The Need for Training in Big Data: Experiences and Case Studies

Guy Lebanon

Amazon


SAY BIG DATA
ONE MORE TIME
memegenerator.net

Background and Disclaimer

All opinions are mine; other perspectives are legitimate.

Based on my experience as a

- professor at Purdue
- professor at Georgia Tech
- part time scientist at Yahoo
- visiting scientist at Google
- manager of scientists at Amazon.

Extracting Meaning from Big Data - Skills

Extracting meaning from big data requires skills in:

- Computing and software engineering
- Machine Learning, statistics, and optimization
- Product sense and careful experimentation

All three areas are important. It is rare to find people with skills in all three areas.

Companies are fighting tooth and nail over the few people with skills in all three categories

Case Study: Recommendation Systems

Recommendation systems technology is essential to many companies:

- Netflix (recommending movies)
- Amazon (recommending products)
- Pandora (recommending music)
- Facebook (recommending friends, “recommending” posts in stream)
- Google (“recommending” ads)

Huge progress has been made, but we are still in the early stages. To overcome the many remaining challenges we need people with skills in computing, machine learning, product sense.

Recommendation Systems: Historical Perspective

Ancient history: 1990s

Given some pairs of (user,item) ratings, predict ratings of other pairs.

Similarity based recommendation systems:

$$R_{ui} = c_u + \sum_{u'} \alpha_{u,u'} R_{u'i}$$

where $\alpha_{u,u'}$ expresses similarity between users u, u', or

$$R_{ui} = c_i + \sum_{i'} \alpha_{i,i'} R_{ui'}$$

where $\alpha_{i,i'}$ expresses similarity between users i, i'.

Historical Perspective

Middle Ages: early 2000s

Matrix Completion Perspective

- Given $M_{a_1,b_1},\ldots,M_{a_m,b_m}$ predict unobserved entries of $M\in\mathbb{R}^{n_1\times n_2}$
- $M_{a,b}$ is the rating given by user a to item b.
- Many possible completions of M consistent with training data
- Standard practice: favor low-rank ("simple") completions of M

$$M = UV^\top \in \mathbb{R}^{n_1\times n_2}, \qquad \begin{array}{l} U\in\mathbb{R}^{n_1\times r} \\ V\in\mathbb{R}^{n_2\times r} \\ r\ll\min(n_1,n_2) \end{array}$$

$$(U,V) = \arg\min_{U,V} \sum_{(a,b)\in\mathtt{A}} ([UV^T]_{a,b} - M_{a,b})^2$$

Historical Perspective

Middle Ages: early 2000s

Matrix Completion Perspective

- Given $M_{a_1,b_1}, \ldots, M_{a_m,b_m}$ predict unobserved entries of $M \in \mathbb{R}^{n_1 \times n_2}$
- $M_{a,b}$ is the rating given by user a to item b.
- Many possible completions of M consistent with training data
- Standard practice: favor low-rank ("simple") completions of M

$$M = UV^\top \in \mathbb{R}^{n_1 \times n_2}, \qquad \begin{array}{l} U \in \mathbb{R}^{n_1 \times r} \\ V \in \mathbb{R}^{n_2 \times r} \\ r \ll \min(n_1, n_2) \end{array}$$

$$(U, V) = \arg\min_{U,V} \sum_{(a,b)\in \mathtt{A}} ([UV^T]_{a,b} - M_{a,b})^2$$

Historical Perspective

Middle Ages: early 2000s

Matrix Completion Perspective

- Given $M_{a_1,b_1}, \dots, M_{a_m,b_m}$ predict unobserved entries of $M \in \mathbb{R}^{n_1 \times n_2}$
- $M_{a,b}$ is the rating given by user a to item b.
- Many possible completions of M consistent with training data
- Standard practice: favor low-rank ("simple") completions of M

$$M = UV^\top \in \mathbb{R}^{n_1 \times n_2}, \qquad \begin{aligned} U &\in \mathbb{R}^{n_1 \times r} \\ V &\in \mathbb{R}^{n_2 \times r} \\ r &\ll \min(n_1, n_2) \end{aligned}$$

$$(U, V) = \arg\min_{U,V} \sum_{(a,b)\in \mathtt{A}} ([UV^T]_{a,b} - M_{a,b})^2$$

