Caltech

# Machine Learning & Data Mining

## CS/CNS/EE 155

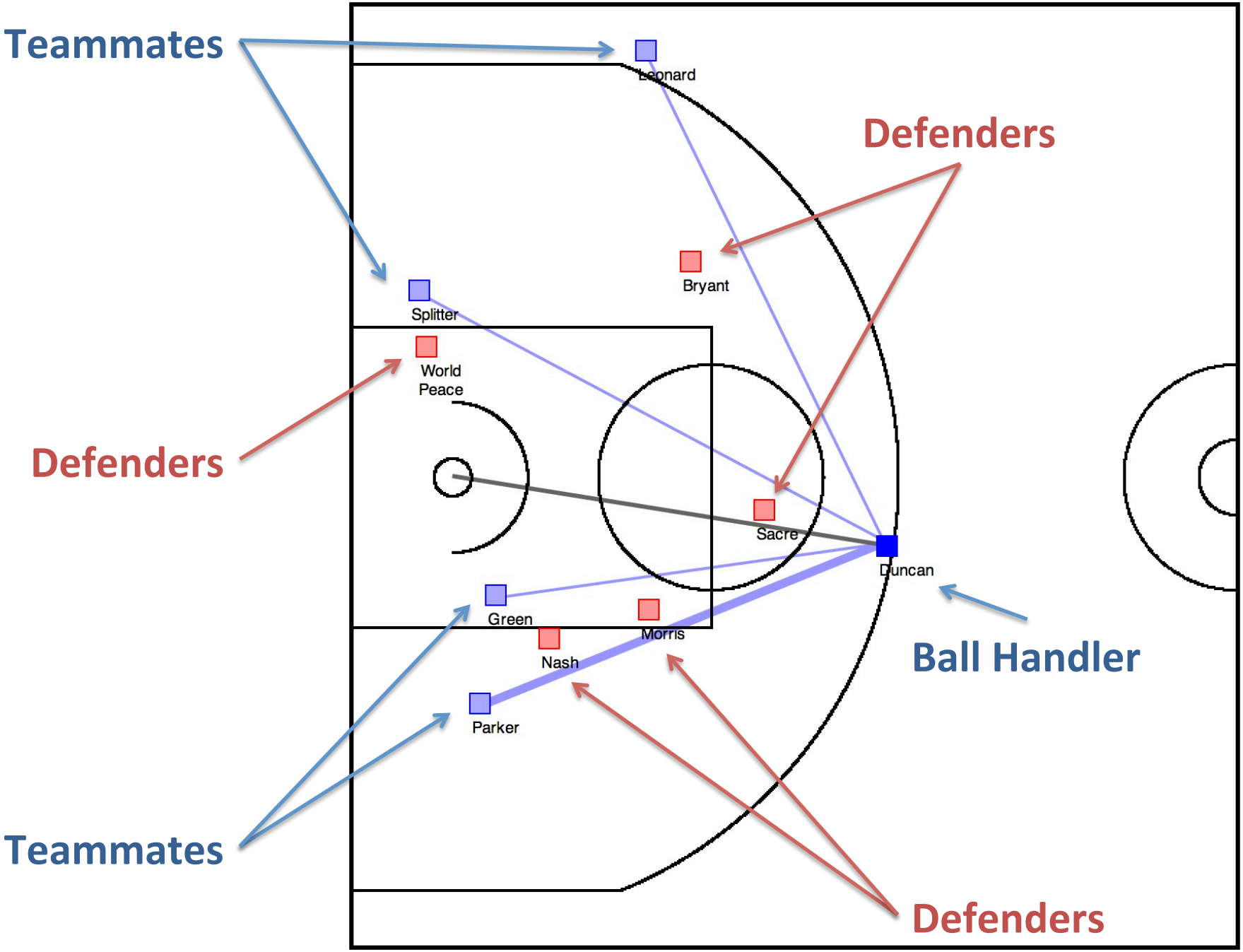
Lecture 15:

