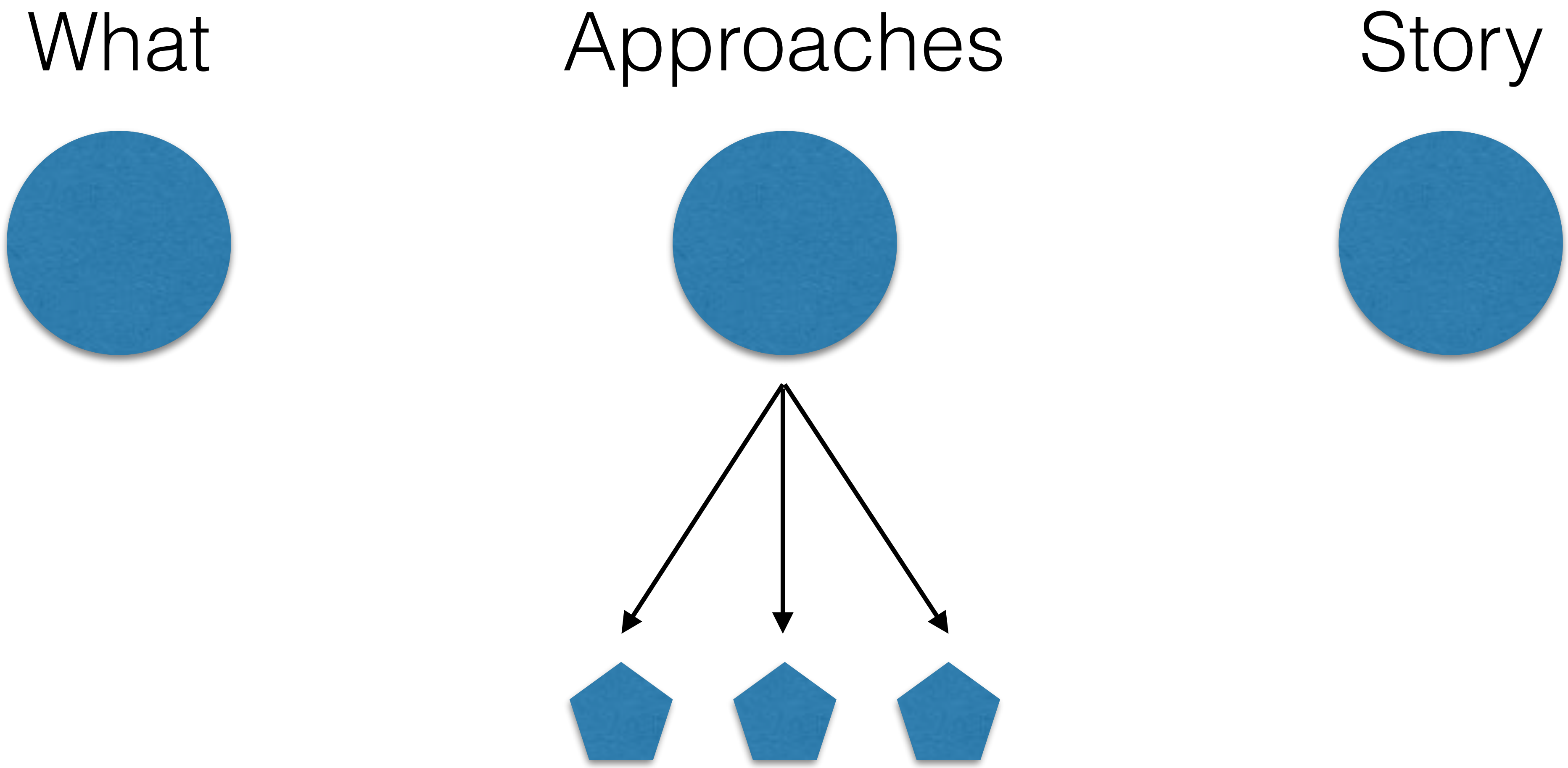
An Otherwise Blank Slide

Recommender Systems

Overview

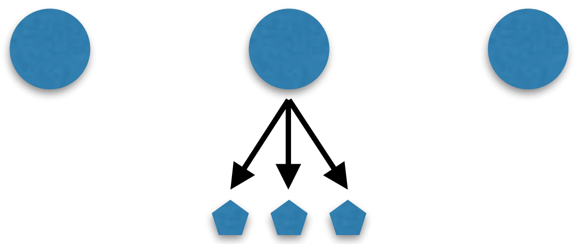
- What are Recommender Systems?
- Three common approaches
- A product development story

