applicable recommender becomes a worthy addition.

See paper for more references and results.