Historical Perspective

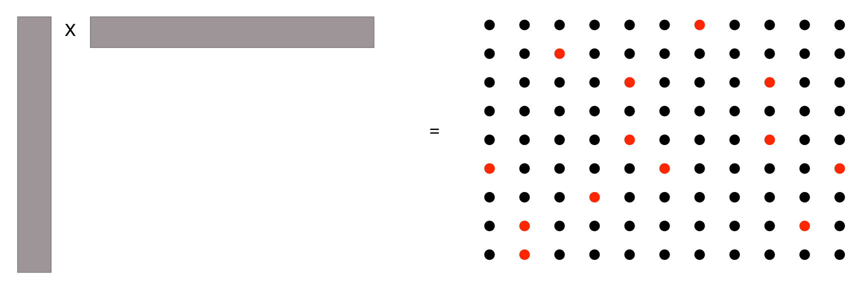
Middle Ages: early 2000s

Matrix Completion Perspective

- Given $M_{a_1,b_1}, \ldots, M_{a_m,b_m}$ predict unobserved entries of $M \in \mathbb{R}^{n_1 \times n_2}$
- $M_{a,b}$ is the rating given by user a to item b.
- Many possible completions of M consistent with training data
- Standard practice: favor low-rank ("simple") completions of M

$$M = UV^\top \in \mathbb{R}^{n_1 \times n_2}, \qquad \begin{aligned} &U \in \mathbb{R}^{n_1 \times r} \\ &V \in \mathbb{R}^{n_2 \times r} \\ &r \ll \min(n_1, n_2) \end{aligned}$$

$$(U, V) = \arg\min_{U,V} \sum_{(a,b) \in \mathtt{A}} ([UV^T]_{a,b} - M_{a,b})^2$$


X
=

Part 1: Skills needed to construct modern recommendation system in industry

Part 2: Skills needed to innovate in recommendation system in industry setting

Based on my experience working on recommendation systems at Google and Amazon

Constructing MF Models (ML)

ML challenge: non-linear optimization over a high dimensional vector space with big-data

Constructing modern matrix factorization models (in industry) requires:

- understanding non-linear optimization algorithms and how to implement them, including online methods such as stochastic gradient descent
- understanding practical tips and tricks such as momentum, step size selection, etc.
- understanding major issues in machine learning such as overfitting

Constructing MF Models (ML)

ML challenge: non-linear optimization over a high dimensional vector space with big-data

Constructing modern matrix factorization models (in industry) requires:

- understanding non-linear optimization algorithms and how to implement them, including online methods such as stochastic gradient descent
- understanding practical tips and tricks such as momentum, step size selection, etc.
- understanding major issues in machine learning such as overfitting

Constructing MF Models (ML)

ML challenge: non-linear optimization over a high dimensional vector space with big-data

Constructing modern matrix factorization models (in industry) requires:

- understanding non-linear optimization algorithms and how to implement them, including online methods such as stochastic gradient descent
- understanding practical tips and tricks such as momentum, step size selection, etc.
- understanding major issues in machine learning such as overfitting

Constructing MF Models (Computing)

Computing challenge: getting data, processing big data in train time, maintaining SLA in test time, service oriented architecture

Constructing modern matrix factorization models (in industry) requires:

- writing production code in C++ or Java
- getting data: querying SQL/NoSQL databases
- processing data: parallel and distributed computing for big data (e.g., Map-Reduce)
- software engineering practices: version control, code documentation, build tools, unit tests, integration tests
- *very efficient* scoring in serve time (SLA)
- constructing *multiple* software services that communicate with each other