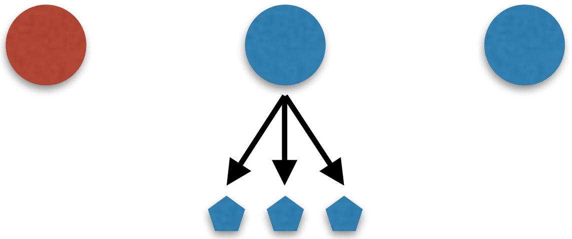


What

Approaches

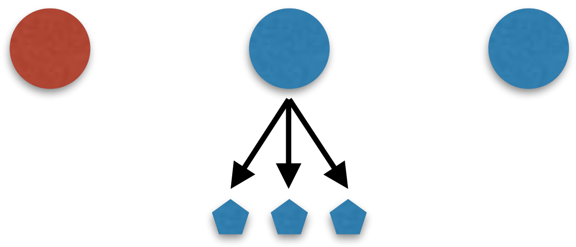
Story





What are Recommender Systems?

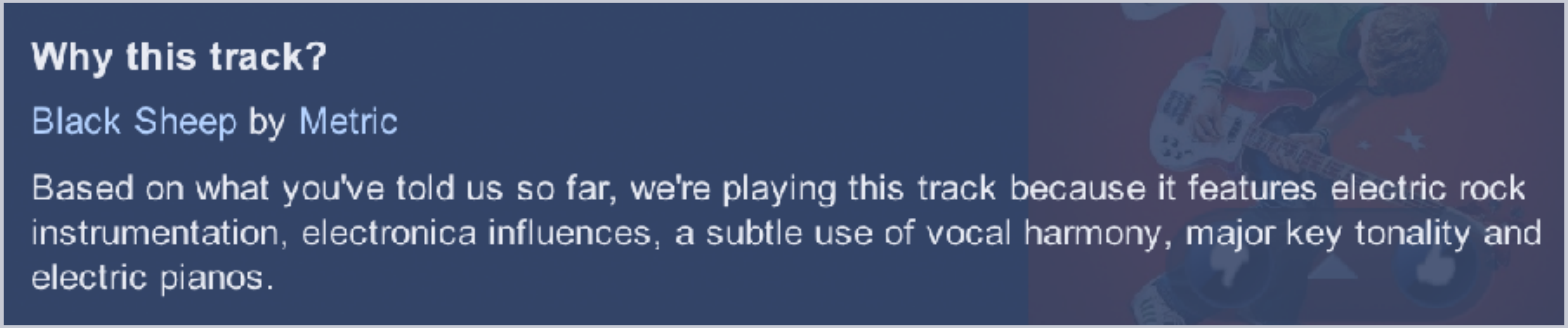
What are Recommender Systems?



- Goal: Offer a user something they might be interested in
  - What is “interest”?
    - Binary yes/no
      - Here are songs you’ll like
    - Ranked list
    - Collection of items
      - You might want a memory card with that camera
  - What can we recommend?
    - another person, place, thing, experience, company, employee
    - dating, travel, shopping, media, HR

What are Recommender Systems?

