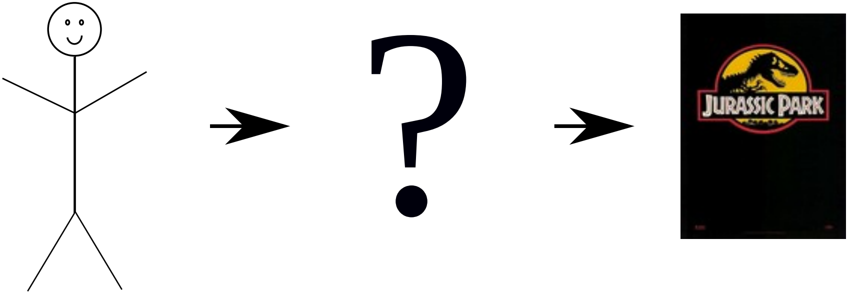
How Many Bits Per Rating

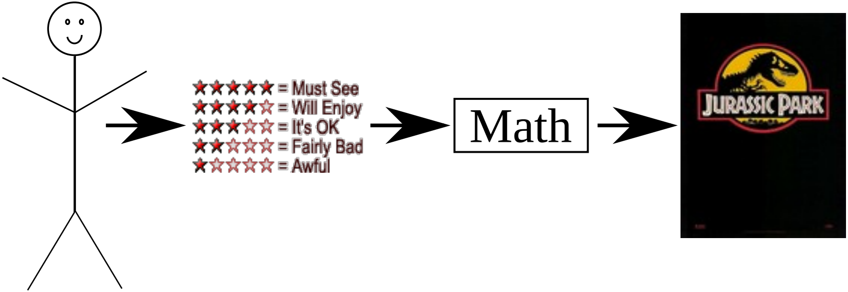
Daniel Kluver[1], Tien T. Nguyen[1], Michael Ekstrand[1], Shilad Sen[2], John Riedl[1]

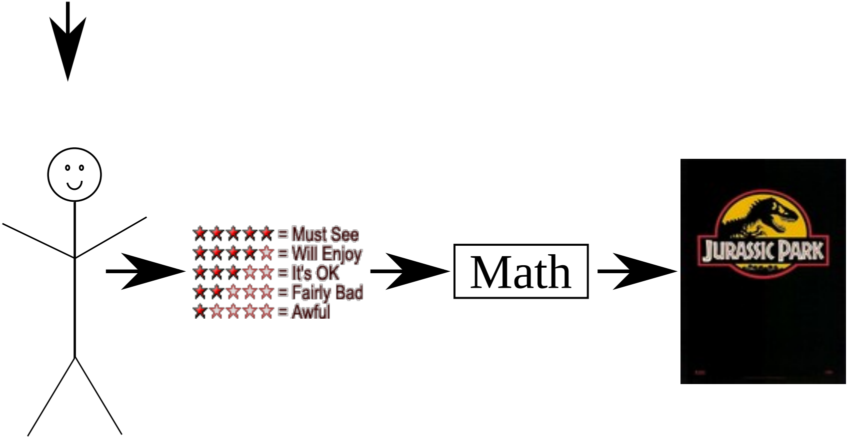
[1] GroupLens Research
Dept. of Computer Science and Engineering
University of Minnesota