Constructing MF Models (Computing)

Computing challenge: getting data, processing big data in train time, maintaining SLA in test time, service oriented architecture

Constructing modern matrix factorization models (in industry) requires:

- writing production code in C++ or Java
- getting data: querying SQL/NoSQL databases
- processing data: parallel and distributed computing for big data (e.g., Map-Reduce)
- software engineering practices: version control, code documentation, build tools, unit tests, integration tests
- *very efficient* scoring in serve time (SLA)
- constructing *multiple* software services that communicate with each other

Constructing MF Models (Computing)

Computing challenge: getting data, processing big data in train time, maintaining SLA in test time, service oriented architecture

Constructing modern matrix factorization models (in industry) requires:

- writing production code in C++ or Java
- getting data: querying SQL/NoSQL databases
- processing data: parallel and distributed computing for big data (e.g., Map-Reduce)
- software engineering practices: version control, code documentation, build tools, unit tests, integration tests
- *very efficient* scoring in serve time (SLA)
- constructing *multiple* software services that communicate with each other

Constructing MF Models (Computing)

Computing challenge: getting data, processing big data in train time, maintaining SLA in test time, service oriented architecture

Constructing modern matrix factorization models (in industry) requires:

- writing production code in C++ or Java
- getting data: querying SQL/NoSQL databases
- processing data: parallel and distributed computing for big data (e.g., Map-Reduce)
- software engineering practices: version control, code documentation, build tools, unit tests, integration tests
- *very efficient* scoring in serve time (SLA)
- constructing *multiple* software services that communicate with each other

Constructing MF Models (Computing)

Computing challenge: getting data, processing big data in train time, maintaining SLA in test time, service oriented architecture

Constructing modern matrix factorization models (in industry) requires:

- writing production code in C++ or Java
- getting data: querying SQL/NoSQL databases
- processing data: parallel and distributed computing for big data (e.g., Map-Reduce)
- software engineering practices: version control, code documentation, build tools, unit tests, integration tests
- *very efficient* scoring in serve time (SLA)
- constructing *multiple* software services that communicate with each other

Constructing MF Models (Computing)

Computing challenge: getting data, processing big data in train time, maintaining SLA in test time, service oriented architecture

Constructing modern matrix factorization models (in industry) requires:

- writing production code in C++ or Java
- getting data: querying SQL/NoSQL databases
- processing data: parallel and distributed computing for big data (e.g., Map-Reduce)
- software engineering practices: version control, code documentation, build tools, unit tests, integration tests
- *very efficient* scoring in serve time (SLA)
- constructing *multiple* software services that communicate with each other

Constructing MF Models (Product)

Product challenge: addressing (very important) details, putting together a meaningful and practical evaluation process (A/B test)

- Design an online evaluation process that measures business goals (customer engagement, sales), and that can reach statistical significant within reasonable time, and yet avoid breaking customer trust by testing poor models
- train based on recent behavior or all available history?
- emphasize popular items or long tail?
- how many items should be shown?
- should some items be omitted (adult or otherwise controversial content)?
- how can the product be modified so that more training data is collected or solicited from users?

Constructing MF Models (Product)

Product challenge: addressing (very important) details, putting together a meaningful and practical evaluation process (A/B test)

- Design an online evaluation process that measures business goals (customer engagement, sales), and that can reach statistical significant within reasonable time, and yet avoid breaking customer trust by testing poor models
- train based on recent behavior or all available history?
- emphasize popular items or long tail?
- how many items should be shown?
- should some items be omitted (adult or otherwise controversial content)?
- how can the product be modified so that more training data is collected or solicited from users?