- Goal: Offer something the user cares about



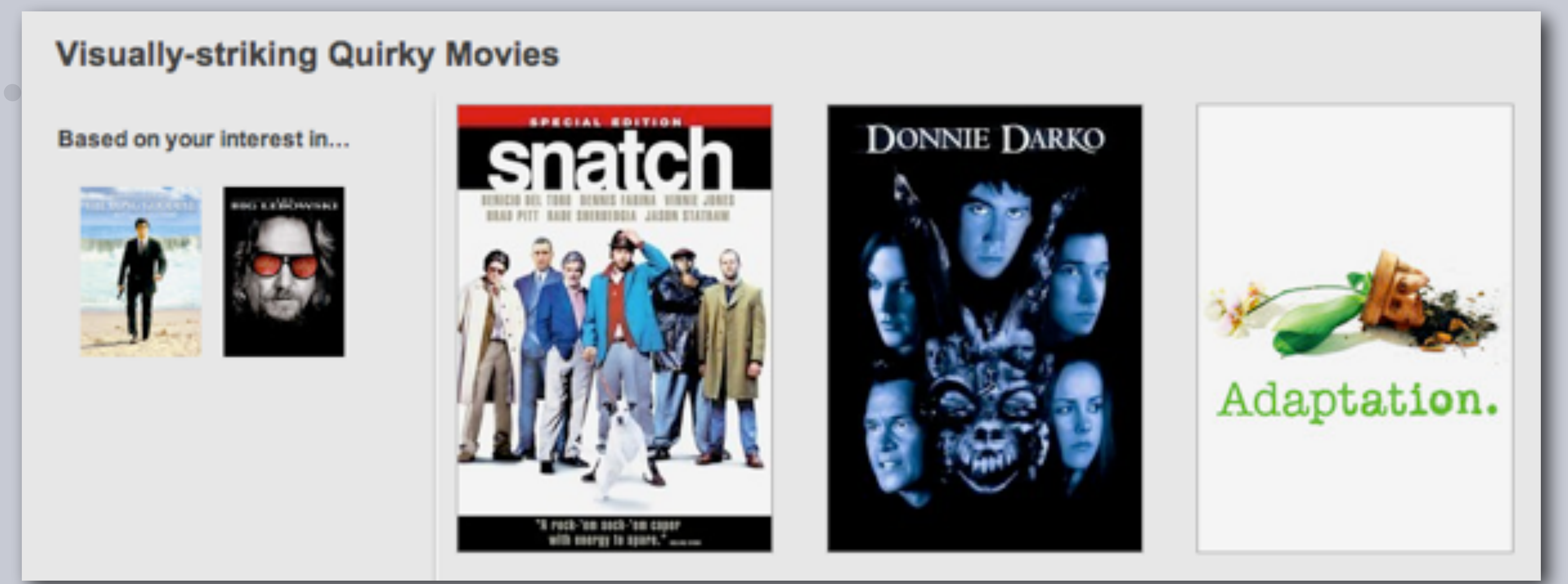
Why this track?

Black Sheep by Metric

Based on what you've told us so far, we're playing this track because it features electric rock instrumentation, electronica influences, a subtle use of vocal harmony, major key tonality and electric pianos.

- Field: Blend of AI, HCI, CogPsy
  - personalization, serendipity, motivation, trust, confidence

What are Recommender Systems?

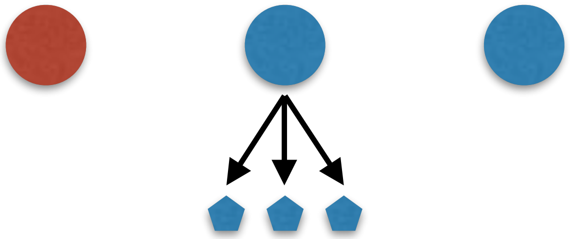


Visually-striking Quirky Movies

Based on your interest in...

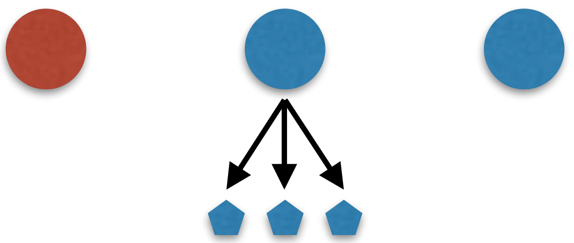
- Field: Blend of AI, HCI, CogPsy
  - personalization, serendipity, motivation, trust, confidence

What are Recommender Systems?



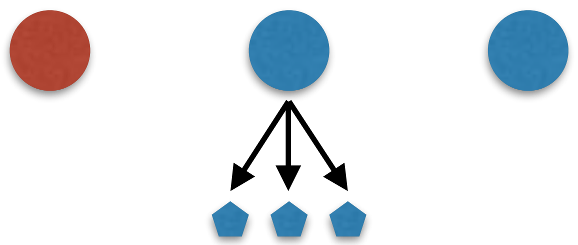
- Field: Blend of CS, AI, HCI, CogPsy
    - Serendipity
    - Personalization
    - Diversity
    - Persistence
    - Modality
    - Privacy
    - Motivation
    - Trust

Kinds of Data



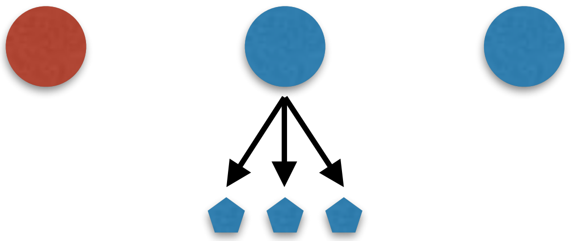
- Target person (recommendee/active user)
  - Are they middle-aged? What's their income? How often do they log in?
- Target item
  - Is it expensive? Does it feature Nicholas Cage?
- User-item interactions
  - Did many users buy/star/like/save/wishlist/rate this item? What was the typical rating?
  - Did this user buy/star/etc many items? What was their typical rating behavior?