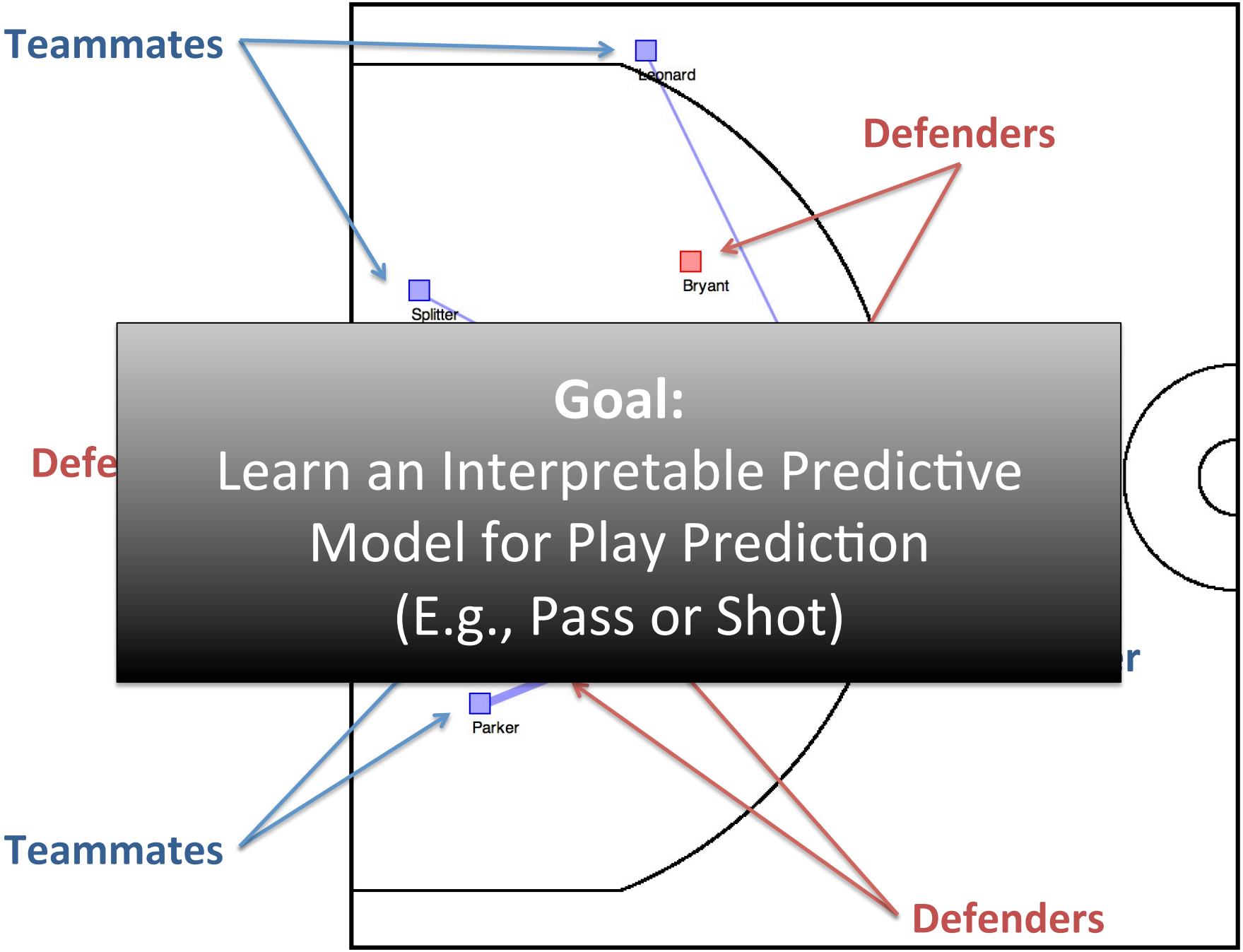
Recent Applications of

Latent Factor Models

# Today

- Miniproject 1 Reports Due
- Latent Spatial Models for Basketball Play Prediction
- Latent Tensor Models for Collaborative Clustering
- aka: Stuff Yisong Likes


Teammates
Leonard
Defenders
Bryant
Splitter
World Peace
Defenders
Sacre
Duncan
Green
Morris
Nash
Ball Handler
Parker
Teammates
Defenders

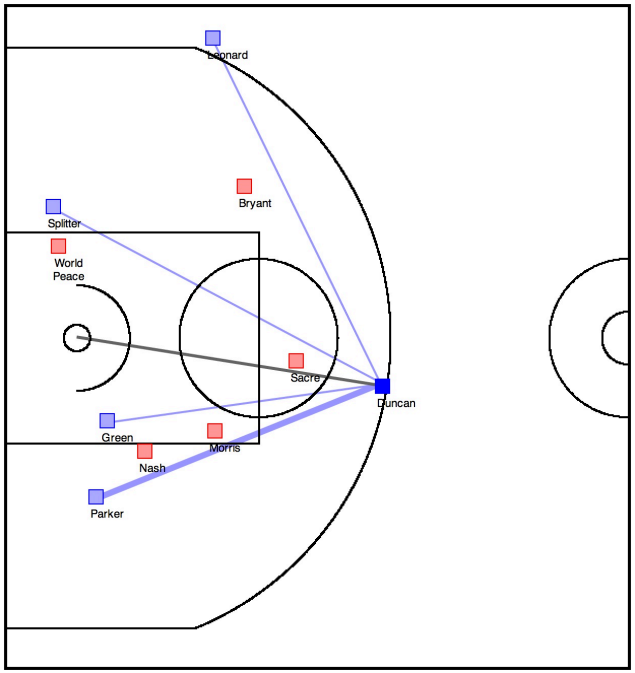

Teammates
Leonard
Defenders
Bryant
Splitter
Goal:
Learn an Interpretable Predictive Model for Play Prediction
(E.g., Pass or Shot)
Parker
Teammates
Defenders

# Demo

http://projects.yisongyue.com/bballpredict/

# Prediction

- Game state: **x**
  - Coordinates of all players
  - Who is the ball handler
- Event: y
  - Ball handler will shoot
  - Ball handler will pass (to whom?)
  - Ball handler will hold onto the ball
  - 6 possibilities
- **Goal:** Learn P(y|**x**)
  - Interpretable




http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# Logistic Regression
## (Simple Version: Just for Shooting)

$$P(y \mid \mathbf{x}) = \frac{\exp\{F(y \mid \mathbf{x})\}}{Z(\mathbf{x} \mid F)} \qquad Z(\mathbf{x} \mid F) = \sum_{y' \in \{s, \perp\}} \exp\{F(y' \mid \mathbf{x})\}$$

$$F(y' \mid \mathbf{x}) = \begin{cases} F_s(\mathbf{x}) & y' = s \quad \text{Shot} \\ F_\perp & y' = \perp \quad \text{Hold on to ball} \end{cases}$$

Offset or bias

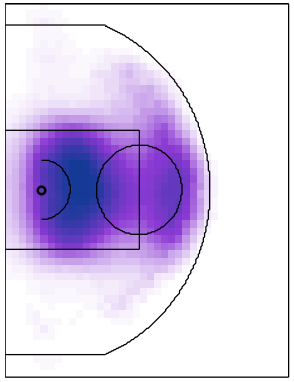
$$P(y = s \mid \mathbf{x}) = \frac{1}{1 + \exp\{-F_s(\boldsymbol{x}) + F_\perp\}}$$

# Example


Tim Duncan


$$P(y = s \mid \mathbf{x}) = \frac{1}{1 + \exp\left\{-F_s(\boldsymbol{x}) + F_\perp\right\}}$$