Constructing MF Models (Product)

Product challenge: addressing (very important) details, putting together a meaningful and practical evaluation process (A/B test)

- Design an online evaluation process that measures business goals (customer engagement, sales), and that can reach statistical significant within reasonable time, and yet avoid breaking customer trust by testing poor models
- train based on recent behavior or all available history?
- emphasize popular items or long tail?
- how many items should be shown?
- should some items be omitted (adult or otherwise controversial content)?
- how can the product be modified so that more training data is collected or solicited from users?

Constructing MF Models (Product)

Product challenge: addressing (very important) details, putting together a meaningful and practical evaluation process (A/B test)

- Design an online evaluation process that measures business goals (customer engagement, sales), and that can reach statistical significant within reasonable time, and yet avoid breaking customer trust by testing poor models
- train based on recent behavior or all available history?
- emphasize popular items or long tail?
- how many items should be shown?
- should some items be omitted (adult or otherwise controversial content)?
- how can the product be modified so that more training data is collected or solicited from users?

Constructing MF Models (Product)

Product challenge: addressing (very important) details, putting together a meaningful and practical evaluation process (A/B test)

- Design an online evaluation process that measures business goals (customer engagement, sales), and that can reach statistical significant within reasonable time, and yet avoid breaking customer trust by testing poor models
- train based on recent behavior or all available history?
- emphasize popular items or long tail?
- how many items should be shown?
- should some items be omitted (adult or otherwise controversial content)?
- how can the product be modified so that more training data is collected or solicited from users?

Constructing MF Models (Product)

Product challenge: addressing (very important) details, putting together a meaningful and practical evaluation process (A/B test)

- Design an online evaluation process that measures business goals (customer engagement, sales), and that can reach statistical significant within reasonable time, and yet avoid breaking customer trust by testing poor models
- train based on recent behavior or all available history?
- emphasize popular items or long tail?
- how many items should be shown?
- should some items be omitted (adult or otherwise controversial content)?
- how can the product be modified so that more training data is collected or solicited from users?

Innovation in Recommendation Systems

Difficulties in conducting academic research on practical recommendation systems:

- accuracy in offline score prediction is poorly correlated with industry metrics (user engagement, sales)
- lack of datasets that reflect the practical scenario in industry (explicit vs. implicit data, context, content-based filtering vs. collaborative filtering)

The biggest potential for impactful innovation lies in: (a) understanding the data, (b) evaluation.

Historical Perspective

Industrial Revolution - early 2000s: Netflix Competition

Netflix released a dataset of “anonymized” (user, item) ratings and a one million dollar prize to top performing team.

- Data was significantly larger than what was previously available. This introduced some scalability arguments into the modeling process.
- Introduced a common evaluation methodology and a clear way of defining a winner.
- Huge boost to the field in terms of number of research papers and interest from the academic community.
- Ended in an embarrassment as researchers from UT Austin de-anonymized the data by successfully joining it with IMDB records.
- Netflix took the data down and is facing a lawsuit.

Historical Perspective

Industrial Revolution - early 2000s: Netflix Competition

Netflix released a dataset of “anonymized” (user, item) ratings and a one million dollar prize to top performing team.

- Data was significantly larger than what was previously available. This introduced some scalability arguments into the modeling process.
- Introduced a common evaluation methodology and a clear way of defining a winner.
- Huge boost to the field in terms of number of research papers and interest from the academic community.
- Ended in an embarrassment as researchers from UT Austin de-anonymized the data by successfully joining it with IMDB records.
- Netflix took the data down and is facing a lawsuit.

Historical Perspective

Industrial Revolution - early 2000s: Netflix Competition

Netflix released a dataset of "anonymized" (user, item) ratings and a one million dollar prize to top performing team.

- Data was significantly larger than what was previously available. This introduced some scalability arguments into the modeling process.
- Introduced a common evaluation methodology and a clear way of defining a winner.
- Huge boost to the field in terms of number of research papers and interest from the academic community.
- Ended in an embarrassment as researchers from UT Austin de-anonymized the data by successfully joining it with IMDB records.
- Netflix took the data down and is facing a lawsuit.