[2] Math, Stats, and Computer Science Dept.
Macalester College

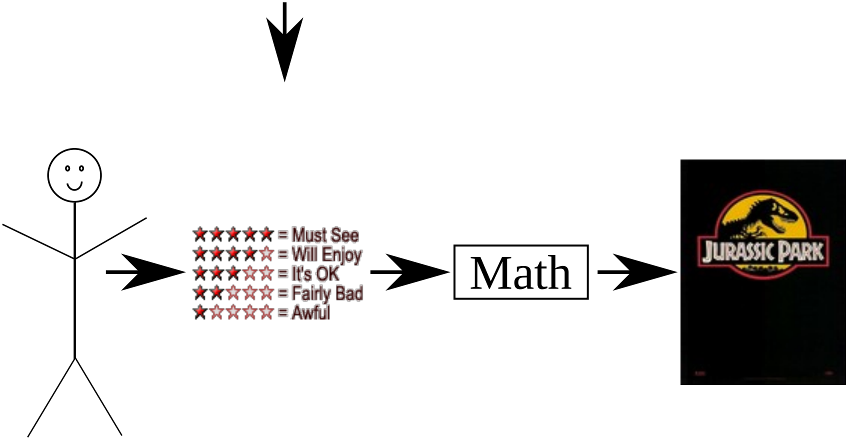
September 11, 2012

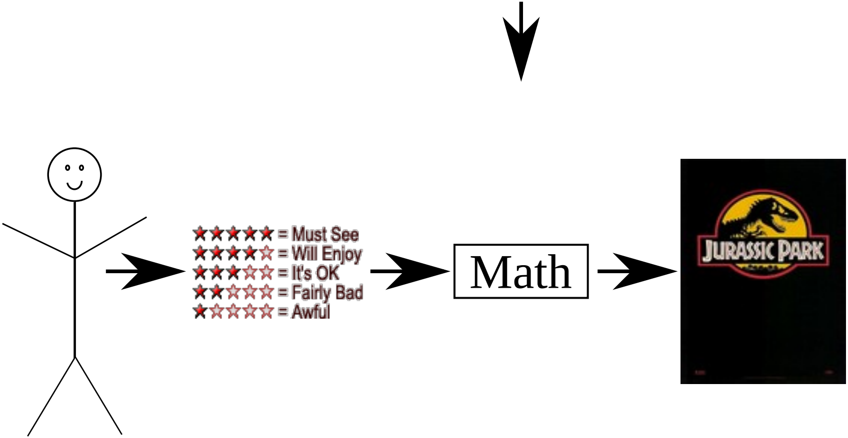

?
JURASSIC PARK

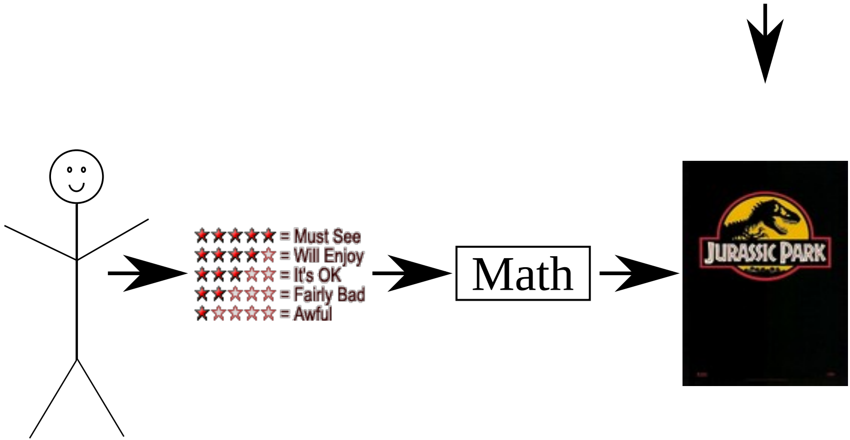

= Must See
= Will Enjoy
= It's OK
= Fairly Bad
= Awful
Math
JURASSIC PARK



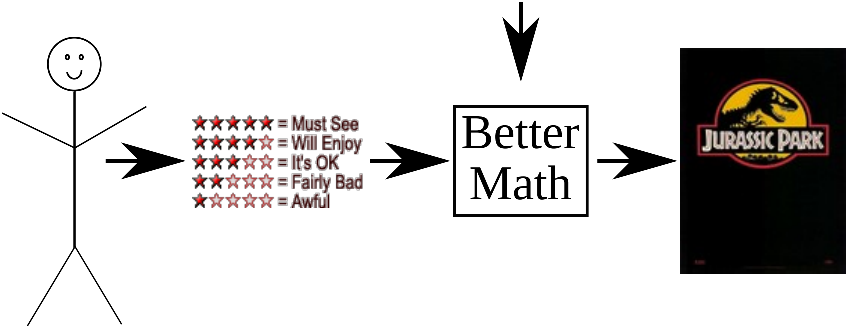
★★★★★ = Must See
★★★★☆ = Will Enjoy
★★★☆☆ = It's OK
★★☆☆☆ = Fairly Bad
★☆☆☆☆ = Awful
Math
JURASSIC PARK