$F_s(\mathbf{x})$



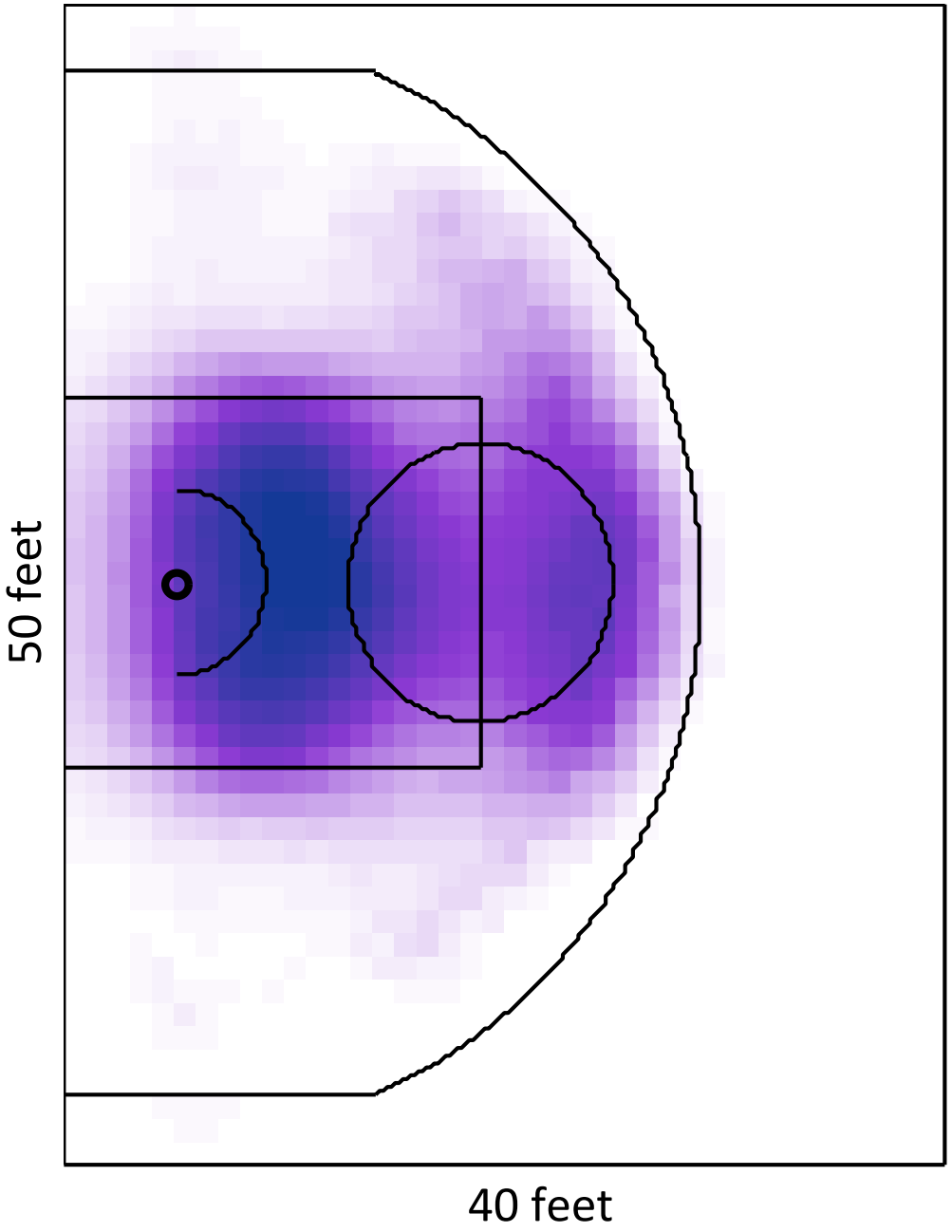
# Fine-Grained Spatial Models

- Discretize court
  - 1x1 foot cells
  - 2000 cells

$$F_s(\mathbf{x}) = F_s^T \phi_s(\boldsymbol{x})$$

2000 dim coeff vector

Indicator feature vector
(1 in cell that ball handler is)
(0 in all other entries)

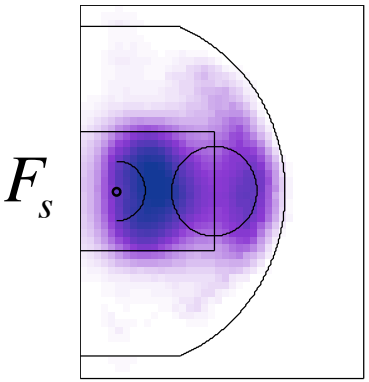


# Story so Far: Spatial Logistic Regression

$$P(y = s \mid \mathbf{x}) = \frac{1}{1 + \exp\left\{-F_s(\boldsymbol{x}) + F_\perp\right\}}$$

$$= \frac{1}{1 + \exp\left\{-\left(F_s^T \phi_s(\boldsymbol{x}) - F_\perp\right)\right\}}$$

- Probability of shooting log-linear
  - In spatial cell feature representation



# Training Data

**STATS SportsVU**

2012/2013 Season, 630 Games,

80K Possessions, 380 frames per possession

# Learning the Model

- Given training data:

$$S = \{(\boldsymbol{x}, y)\}$$

Player Configuration

What Happened Next

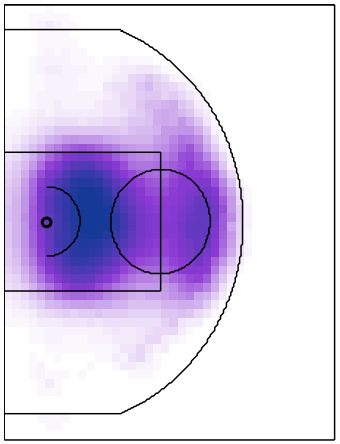
- Learn parameters of model:

$$\underset{F_s, F_\perp}{\operatorname{argmin}} \lambda \|F_s\|^2 + \sum_{(\boldsymbol{x},y)\in S} \ell\left(y, F_s^T \phi_s(\boldsymbol{x}) - F_\perp\right)$$

Log Loss

# Spatial Regularization

- Self-defined feature vector
  - Has spatial structure
- Expect $F_s$ to vary smoothly
  - But not enough training data
  - **Spatial Regularization:**



$$\underset{F_s, F_\perp}{\operatorname{argmin}} \lambda_1 \left\| F_s \right\|^2 + \lambda_2 \sum_{i,j \in N(i)} \kappa_{ij} \left( F_{s,i} - F_{s,j} \right)^2 + \sum_{(\boldsymbol{x}, y) \in S} \ell\left( y, F_s^T \phi_s(\boldsymbol{x}) - F_\perp \right)$$

$$\kappa_{ij} = \exp\left\{ -d(i,j)^2 / \sigma \right\}$$

# Combining Several Ideas

- Multiclass prediction
  - Predict different outcomes (e.g., shot, pass, etc)
- Spatial Model
  - Captures spatial structure
- Multitask prediction
  - One prediction task per player
- Interpretable Model
  - Low-dimensional representation
  - Easy to visualize

We've only seen shots so far

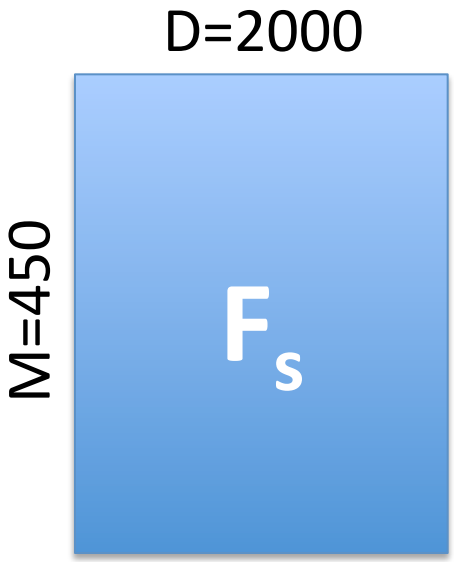
Discretize into fine-grained cells
(regularize neighbors to be similar)

Talk about this next

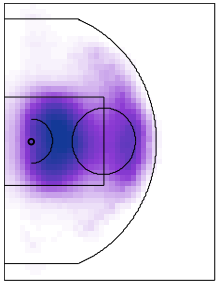
Latent factor part
Related to multitask prediction

# Multitask Prediction

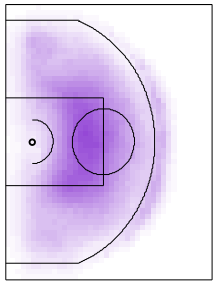
- One task per player
  - $F_s$ is actually a matrix of coefficients
  - Each row is a player
- Input data **x** contains id of ball handler
  - One task per ball handler