Historical Perspective

Industrial Revolution - early 2000s: Netflix Competition

Netflix released a dataset of “anonymized” (user, item) ratings and a one million dollar prize to top performing team.

- Data was significantly larger than what was previously available. This introduced some scalability arguments into the modeling process.
- Introduced a common evaluation methodology and a clear way of defining a winner.
- Huge boost to the field in terms of number of research papers and interest from the academic community.
- Ended in an embarrassment as researchers from UT Austin de-anonymized the data by successfully joining it with IMDB records.
- Netflix took the data down and is facing a lawsuit.

Historical Perspective

Industrial Revolution - early 2000s: Netflix Competition

Netflix released a dataset of "anonymized" (user, item) ratings and a one million dollar prize to top performing team.

- Data was significantly larger than what was previously available. This introduced some scalability arguments into the modeling process.
- Introduced a common evaluation methodology and a clear way of defining a winner.
- Huge boost to the field in terms of number of research papers and interest from the academic community.
- Ended in an embarrassment as researchers from UT Austin de-anonymized the data by successfully joining it with IMDB records.
- Netflix took the data down and is facing a lawsuit.

Evaluation

- Standard evaluation methods based on rating prediction have very low correlation with what matters in industry: revenue and user engagement.
- Unfortunately, this is also true for evaluation methods based on NDCG or Precision@k.
- Traditional measures partition data into train and test, train models on train set, and measure precision or a related metric on the held out test set.
- In reality, what really matters are the action taken by users when presented with a specific recommendation.

Evaluation

- Standard evaluation methods based on rating prediction have very low correlation with what matters in industry: revenue and user engagement.
- Unfortunately, this is also true for evaluation methods based on NDCG or Precision@k.
- Traditional measures partition data into train and test, train models on train set, and measure precision or a related metric on the held out test set.
- In reality, what really matters are the action taken by users when presented with a specific recommendation.

Evaluation

- Standard evaluation methods based on rating prediction have very low correlation with what matters in industry: revenue and user engagement.
- Unfortunately, this is also true for evaluation methods based on NDCG or Precision@k.
- Traditional measures partition data into train and test, train models on train set, and measure precision or a related metric on the held out test set.
- In reality, what really matters are the action taken by users when presented with a specific recommendation.

Evaluation

- Standard evaluation methods based on rating prediction have very low correlation with what matters in industry: revenue and user engagement.
- Unfortunately, this is also true for evaluation methods based on NDCG or Precision@k.
- Traditional measures partition data into train and test, train models on train set, and measure precision or a related metric on the held out test set.
- In reality, what really matters are the action taken by users when presented with a specific recommendation.

Evaluation

Traditional measures assume:

- action of users in train and test sets are independent of their context
- choices of what to view or buy are independent of options displayed to the user.
- patently false when number of recommended items is large; users make choices based on small number of alternative they consider at the moment.

As a result:

- traditional measures based on train/test set partition do not correlate well with revenue or user engagement in an A/B test.
- Method that perform well in A/B test and in launch may not be the winning methods in studies that use traditional evaluation.

Evaluation

Traditional measures assume:

- action of users in train and test sets are independent of their context
- choices of what to view or buy are independent of options displayed to the user.
- patently false when number of recommended items is large; users make choices based on small number of alternative they consider at the moment.

As a result:

- traditional measures based on train/test set partition do not correlate well with revenue or user engagement in an A/B test.
- Method that perform well in A/B test and in launch may not be the winning methods in studies that use traditional evaluation.

Evaluation

Traditional measures assume:

- action of users in train and test sets are independent of their context
- choices of what to view or buy are independent of options displayed to the user.
- patently false when number of recommended items is large; users make choices based on small number of alternative they consider at the moment.

As a result:

- traditional measures based on train/test set partition do not correlate well with revenue or user engagement in an A/B test.
- Method that perform well in A/B test and in launch may not be the winning methods in studies that use traditional evaluation.