= Must See
= Will Enjoy
= It's OK
= Fairly Bad
= Awful
Math
JURASSIC PARK

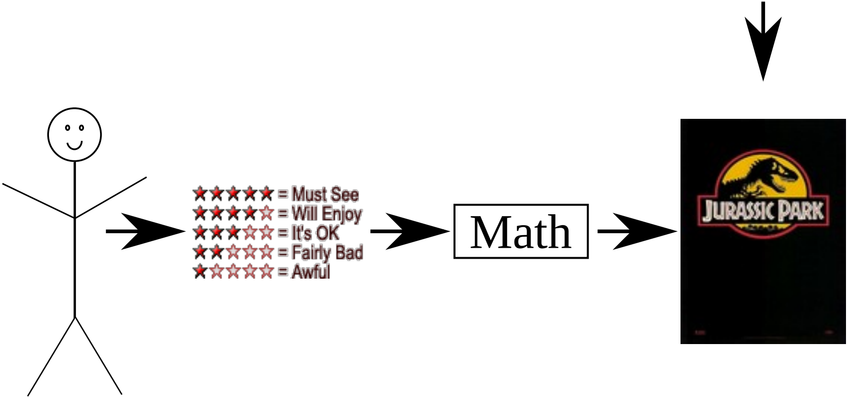

★★★★★ = Must See
★★★★☆ = Will Enjoy
★★★☆☆ = It's OK
★★☆☆☆ = Fairly Bad
★☆☆☆☆ = Awful
Math
JURASSIC PARK



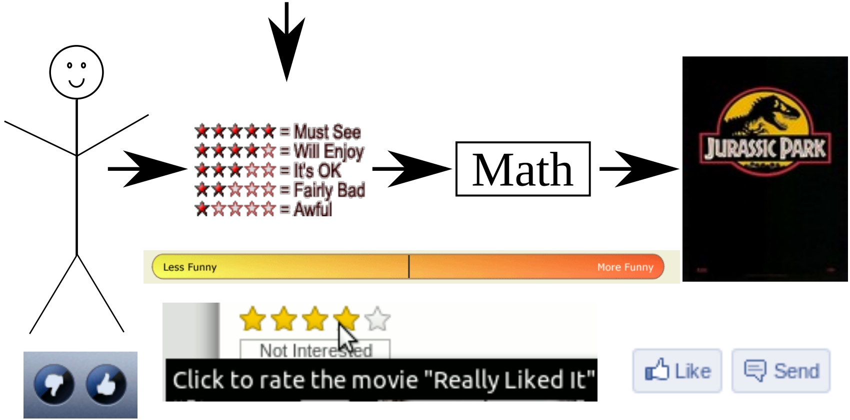
★★★★★ = Must See
★★★★☆ = Will Enjoy
★★★☆☆ = It's OK
★★☆☆☆ = Fairly Bad
★☆☆☆☆ = Awful
Math
JURASSIC PARK