Tim Duncan



Dirk Nowitski

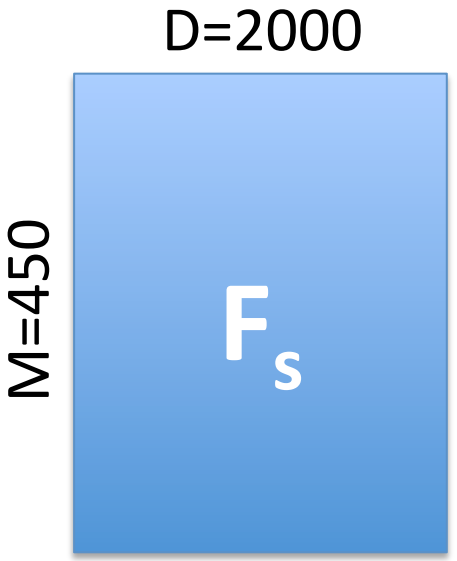


# Multitask Prediction

$$P(y = s \mid \mathbf{x}) = \frac{1}{1 + \exp\left\{-F_s(\boldsymbol{x}) + F_\perp\right\}}$$

$$= \frac{1}{1 + \exp\left\{-\left(F_{s,b(\boldsymbol{x}),l(\boldsymbol{x})} - F_\perp\right)\right\}}$$

Ballhander ID  Location




**Learning Objective:**

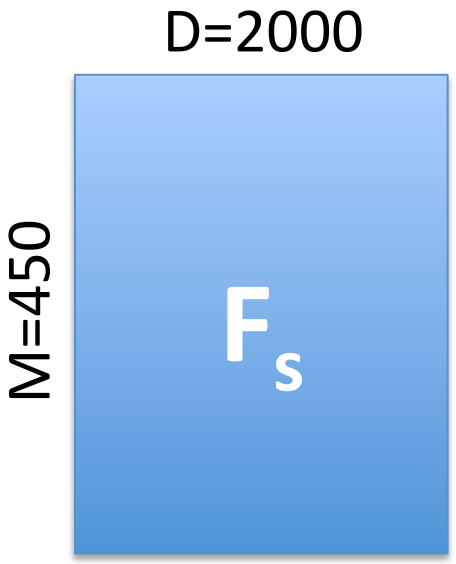
$$\underset{F_s, F_\perp}{\operatorname{argmin}} \; \lambda_1 \left\| F_s \right\|^2 + \lambda_2 \sum_b \sum_{i,j \in N(i)} \kappa_{ij} \left( F_{s,b,i} - F_{s,b,j} \right)^2$$

$$+ \sum_{(\boldsymbol{x},y) \in S} \ell\left(y, F_{s,b(\boldsymbol{x}),l(\boldsymbol{x})} - F_\perp\right)$$

$$\kappa_{ij} = \exp\left\{-d(i,j)^2 / \sigma\right\}$$

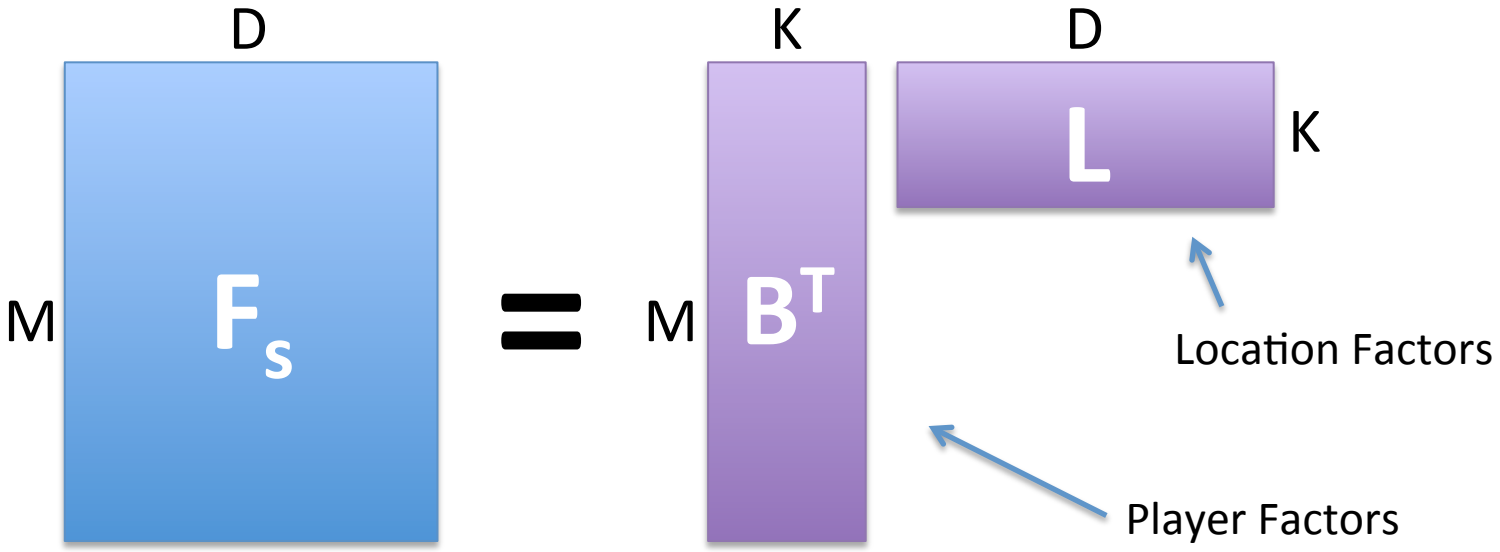
# Not Enough Training Data!

- Most players don't shoot that often
  - 2000 weights per player
- Cell discretization is very fine-grained
  - Spatial regularization helps some
- How to share data across players?
  - **Latent Factor Model!**



# Latent Factor Model (Shooting)

Assume that players can be represented by K-dimensional vector
Learn a common K-dimensional latent feature representation




- **Shooting Score:** $F_s(x) = B^T_{b(x)} L_{l(x)}$

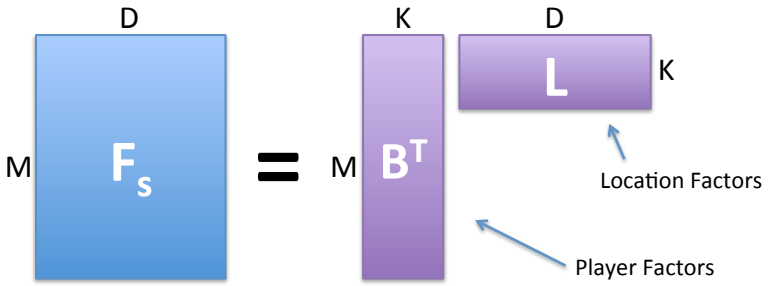
# New Prediction Model

$$P(y = s \mid \mathbf{x}) = \frac{1}{1 + \exp\{-F_s(\boldsymbol{x}) + F_\perp\}}$$