Encoding Information



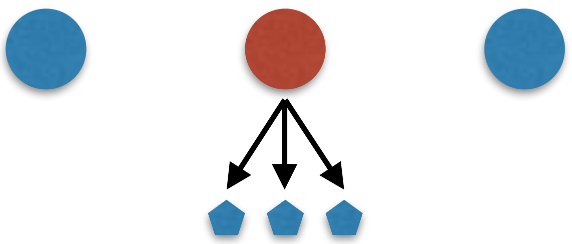
- What does a "like" mean? How about a "view"? How about 10 views? How about a "saved but kept not buying it"? (Did they think it was funny? Are they actually using 'like' as a 'bookmark' functionality?)
- Explicit v Implicit
    - Did the user take a clear and explicit action?
        - Thumbs up/down on Pandora
        - Did the user, UI/UX designer, and data scientist all interpret this action to be the same thing?
    - Did the user take some actions that might indicate something, but which weren't explicit?
        - Did they view a product? Did they share it? Did they spend a long time on the page? (Was their mouse active while they were on the page?)
- Incentive pollution
    - Did you ask your friend to review your recommender system/app?

User Utility

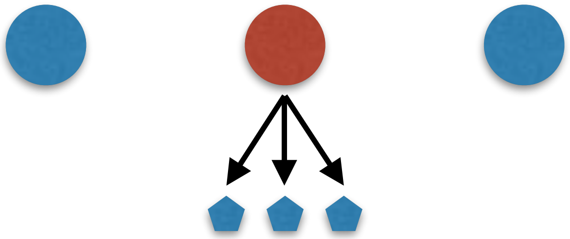


- How and when do the recommendations become useful?
  - As a function of data, some algorithms…
    - rely on a single user rating many items before becoming useful
    - rely on many users rating many items, collaboratively
    - require lots of info about the users themselves
    - have a higher or lower ceiling of learning
  - As a function of time, some algorithms…
    - are easy to update/recalculate, others are very hard
- The approach used when your company/data is small isn't necessarily what you'll use later

Three Common Approaches

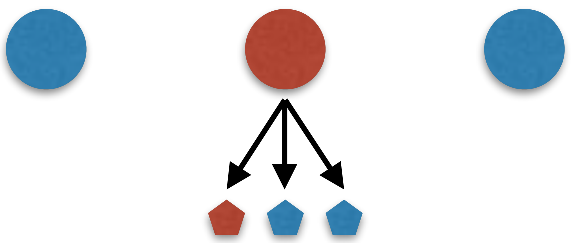


- Many ways to frame the challenge
- Different data requirements, runtime requirements, types of success
- Lots of moving parts; ‘core’ algorithms at the heart of various approaches



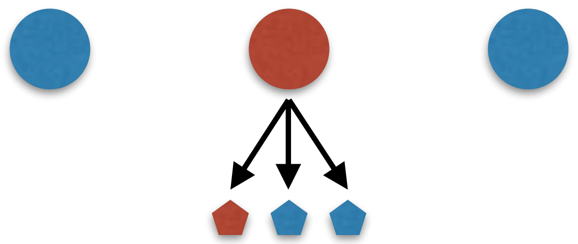
Classification/Regression

Classification/Regression



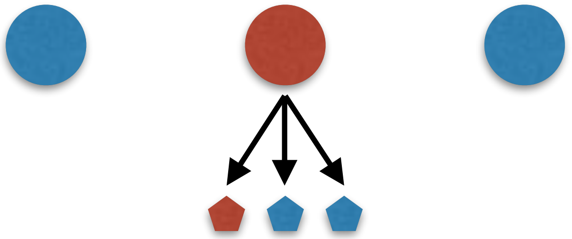
- Classifier to predict interaction/rating, using past interactions/ratings as training data
  - Using features of the people I’ve liked so far on <dating site> as training data, predict/rank who I’ll like
  - Using features of the people who have bought product X, predict/rank who else is likely to buy it