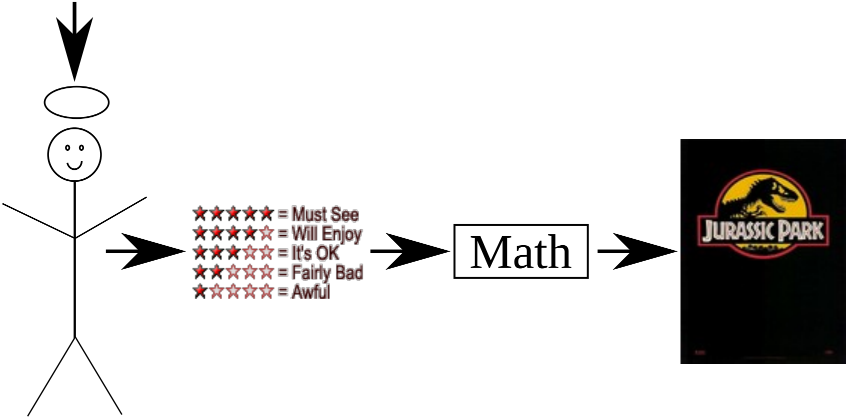
Item-Item, User-User, Matrix Factorization, Feature Weighted Linear Stacking



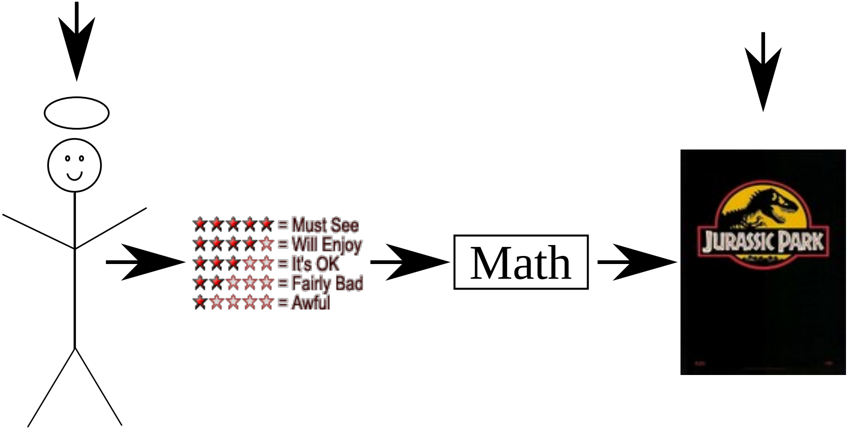
Predict, Recommend, Explain Predictions, Diversify Recomendations



Normalization, Re-rating based Denosing

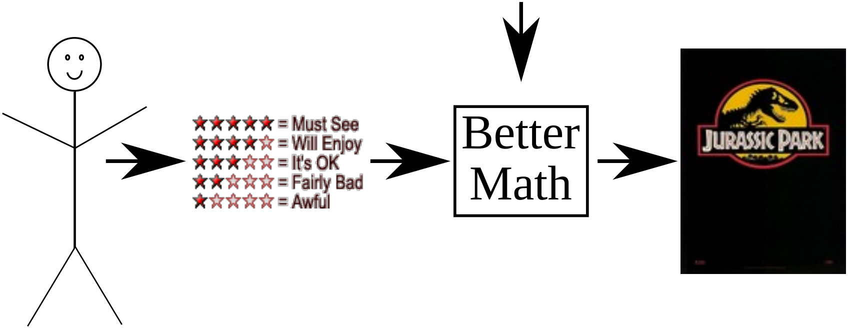


User training, Surgery, Fraud detection, Intercrainial Preference Elicitation



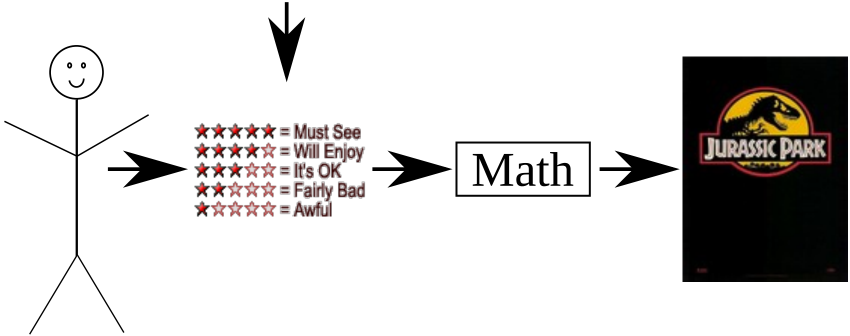
Problem - ?