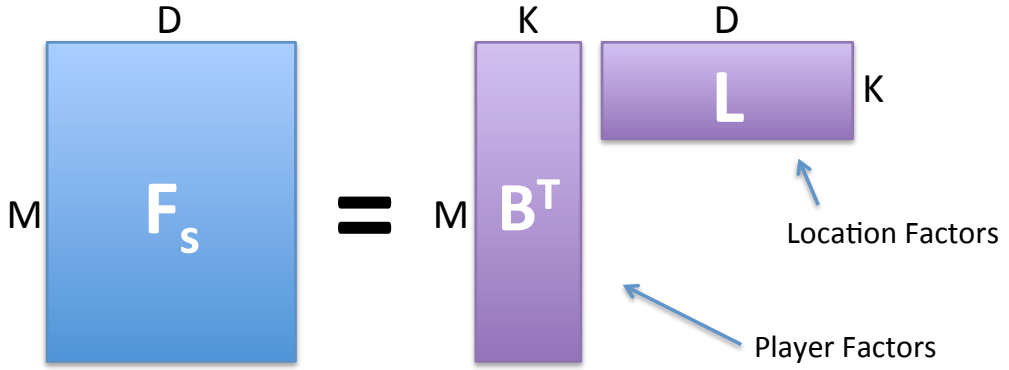
$$= \frac{1}{1 + \exp\left\{-\left(B^T_{b(\boldsymbol{x})} L_{l(\boldsymbol{x})} - F_\perp\right)\right\}}$$

Ballhander ID

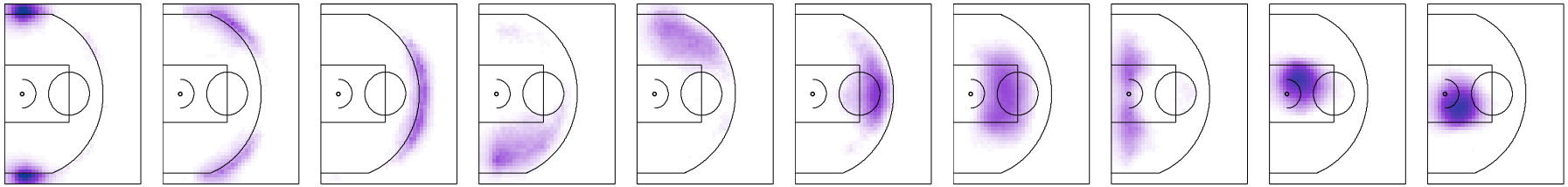
Location



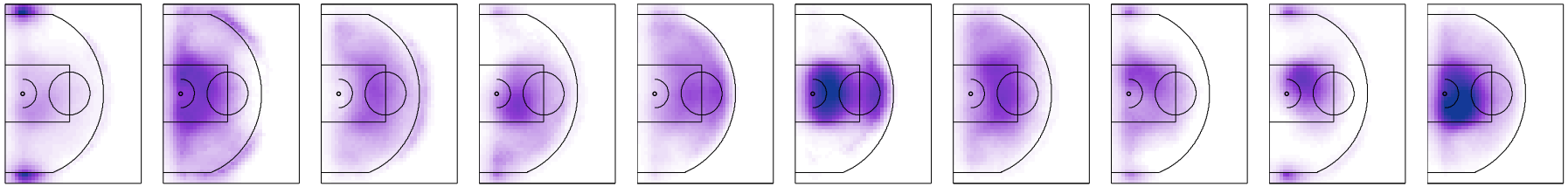

http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf




Enforce Non-Negativity
(Accuracy Worse)
(More Interpretable)



Visualizing location factors L (K=10)




Visualizing players

http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

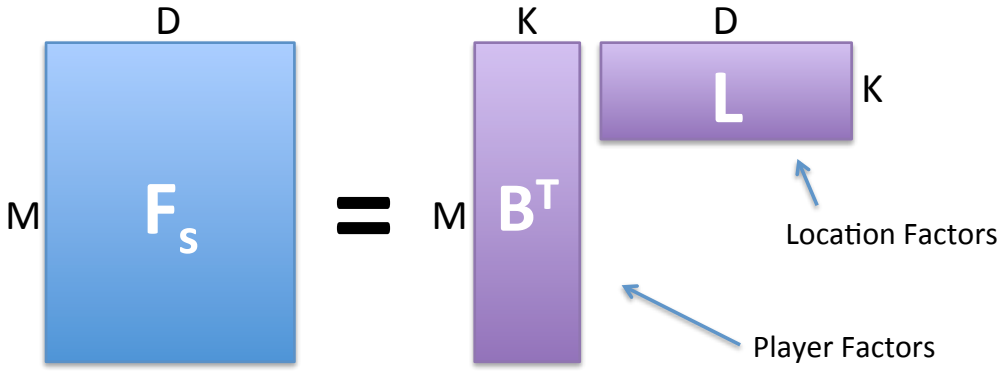
# Training Objective

Spatial Regularization

$$\underset{B\geq 0, L\geq 0, F_{\perp}}{\operatorname{argmin}} \lambda_1 \left( \|B\|^2 + \|L\|^2 \right) + \lambda_2 \sum_{k} \sum_{i,j \in N(i)} \kappa_{ij} \left( L_{k,i} - L_{k,j} \right)^2$$

$$+ \sum_{(\boldsymbol{x},y)\in S} \ell\left( y, B^T_{b(\boldsymbol{x})} L_{l(\boldsymbol{x})} - F_{\perp} \right)$$

Log-Loss of Latent Factor Score




$$S = \{(\boldsymbol{x}, y)\}$$

$$\kappa_{ij} = \exp\{-d(i,j)^2 / \sigma\}$$

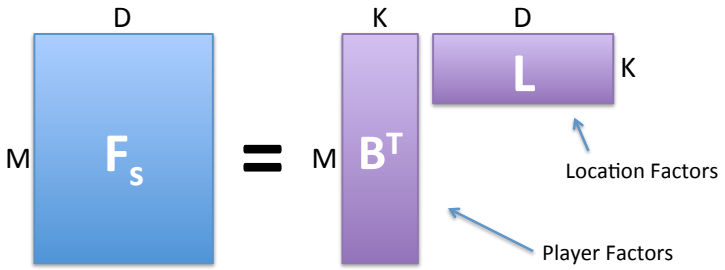
http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# Optimization via Gradient Descent

$$\underset{B\geq 0, L\geq 0, F_\perp}{\operatorname{argmin}} \; \lambda_1 \left( \|B\|^2 + \|L\|^2 \right) + \lambda_2 \sum_k \sum_{i,j \in N(i)} \kappa_{ij} \left( L_{k,i} - L_{k,j} \right)^2 + \sum_{(\boldsymbol{x},y)\in S} \ell\left( y, B^T_{b(\boldsymbol{x})} L_{l(\boldsymbol{x})} - F_\perp \right)$$

$$\partial_{L_i} = 2\left( \lambda_1 L_i + \lambda_2 \sum_k \sum_{j \in N(i)} \left( L_{k,j} - L_{k,i} \right) \right) - \sum_{(\boldsymbol{x},y)\in S} \frac{\partial \log P(y \mid \boldsymbol{x})}{\partial L_i}$$