Classification/Regression



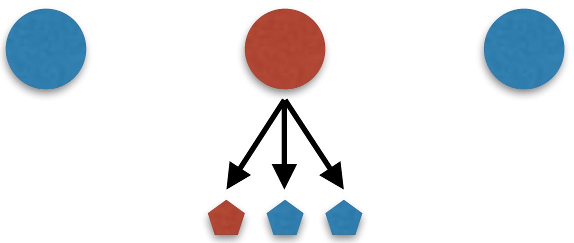
6’2	Blond	5 o’clock shadow	Y
5’11	Brown	Clean-shaven	N
6’01	Black	Gandalf	N
6’0	Blond	Moustache	?

Classification/Regression

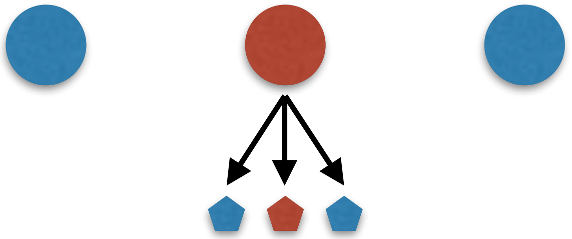


- Advantages
    - Familiar tools/algorithms
    - Well studied success metrics (for one classifier)
        - Diagnose bias/variance, swap algorithms, add features
    - Well studied confidence metrics
    - Potential for very personalized results, learning overall trends
    - Potential for good explanations (confidence&trust)
    - Single-user case is OK
    - Information privacy maintained

Classification/Regression

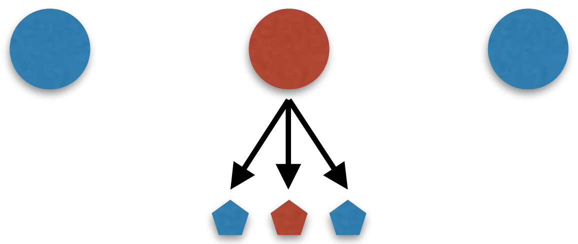


- Limitations
  - Training data required: interactions/ratings (Y)
  - Feature engineering required: meaningful descriptive data (X)
  - Tuning one classifier is tricky; tuning n classifiers?
    - Devops challenge (training, caching)
    - Data Science challenge (success metric for experiments)
  - Time to first utility (from User’s POV): interactions, compute time
  - Learning overall trends may be an oversimplification, compressing modalities
    - Complex decision boundaries can be learned w enough data, still a UX issue
  - Rating/Interaction gathering&interpretation creates product restrictions
  - Solipsistic (not inherently social)



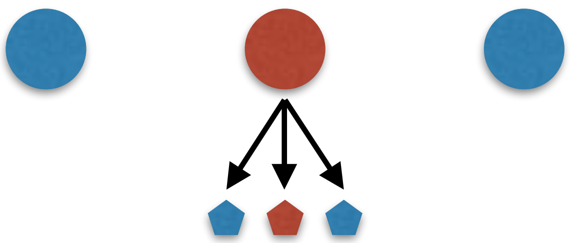
Collaborative Filtering

Collaborative Filtering



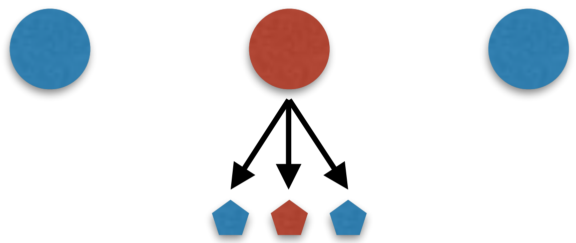
- Use similarity to other users/products to predict interaction/rating, using past interactions/ratings as training data
    - To the degree to which I'm similar to Bob, I may feel like he did about this item. Likewise for Carly, Danielle, Edgar…
    - I'm similar to Bob to the degree to which we've agreed on other items

Collaborative Filtering



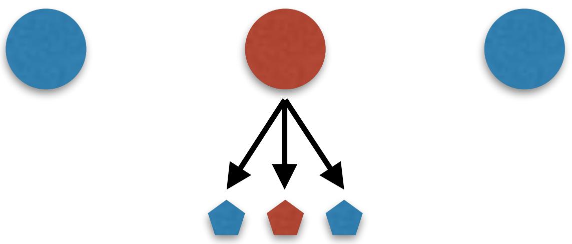
Bob	0	1	1
Carly	1	0	0
Danielle	1	1	1
Dan	0	1	?