Evaluation

Traditional measures assume:

- action of users in train and test sets are independent of their context
- choices of what to view or buy are independent of options displayed to the user.
- patently false when number of recommended items is large; users make choices based on small number of alternative they consider at the moment.

As a result:

- traditional measures based on train/test set partition do not correlate well with revenue or user engagement in an A/B test.
- Method that perform well in A/B test and in launch may not be the winning methods in studies that use traditional evaluation.

Evaluation

Traditional measures assume:

- action of users in train and test sets are independent of their context
- choices of what to view or buy are independent of options displayed to the user.
- patently false when number of recommended items is large; users make choices based on small number of alternative they consider at the moment.

As a result:

- traditional measures based on train/test set partition do not correlate well with revenue or user engagement in an A/B test.
- Method that perform well in A/B test and in launch may not be the winning methods in studies that use traditional evaluation.

New Evaluation Methodology

- Study correlations between existing evaluation methods and increased user engagement in A/B test
- New offline evaluation methods that take context into account (what was displayed to the user when they took their action) may have higher correlation with real world success.
- Potential breakthrough: efficient search in the space of model possibilities that maximizes A/B test performance
    - human gradient descent?
    - challenges: what happens to statistical significance, p-values?

New Evaluation Methodology

- Study correlations between existing evaluation methods and increased user engagement in A/B test
- New offline evaluation methods that take context into account (what was displayed to the user when they took their action) may have higher correlation with real world success.
- Potential breakthrough: efficient search in the space of model possibilities that maximizes A/B test performance
  - human gradient descent?
  - challenges: what happens to statistical significance, p-values?

New Evaluation Methodology

- Study correlations between existing evaluation methods and increased user engagement in A/B test
- New offline evaluation methods that take context into account (what was displayed to the user when they took their action) may have higher correlation with real world success.
- Potential breakthrough: efficient search in the space of model possibilities that maximizes A/B test performance
  - human gradient descent?
  - challenges: what happens to statistical significance, p-values?

New Evaluation Methodology

- Study correlations between existing evaluation methods and increased user engagement in A/B test
- New offline evaluation methods that take context into account (what was displayed to the user when they took their action) may have higher correlation with real world success.
- Potential breakthrough: efficient search in the space of model possibilities that maximizes A/B test performance
  - human gradient descent?
  - challenges: what happens to statistical significance, p-values?

New Evaluation Methodology

- Study correlations between existing evaluation methods and increased user engagement in A/B test
- New offline evaluation methods that take context into account (what was displayed to the user when they took their action) may have higher correlation with real world success.
- Potential breakthrough: efficient search in the space of model possibilities that maximizes A/B test performance
  - human gradient descent?
  - challenges: what happens to statistical significance, p-values?

Datasets

- Few public recommendation systems datasets are available, and vast majority of academic research focuses on modeling these datasets.
- By far the most popular task is predict five star ratings based on Movielens/EachMovie/Netflix dataset.
- This has led to an obsession with (often imaginary) incremental gains in modeling these datasets.

Datasets

- Few public recommendation systems datasets are available, and vast majority of academic research focuses on modeling these datasets.
- By far the most popular task is predict five star ratings based on Movielens/EachMovie/Netflix dataset.
- This has led to an obsession with (often imaginary) incremental gains in modeling these datasets.

Datasets

- Few public recommendation systems datasets are available, and vast majority of academic research focuses on modeling these datasets.
- By far the most popular task is predict five star ratings based on Movielens/EachMovie/Netflix dataset.
- This has led to an obsession with (often imaginary) incremental gains in modeling these datasets.