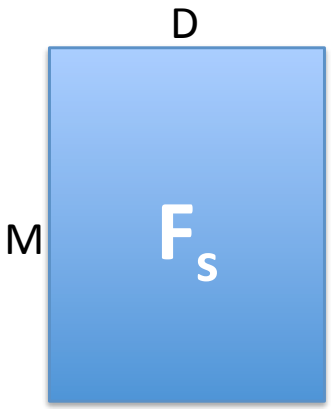
$$\frac{\partial \log P(y \mid \boldsymbol{x})}{\partial L_i} = \left( 1_{[y=s]} - P(s \mid \boldsymbol{x}) \right) B_{b(\boldsymbol{x})}$$




http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# Initialization

- Need to initialize B & L before doing gradient descent
  - Need to preserve spatial structure
  - Random initialization often doesn't work
- Train $F_s$ first w/o factorization
  - Then factorize F into B*L
    - Non-negative Matrix Factorization
  - Use as initialization




http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# General Prediction Problem
## (Multiclass Logistic Regression)

$$P(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x}|F)} \exp\{F(y|\mathbf{x})\}$$

Scoring Function

Multiclass Classification

$$Z(\mathbf{x}|F) = \sum_{y' \in Y(\mathbf{x})} \exp\{F(y'|\mathbf{x})\}$$

All possible y's

Partition Function

$$F(y|\mathbf{x}) = \begin{cases} F_s(\mathbf{x}) & \text{if } y = s \\ F_p(i, \mathbf{x}) & \text{if } y = p_i \\ F_\perp(\mathbf{x}) & \text{if } y = \perp \end{cases}$$

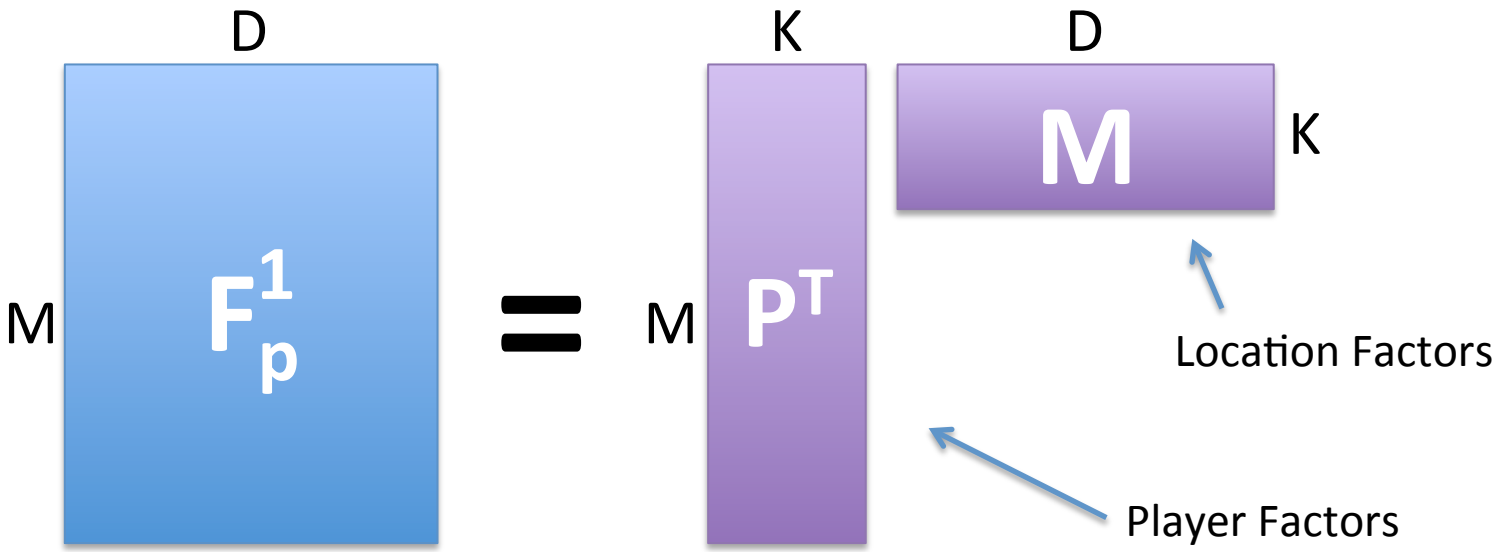
Shot

Pass to teammate

Hold on to ball

http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

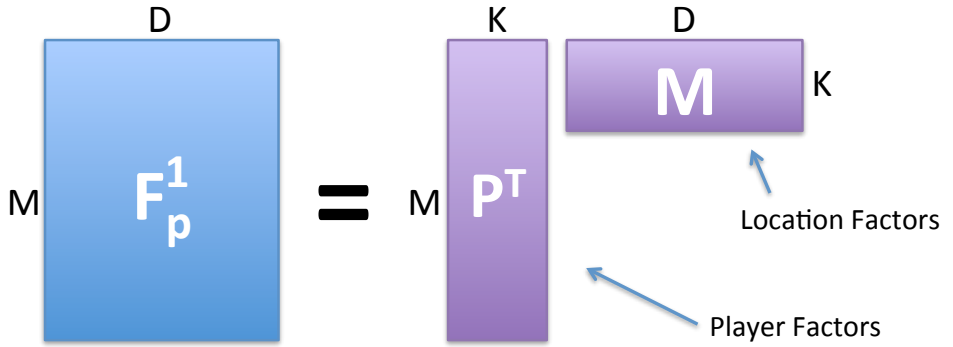
# Latent Factor Model (Passing)



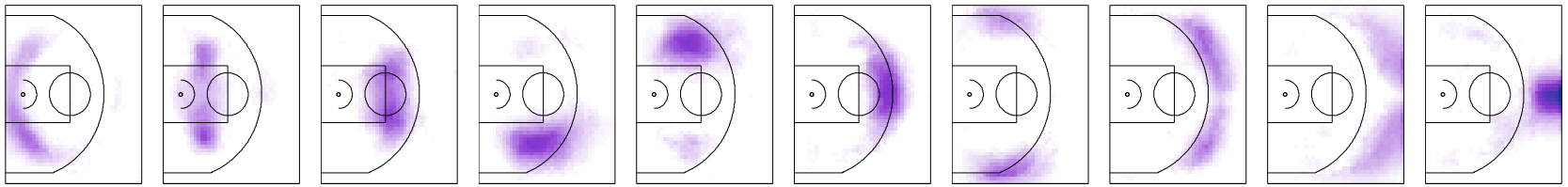

- **Passing Score:** $F_p(i, \boldsymbol{x}) = P_i^T L_{l(i,\boldsymbol{x})}$

http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

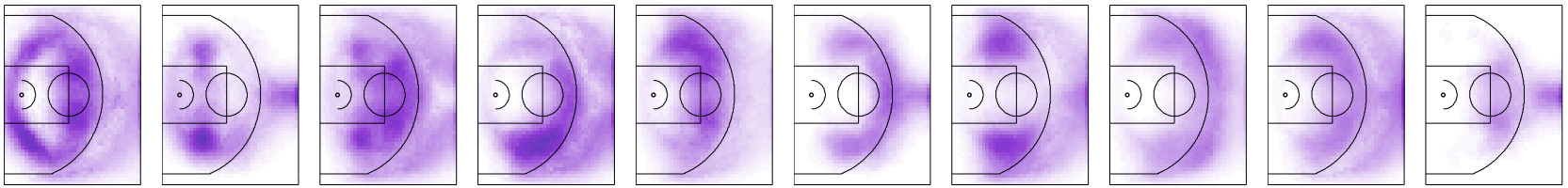
# Where are Players Likely to Receive Passes?




