Evaluation of Recommender Systems

What is a good recommendation?

What are the measures in practice?

- Total sales numbers
- Promotion of certain items
- …
- Click-through-rates
- Interactivity on platform
- …
- Customer return rates
- Customer satisfaction and loyalty

Analytics Vidhya

Evaluating Recommender Systems

- A myriad of techniques has been proposed, **but**
  - Which one is the best in a given application domain?
  - What are the success factors of different techniques?
  - Comparative analysis based on an optimality criterion?
- Research questions are:
  - Is a RS efficient with respect to a specific criteria like accuracy, user satisfaction, response time, serendipity, online conversion, ramp-up efforts, ….
  - Do customers like/buy recommended items?
  - Do customers buy items they otherwise would have not?
  - Are they satisfied with a recommendation after purchase?

Why is evaluation difficult?

- First, different algorithms may be better or worse on different data sets
  - Movie Recommender (n_users >> n_items) vs Resarch Paper Recommender (n_items >> n_users)

- Goals for evaluation may differ
  - Most of the early work in RecSys looked at accuracy
  - User Satisfaction is important – Difficult to measure
  - Coverage
  - Novelty

User Tasks for Recommender Systems

- Find Good Items
- Recommend Sequence – suggest a series of items
- Just Browsing – Feel good factor from browsing
- Finding Credible Recommender Systems – play around to find best recommender systems

Selecting the metric – Some questions to ask

- Will a given metric measure the effectiveness of a system with respect to the user tasks for which it was designed?
- Are results with the chosen metric comparable to other published research work in the field?
- Are the assumptions that a metric is based on true?
- Will a metric be sensitive enough to detect real differences that exist?

Predictive Accuracy Metrics

- Datasets with items rated by users
  - MovieLens datasets 100K-10M ratings
  - Netflix 100M ratings
- Historic user ratings constitute ground truth
- Metrics measure error rate
  - Mean Absolute Error (*MAE*) computes the deviation between predicted ratings and actual ratings
  - Root Mean Square Error (*RMSE*) is similar to *MAE*, but places more emphasis on larger deviation

$$MAE = \frac{1}{n}\sum_{i=1}^{n}| p_i - r_i |$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - r_i)^2}$$

Analytics Vidhya

Predictive Accuracy Metrics – Pros & Cons

- **Advantages**
  - The mechanics of the computation are simple and easy to understand
  - Mean absolute error has well studied statistical properties that provide for testing the significance of a difference between the mean absolute errors of two systems.
- **Disadvantages**
  - Mean absolute error may be less appropriate for tasks such as Find Good Items where a ranked result is returned to the user, who then only views items at the top of the ranking.
  - Mean absolute error may be less appropriate when the granularity of true preference (a domain feature) is small

Classification Accuracy Metrics: Precision and Recall

Retrieve (recommend) all items which are predicted to be “good”.

- **Precision:** a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved
  - E.g. the proportion of recommended movies that are actually good

$$Precision = \frac{tp}{tp + fp} = \frac{|good\ movies\ recommended|}{|\text{all recommendations}|}$$

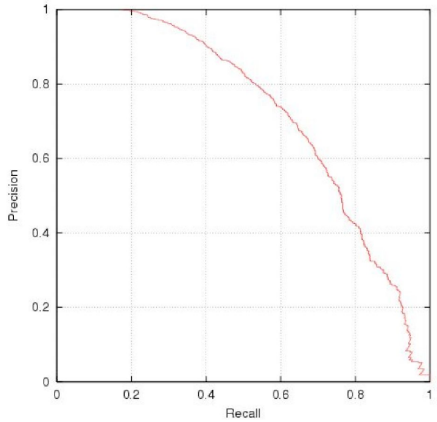
- **Recall:** a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items
  - E.g. the proportion of all good movies recommended

$$Recall = \frac{tp}{tp + fn} = \frac{|good\ movies\ recommended|}{|all\ good\ movies|}$$

Analytics Vidhya

Precision & Recall

- E.g. typically when a recommender system is tuned to increase precision, recall decreases as a result (or vice versa)



Analytics Vidhya

F1 Metric

- The **F_1 Metric** attempts to combine Precision and Recall into a single value for comparison purposes.
  - May be used to gain a more balanced view of performance

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- The F_1 Metric gives equal weight to precision and recall
  - Other F_β metrics weight recall with a factor of β.

$$F_\beta = \frac{(1 + \beta^2) \cdot (\text{precision} \cdot \text{recall})}{(\beta^2 \cdot \text{precision} + \text{recall})}$$