Better - ?



Better - Prediction Accuracy

Issue - Magic Barrier



Better - ?

Problem - Users rate inconsistantly

Define *Best*

- Looks good
- Makes users happy
- Ratings are fast
- Most information about user preferences

Define *Best*

- Looks good
- Makes users happy
- Ratings are fast
- **Most information about user preferences**

Preference bits A measure of information about user preferences.

Bits per rating Measures how much preference information is contained in ratings.

Bits per second Measures the *efficiency* of an interface at capturing preference bits.

Bits per prediction Measures how much preference information is contained in predictions.

Define *Preference*

Define *Information*

Define *Preference*

The tendency to consistently behave as if you placed value (positive or negative) on something.

Preferences *n.* - The tendency to consistently behave as if you placed *value* on something.

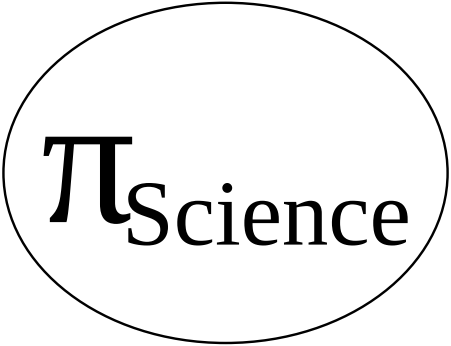
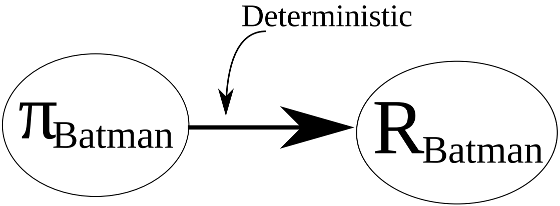
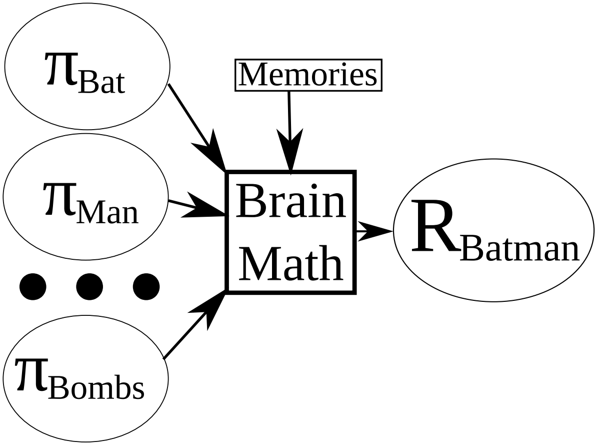


Figure: My value for science is rather large

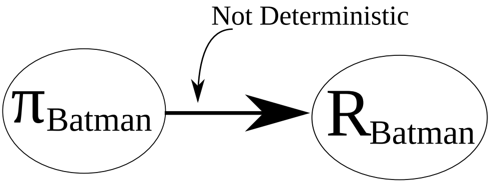
Articulated Values



Basic Values



Partially Articulated Values



- All noise in rating comes from completely hidden context.

Define *Information*

Thanks to Claude Shannon, Information is a solved problem.

Mutual Information

$$I(X;Y) = \sum_x \sum_y P(x,y) log(\frac{P(x,y)}{P(x)P(y)})$$

- Measurement of how much knowing X increases our certainty about Y on average.
- Normally given in *bits*

Bringing it all together

- We can use mutual information to measure how much information anything tells us about user preferences.
- We call this measurement *Preference Bits*.
- If something has a lot of preference bits then it is good at explaining user preference.

Words to know

Preference bits A measure of information about user preferences.