Enforce Non-Negativity
(Accuracy Worse)
(More Interpretable)

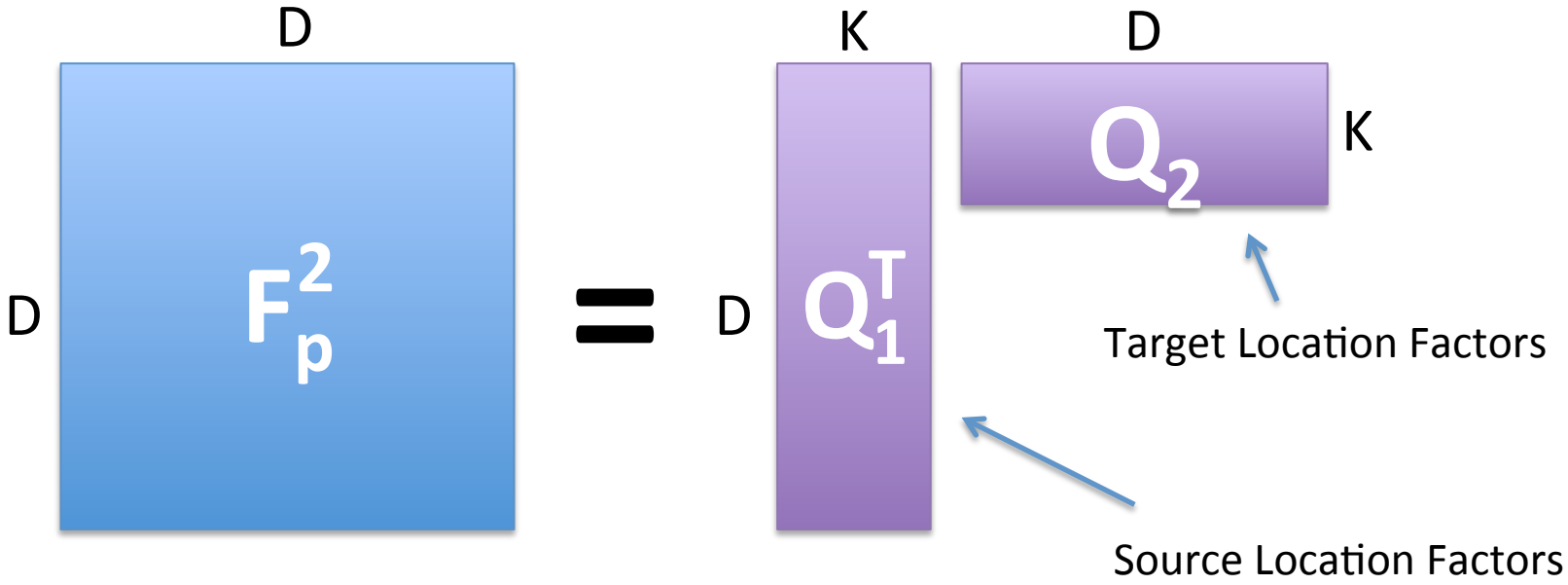


Visualizing Location Factors M




http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# Latent Factor Model (Passing) #2

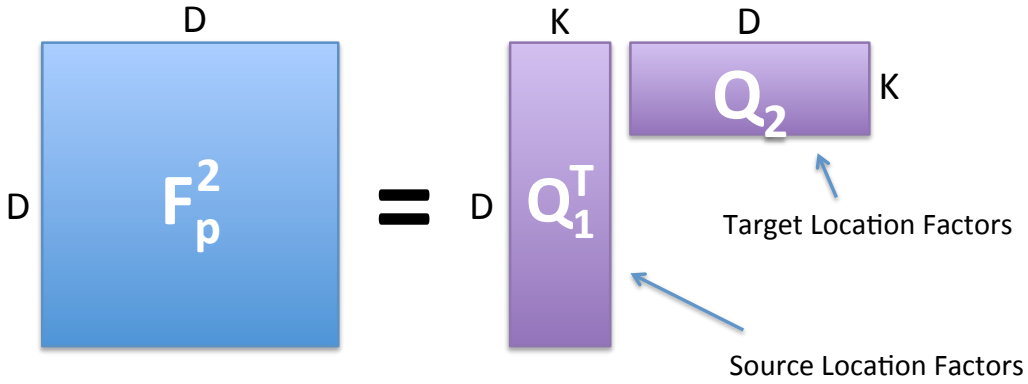


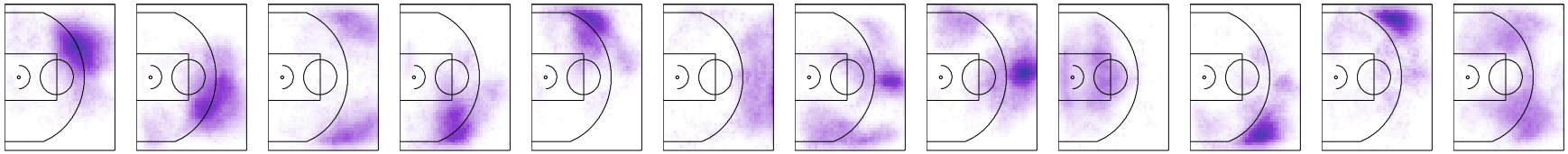

**Additive Decomposition:**

- **Passing Score:** $F_p(i,\boldsymbol{x}) = P_i^T L_{l(i,\boldsymbol{x})} + Q_{1,l(\boldsymbol{x})}^T Q_{2,l(i,\boldsymbol{x})}$

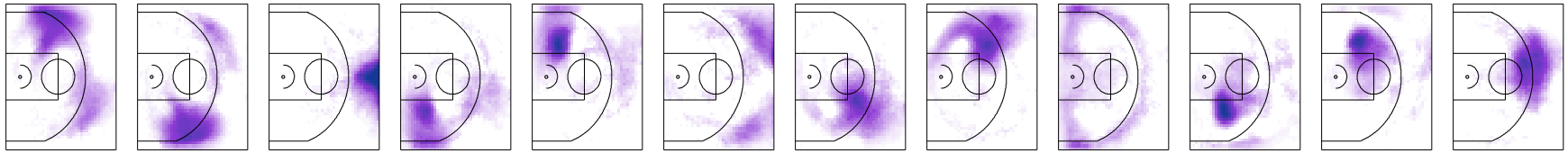
http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# How do passes tend to flow?