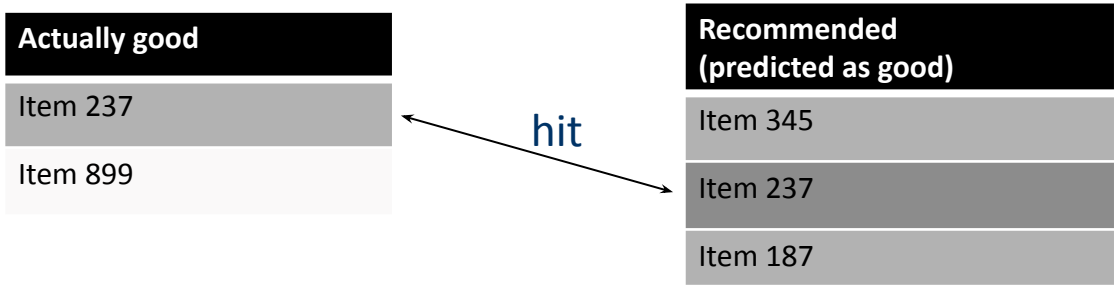
Analytics Vidhya

Classification Accuracy Metrics – Pros & Cons

- Appropriate for tasks such as Find Good Items when users have true binary preferences
- When the quality of the list is evaluated, recommendations may be encountered that have not been rated. How those items are treated in the evaluation can lead to certain biases.
- One approach to evaluation using sparse data sets is to ignore recommendations for items for which there are no ratings
- Another approach to evaluation of sparse data sets is to assume default ratings, often slightly negative, for recommended items that have not been rated. The downside of this approach is that the default rating may be very different from the true rating (unobserved) for an item
- The problem is that the quality of the items that the user would actually see may never be measured

Rank Aware Metrics: Rank Position Matters

For a user:



- **Rank metrics** extend recall and precision to take the positions of correct items in a ranked list into account
  - Relevant items are more useful when they appear earlier in the recommendation list
  - Particularly important in recommender systems as lower ranked items may be overlooked by users

Metrics: Precision@K

Evaluates the list of recommendations

- Precision at k is the proportion of recommended items in top-k set that are relevant

$$P@k(u) = \frac{\#\{\textit{relevant content in the top k postitions}\}}{k}$$

- Suppose that my precision at 10 in a top-10 recommendation problem is 80%, this means that 80% of recommendations I make are relevant to the user

Metrics: Recall@K

Evaluates the list of recommendations

- Recall at k is the proportion of relevant items found in the top-k recommendations

$$Recall@k = \frac{\#\ of\ recommended\ items\ @k\ that\ are\ relevant}{total\ \#\ of\ relevant\ items}$$

- Suppose that we computed recall at 10 and found it is 40% in our top-10 recommendation system. This means that 40% of the total number of the relevant items appear in the top-k result

Example for Precison@K & Recall@K

item	user1 (Actual/Predicted)
item1	4/2.3
item2	2/3.6
item3	3/3.4
item4	?/4.3
item5	5/4.5
item6	?/2.3
item7	2/4.9
item8	?/4.3
item9	?/3.3
item10	4/4.3

Let's say relevant items are ones with rating greater than 3.5

- Relevant items: item5, item10 and item1
  total # of relevant items = 3
- Recommended items @ 3: item7, item5 and item10
  # of recommended items at 3 = 3

- Precision@3 $= \frac{2}{3}$
- Recall@3 $= \frac{2}{3}$

Analytics Vidhya

Mean Reciprocal Rank

Evaluates the list of recommendations

$$\text{MRR} = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{\text{rank}_i}$$

- Suppose we have recommended 3 movies to a user, say A, B, C in the given order, but the user only liked movie C. As the rank of movie C is 3, the reciprocal rank will be 1/3
- For multiple recommendations, the Mean Reciprocal Rank is the mean of all reciprocal ranks.
- Larger the mean reciprocal rank, better the recommendations

Average Precision

- Average Precision (*AP*) is a ranked precision metric that places emphasis on highly ranked correct predictions (hits)
- Essentially it is the average of precision values determined after each successful prediction, i.e.
- If a relevant document never gets retrieved, we assume the precision to be 0

Rank	Hit?
1	
2	X
3	X
4	X
5	

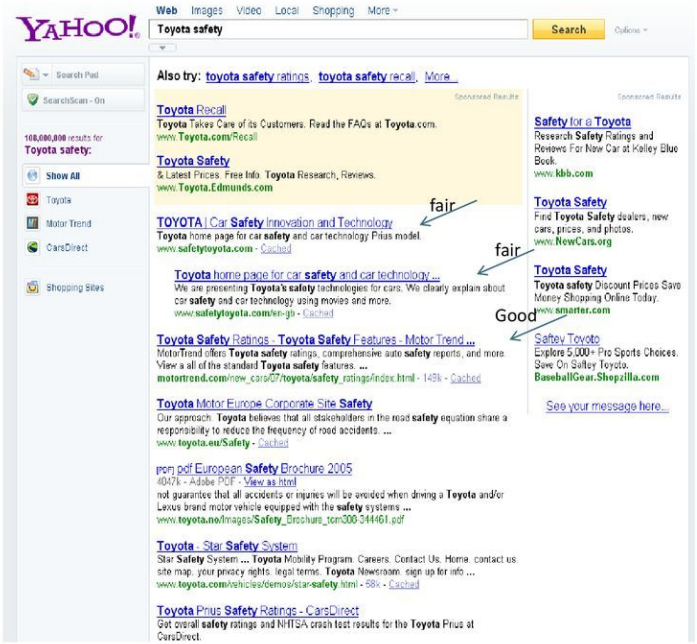
$$AP = \frac{1}{3}\left(\frac{1}{2} + \frac{2}{3} + \frac{3}{4}\right) = \frac{23}{36} \approx 0.639$$

$$AP = \frac{1}{3}\left(\frac{1}{1} + \frac{2}{4} + \frac{3}{5}\right) = \frac{21}{30} = 0.7$$