Datasets

Predicting five star ratings for Netflix data has little relevance to real world applications

- In real world we don't care about predicting ratings; we care about increasing user engagement and revenue.
- In real world we know more information on users and items (not just user ID and item ID). Recommendation systems can use additional information such as user profile, address, etc.
- In real world we often deal with implicit binary ratings (impressions, purchases) rather than explicit five star ratings. In explicit ratings, training data consists of both positive and negative feedback while in implicit ratings all training data is positive.
- Public datasets are typically much smaller than industry data, making it impossible to seriously explore scalability issues.

Datasets

Predicting five star ratings for Netflix data has little relevance to real world applications

- In real world we don't care about predicting ratings; we care about increasing user engagement and revenue.
- In real world we know more information on users and items (not just user ID and item ID). Recommendation systems can use additional information such as user profile, address, etc.
- In real world we often deal with implicit binary ratings (impressions, purchases) rather than explicit five star ratings. In explicit ratings, training data consists of both positive and negative feedback while in implicit ratings all training data is positive.
- Public datasets are typically much smaller than industry data, making it impossible to seriously explore scalability issues.

Datasets

Predicting five star ratings for Netflix data has little relevance to real world applications

- In real world we don't care about predicting ratings; we care about increasing user engagement and revenue.
- In real world we know more information on users and items (not just user ID and item ID). Recommendation systems can use additional information such as user profile, address, etc.
- In real world we often deal with implicit binary ratings (impressions, purchases) rather than explicit five star ratings. In explicit ratings, training data consists of both positive and negative feedback while in implicit ratings all training data is positive.
- Public datasets are typically much smaller than industry data, making it impossible to seriously explore scalability issues.

Datasets

Predicting five star ratings for Netflix data has little relevance to real world applications

- In real world we don't care about predicting ratings; we care about increasing user engagement and revenue.
- In real world we know more information on users and items (not just user ID and item ID). Recommendation systems can use additional information such as user profile, address, etc.
- In real world we often deal with implicit binary ratings (impressions, purchases) rather than explicit five star ratings. In explicit ratings, training data consists of both positive and negative feedback while in implicit ratings all training data is positive.
- Public datasets are typically much smaller than industry data, making it impossible to seriously explore scalability issues.

Implicit Ratings

- Treat positive ratings (impressions, purchases) as a rating of 1 and all other signal as a rating of 0.
- Use ranked loss minimization over the above signal (favor ordering all impressions/purchases over lack of impressions/purchases)
- When number of items is large, negative signal formed by lack of impressions/purchases is overwhelming in size.
  - Solution: Sample items marking lack of impressions/purchases and apply loss function only to that sample.

This assumes that whatever was not viewed or bought was not done so on purpose.

Implicit Ratings

- Treat positive ratings (impressions, purchases) as a rating of 1 and all other signal as a rating of 0.
- Use ranked loss minimization over the above signal (favor ordering all impressions/purchases over lack of impressions/purchases)
- When number of items is large, negative signal formed by lack of impressions/purchases is overwhelming in size.
    - Solution: Sample items marking lack of impressions/purchases and apply loss function only to that sample.

This assumes that whatever was not viewed or bought was not done so on purpose.

Implicit Ratings

- Treat positive ratings (impressions, purchases) as a rating of 1 and all other signal as a rating of 0.
- Use ranked loss minimization over the above signal (favor ordering all impressions/purchases over lack of impressions/purchases)
- When number of items is large, negative signal formed by lack of impressions/purchases is overwhelming in size.
  - Solution: Sample items marking lack of impressions/purchases and apply loss function only to that sample.

This assumes that whatever was not viewed or bought was not done so on purpose.

Implicit Ratings

- Treat positive ratings (impressions, purchases) as a rating of 1 and all other signal as a rating of 0.
- Use ranked loss minimization over the above signal (favor ordering all impressions/purchases over lack of impressions/purchases)
- When number of items is large, negative signal formed by lack of impressions/purchases is overwhelming in size.
  - Solution: Sample items marking lack of impressions/purchases and apply loss function only to that sample.

This assumes that whatever was not viewed or bought was not done so on purpose.