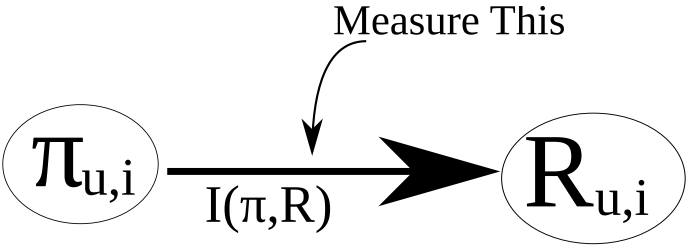
Bits per rating Measures how much preference information is contained in ratings.

Bits per second Measures the *efficiency* of an interface at capturing preference bits.

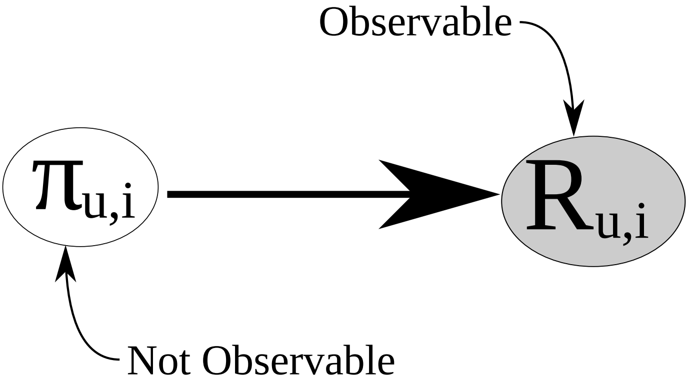
Bits per prediction Measures how much preference information is contained in predictions.

Goal: Measure information entering the recommender.

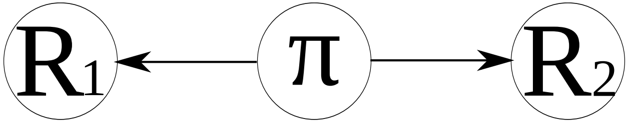
- Ratings($R_{u,i}$) enter the recommender.
- Ratings measure user preferences ($\pi_{u,i}$).
- Therefore we want to measure this:



Goal: Measure information entering the recommender.

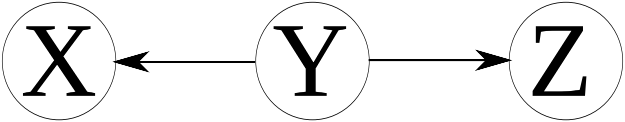


Measuring Input Preference Bits

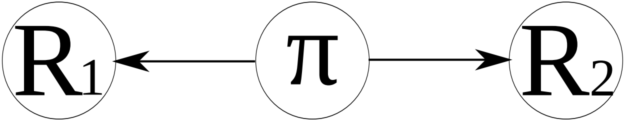


- Prior work solves this problem with two ratings.

Information Processing Inequality



- X and Z are *conditionally independent* given Y
- When this is true, $I(X;Z) \leq I(X;Y)$



- For two conditionally independent re-ratings $I(R_1; R_2) \leq I(R; \pi)$
- We will use this to measure input preference bits.

2 Big assumptions:

- R_1 conditionally independent with R_2 given π
- R_1 and R_2 are generated by the same π

Picking User Interfaces

- Split users between rating interfaces
- Have users rate a bunch of movies
- Some time later, have the users rate the same items
- Compare preference bits between conditions

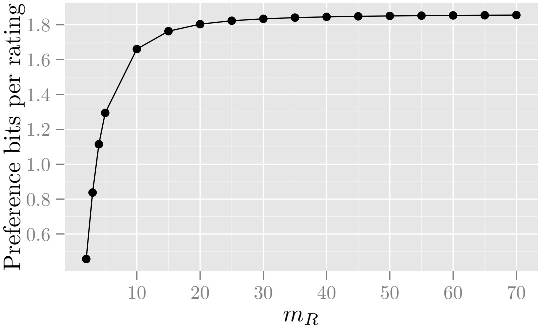
- Split users between rating interfaces
- Have users rate a bunch of movies
- Some time later, have the users rate the same items
- Compare preference bits between conditions
- We haven't run this (yet)

No one else has either

From Cosley et. al. Seeing is believing.