Rank	Hit?
1	X
2	
3	
4	X
5	X

Analytics Vidhya

Beyond Binary Relevance



Analytics Vidhya

Normalised Discounted Cumulative Gain

- Discounted cumulative gain (DCG)
  - Logarithmic reduction factor

$$DCG_{pos} = rel_1 + \sum_{i=2}^{pos} \frac{rel_i}{\log_2 i}$$

Where:

- *pos* refers to the position up to which relevance is accumulated
- *rel* returns the relevance of recommendation at position *i*

- Idealized discounted cumulative gain (IDCG)
  - Assumption that items are ordered by decreasing relevance

- Normalized discounted cumulative gain (nDCG)
  - Normalized to the interval [0..1]

$$IDCG_{pos} = rel_1 + \sum_{i=2}^{|A|} \frac{rel_i}{\log_2 i}$$

$$nDCG_{pos} \frac{DCG_{pos}}{IDCG_{pos}}$$

Analytics Vidhya

- **Let's say there are 10 ranked movies on 0-3 relevance scale:**
  - 3, 2, 3, 0, 0, 1, 2, 2, 3, 0
- **Discounted Gain**
  - 3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0

  = 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0
- **Discounted Cumulative Gain**
  - 3, 5, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61
- **Inverse Discounted Cumulative Gain**
  - 3, 3, 3, 2, 2, 2, 1, 0, 0, 0

  = 3, 3/1, 3/1.59, 2/2, 2/2.32, 2/2.59, 1/2.81, 0, 0, 0

  = 3, 3, 1.89, 1, 0.86, 0.77, 0.36, 0, 0, 0

  Cumulative □ 3, 6, 7.89, 8.75, 9.52, 9.88, 9.88, 9.88

NDCG = DCG/IDCG

1, 0.83, 0.87, 0.832, 0.808, 0.88, 0.97, 0.97

Analytics Vidhya

NDCG: Example 2

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$DCG_5 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} = 2.13$$

$$IDCG_5 = 1 + \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 2.63$$

$$nDCG_5 \frac{DCG_5}{IDCG_5} \approx 0.81$$

Analytics Vidhya

Online vs Offline Experimentation

Offline experimentation	Online experimentation
Ratings, transactions	Ratings, feedback
Historic session (not all recommended items are rated)	Live interaction (all recommended items are rated)
Ratings of unrated items unknown, but interpreted as “bad” (default assumption, user tend to rate only good items)	“Good/bad” ratings of not recommended items are unknown
If default assumption does not hold: True positives may be too small False negatives may be too small	False/true negatives cannot be determined

Analytics Vidhya

Beyond Accuracy

- Understanding that good recommendation accuracy alone does not give users of recommender systems an effective and satisfying experience
- For instance, a recommender might achieve high accuracy by only computing predictions for easy-to-predict items—but those are the very items for which users are least likely to need predictions
- Coverage:
  - The coverage of a recommender system is a measure of the domain of items in the system over which the system can form predictions or make recommendations
  - Coverage can be most directly defined on predictions by asking “What percentage of items can this recommender form predictions for?
  - What percentage of available items does this recommender ever recommend to users? – more popular with ecommerce
  - Must be measured in combination with accuracy to prevent bogus predictions

Beyond Accuracy

- Obvious recommendations have two disadvantages:
  - Customers who are interested in those products have already purchased them
  - We do not need recommender systems to tell them which products are popular overall.
- Novelty
  - recommendation system that simply recommends movies that were directed by the user's favorite director
  - If the system recommends a movie that the user wasn't aware of, the movie will be novel, but probably not serendipitous.
  - A simple modification is to create a list of "obvious" recommendations, and remove the obvious ones from each recommendation list before presenting it to users.
- Serendipity
  - a recommender that recommends a movie by a new director is more likely to provide serendipitous recommendations

Discussion & Summary

- Focus on how to perform empirical evaluations on historical datasets
- Discussion about different methodologies and metrics for measuring the accuracy or coverage of recommendations.
- Overview of which research designs are commonly used in practice.
- From a technical point of view, measuring the accuracy of predictions is a well accepted evaluation goal
  - but other aspects that may potentially impact the overall effectiveness of a recommendation system remain largely underdeveloped.

Notebook:
http://localhost:8888/notebooks/Desktop/SBSA-2020/22.%20Recommender%20Systems/4.%20Evaluation%20metrics/Evaluation%20Metrics%20with%20CF.ipynb

Analytics Vidhya