Collaborative Filtering



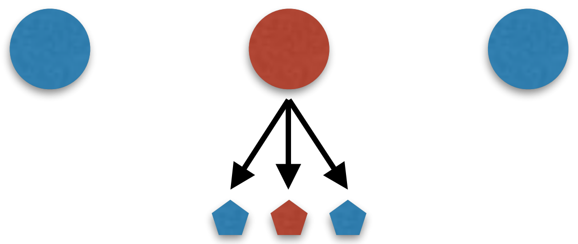
$$prcd(p_i, t_i) = \bar{r}_{p_i} + \frac{\sum_{p_j \in P, i \neq j} sim(p_i, p_j) \times (r_{p_j, t_i} - \bar{r}_{p_j})}{\sum_{p_j \in P, i \neq j} sim(p_i, p_j)}$$

Collaborative Filtering



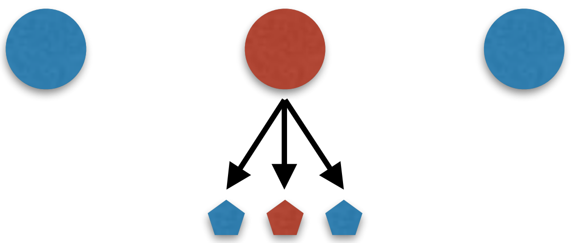
- Variations
  - Neighborhoods
    - Similarity amongst n most similar people/items
    - Faster (can also cache similarity)
  - Flip the matrix: items “choose” people rather than vice versa
  - Use a different similarity function: “content-boosted” collaborative filtering
    - Dan and Bob are similar because they have similar Facebook profiles, rather than because they rated things similarly

Collaborative Filtering

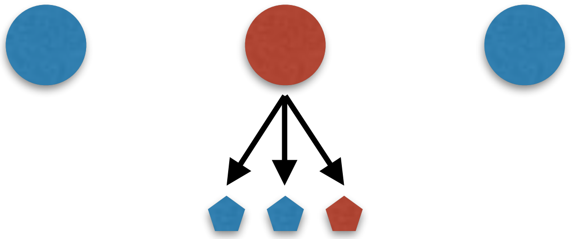


- Advantages
  - Content agnostic
    - Feature engineering not required
  - Inherently social
    - Potential for social explanations
    - Popularity naturally built-in, in a way
  - One rating helps many people (“collaborative”)
    - Time to first utility (interactions/ratings)

Collaborative Filtering

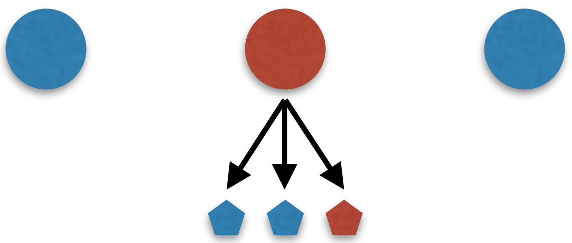


- Limitations
  - Content agnostic
    - May not take advantage of all information known about people/items
  - Inherently social
    - Potentially violates user privacy expectations
    - Less-rated areas can lead to lower-quality recommendations
  - Sparsity … it’s really tough
    - Low ratings or overlaps (similarity) can lead to relatively little information going into each prediction
    - High person/item churn can lead to low overlaps
    - Single-user case is a fail case