2-point	6-point	10-point
0.423	0.825	0.813

Effect of Rating Scale on Input Preference Bits



- More rating choices, more bits
- Information hits a limit
  - More noise less bits
  - More preference options more bits

Words to know

Preference bits A measure of information about user preferences.

Bits per rating Measures how much preference information is contained in ratings.

Bits per second Measures the *efficiency* of an interface at capturing preference bits.

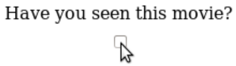
Bits per prediction Measures how much preference information is contained in predictions.

A Problem

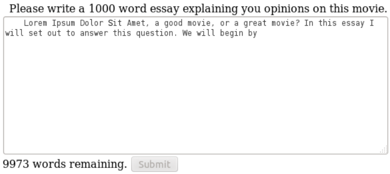
- More rating options $\Rightarrow$ more information
- More rating options $\Rightarrow$ more cognitive load
  More rating options $\Rightarrow$ slower ratings [1]
- slower ratings $\Rightarrow$ less ratings
- less ratings $\Rightarrow$ less information.
- More rating options $\Rightarrow$ less *and* more information?

[1] Sparling et. al. Rating: How difficult is it?

Fast Rating
Low Information



Slow Rating
High Information



- Measuring bits per rating is easy.
- Measuring ratings per second is also easy.
- It turns out measure bits per second is also easy.

$$\frac{\#\ \mathit{Bits}}{1\ \mathit{Rating}} \times \frac{\#\ \mathit{Ratings}}{1\ \mathit{second}} = \frac{\#\ \mathit{Bits}}{1\ \mathit{second}}$$

- Using Sparling et. al. Rating: how difficult is it? and Cosley et. al. Seeing is Believing we can estimate.

$$\begin{aligned} \text{2-point} &= 0.1082 \\ \text{10-point} &= 0.1878 \end{aligned}$$

Words to know

Preference bits A measure of information about user preferences.

Bits per rating Measures how much preference information is contained in ratings.

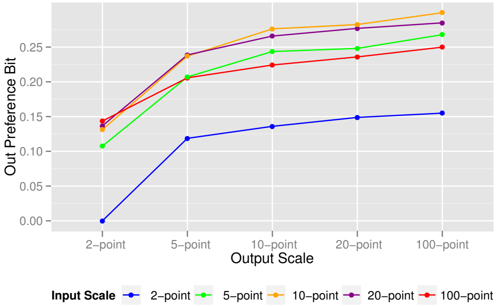
Bits per second Measures the *efficiency* of an interface at capturing preference bits.

Bits per prediction Measures how much preference information is contained in predictions.

- Predictions ($P_{u,i}$) leave the recommender.
- Predictions predict user preferences ($\pi_{u,i}$)
- The amount of preference information leaving the recommender with $I(\pi; P)$
- We measure this as $I(R; P) \leq I(\pi; P)$
- Yes, its just a fancy accuracy measure,
- But it handles varying scales well

Suggested use: Choosing how many stars to use.

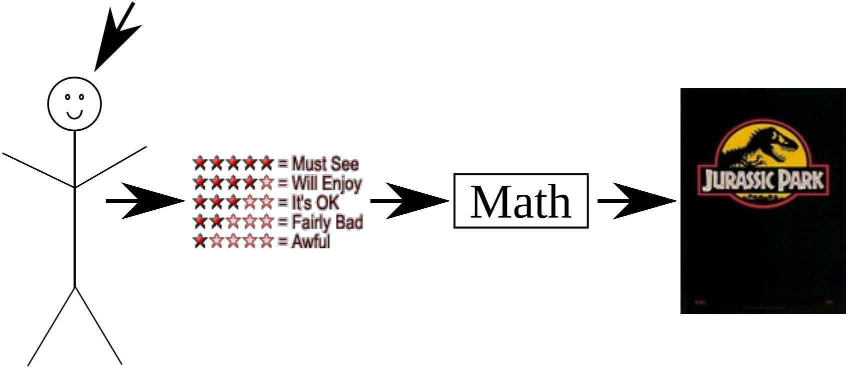
From Jester dataset (Goldberg et. al.)



- More prediction options, more bits
- Information hits a limit (again)
  - input scale controls limit
  - most bits at 10 point scale

Preference Bits

Measure with: Mutual information

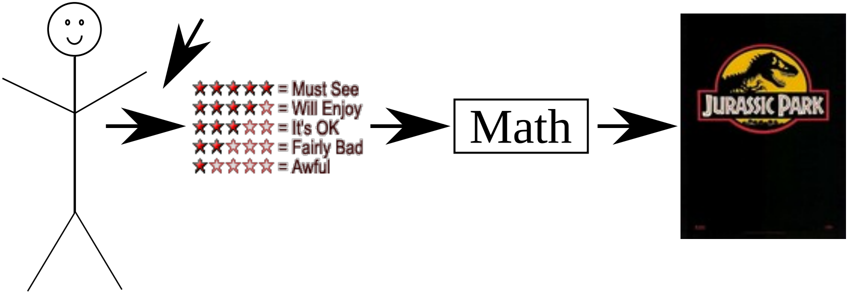


A measure of information about user preferences.

Bits Per Rating

Mutual information between ratings and preferences I(π;R)

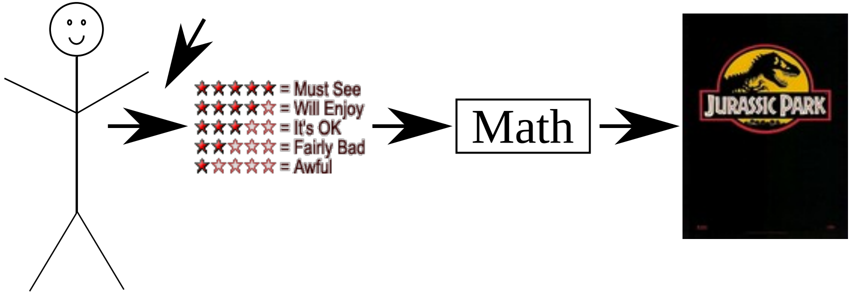
Measure with: $I(R_1;R_2)$



Measures how much preference information is contained in ratings.

Bits Per Second

Measure with: Bits per rating times Ratings per second

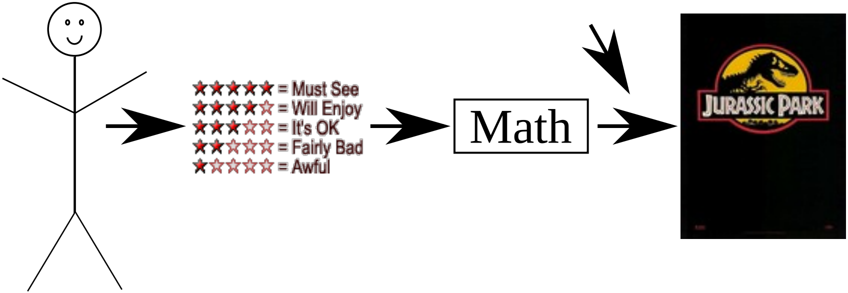


Measures the efficiency of an interface at capturing preference bits.

Bits Per Prediction

Mutual information between prediction and preferences I(π;P)

Measure with: I(R;P)



Measures how much preference information is contained in Predictions.

The Next Steps

- How many stars should we use?
- What information helps users the most?
- What are the difference the preference bits of different domains?
- Does preference bits hold any relationship with user satisfaction?

Thank you