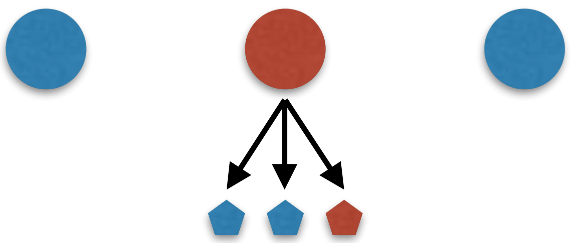
Similarity

Similarity



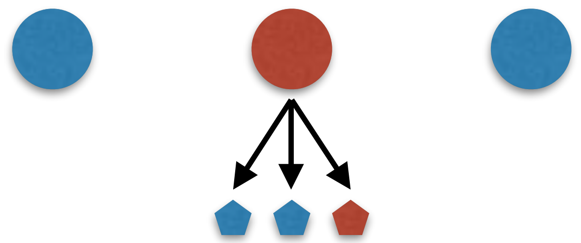
- Go all-in on similarity: do a really good job of figuring out which things are similar to one another
  - Use that similarity to make connections: because I liked A and A is similar to B (in reduced dimensionality subspace), I may like B
  - “Smooth out” similarity space, and make distance more meaningful (anti-Curse of Dimensionality): Dimensionality Reduction
    - Loss of information, nuances may be lost
    - More feature ‘overlaps’, more ways to compare
    - Depending on alg, can be very explainable (eg LDA topic modeling is relatively easy to interpret)
  - Runtime: run a potentially-computationally-expensive offline dimensionality reduction (eg, LDA/topic modeling/NMF), then index similarity in reduced-dimensionality subspace for fast queries

Similarity



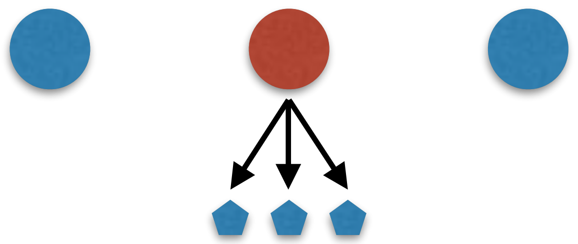
- Advantages
  - User information requirements are very low
    - Interactions/ratings not needed to precompute similarities
    - Time to first utility is 1 rating
  - Potential for high interpretability of similarities and recommendations
  - Modality preserved*
  - Runtime requirements are low
    - Heavy computation can be done offline
  - Single-user use case OK
  - Potentially useful side-effects: similarities, topics, topic distributions

Similarity



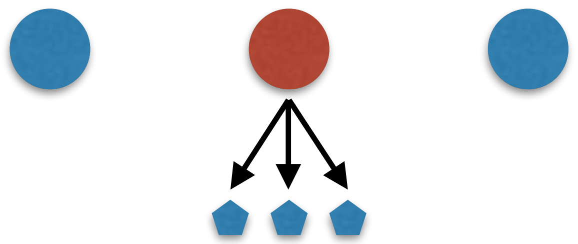
- Limitations
    - Item information requirements fairly high
        - Feature engineering required: meaningful descriptive data about items
    - Information about person not taken into account
    - High compute requirements for dimensionality reduction, indexing
    - Can be difficult to evaluate quality of dimensionality reduction
    - Solipsistic (not inherently social)
    - Overall trends ignored; pointwise recommendations only

Three Common Approaches



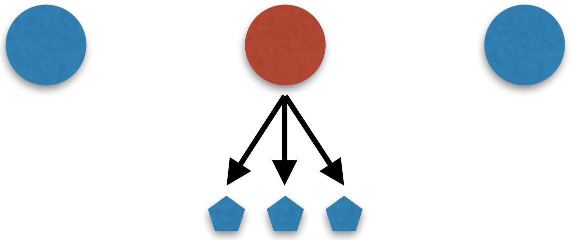
- Classification/Regression
- Collaborative Filtering
- Similarity (Dimensionality Reduction)

Three Common Approaches

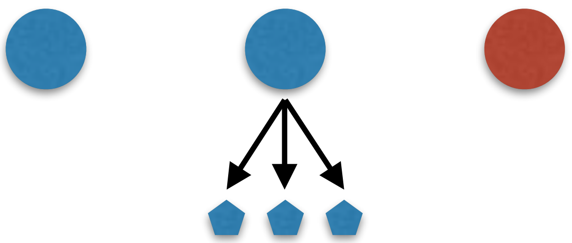


- Recommendation can be framed in many ways
  - These approaches can be stacked & combined & inverted, etc.
    - Reduced dimensions as input to classification?
    - Collaborative Filtering matrix blanks filled in with regression output?
    - Similarity function based on dimensionality reduction?
    - Inversion of person<>item framing?

Three Common Approaches



- Additional considerations
  - User state (already owns X, needs Y?)
    - basket analysis
  - User preference shift (used to like X; decay?)
  - Overall success evaluation (A/B test bad recs?)



A Story

Thanks!