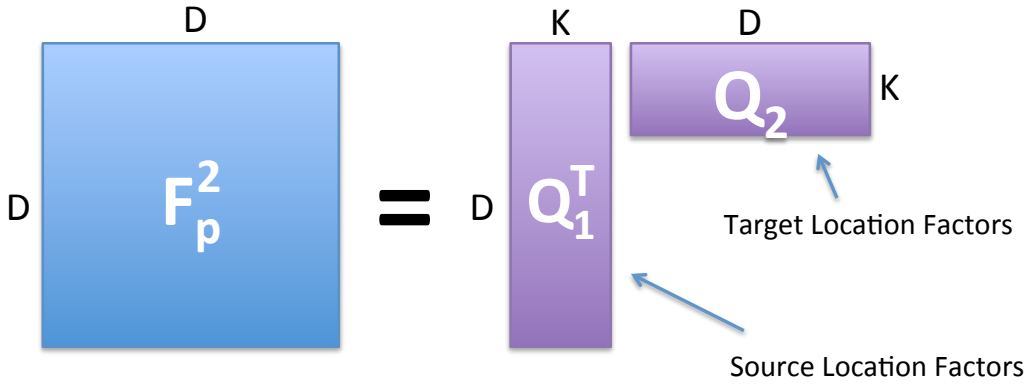
$Q_1$

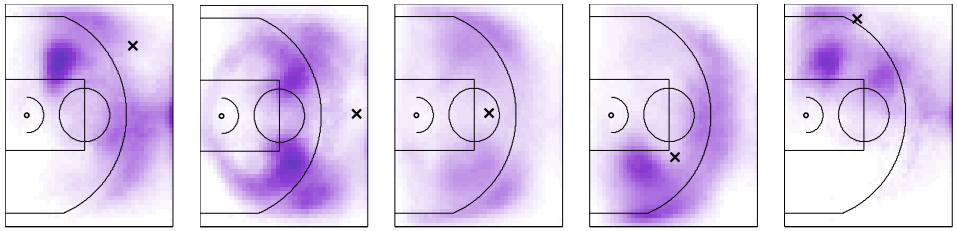


$Q_2$

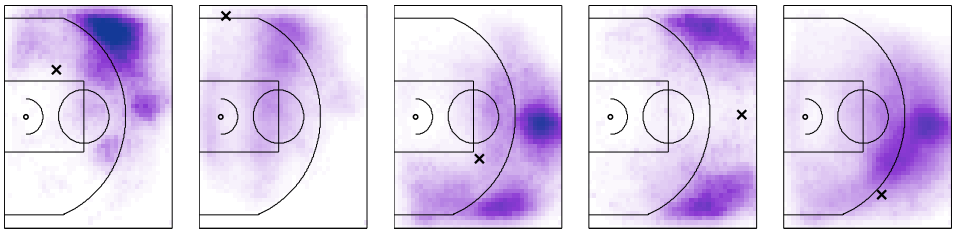
http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# How do passes tend to flow?



Passing From "X"

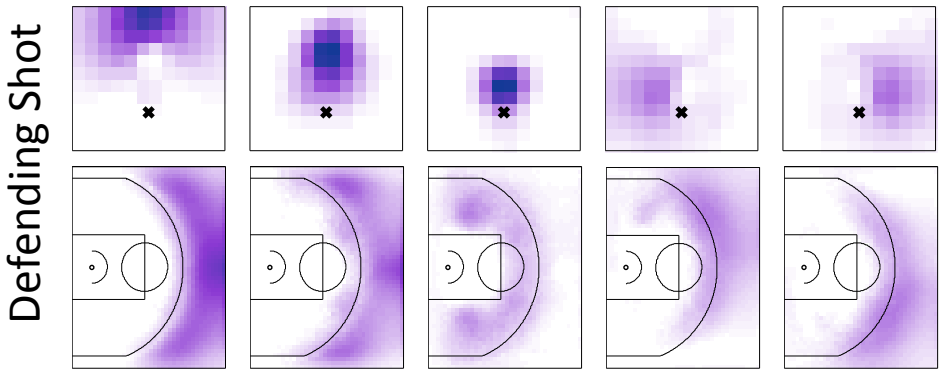


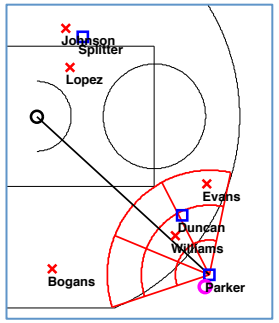
Passing To "X"

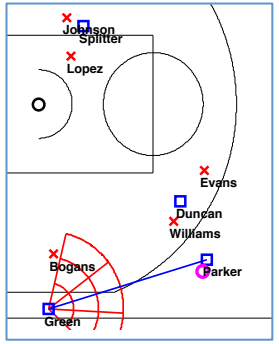
http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

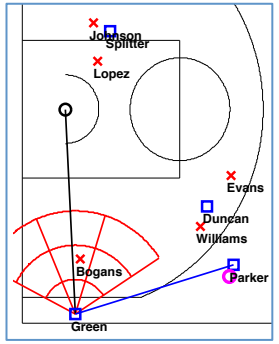
# Visualizing Defender Factors

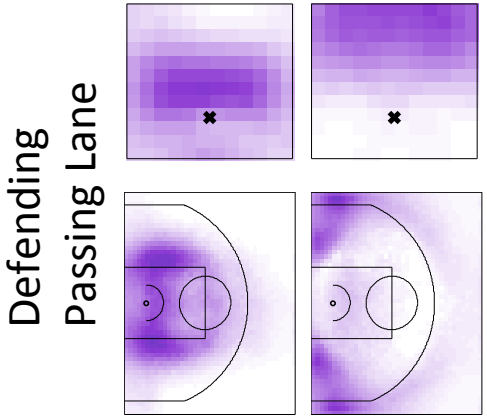
## (Subtractive Decomposition)

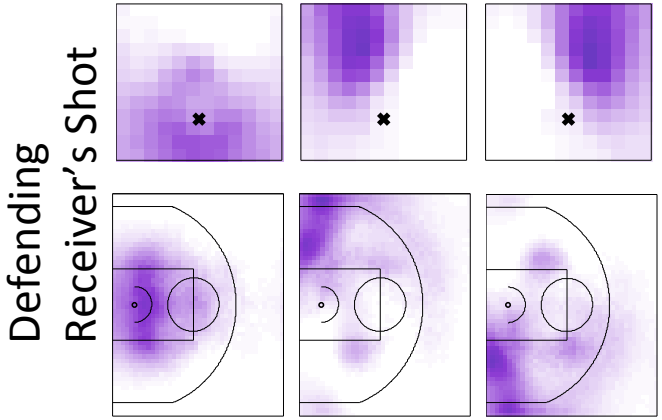


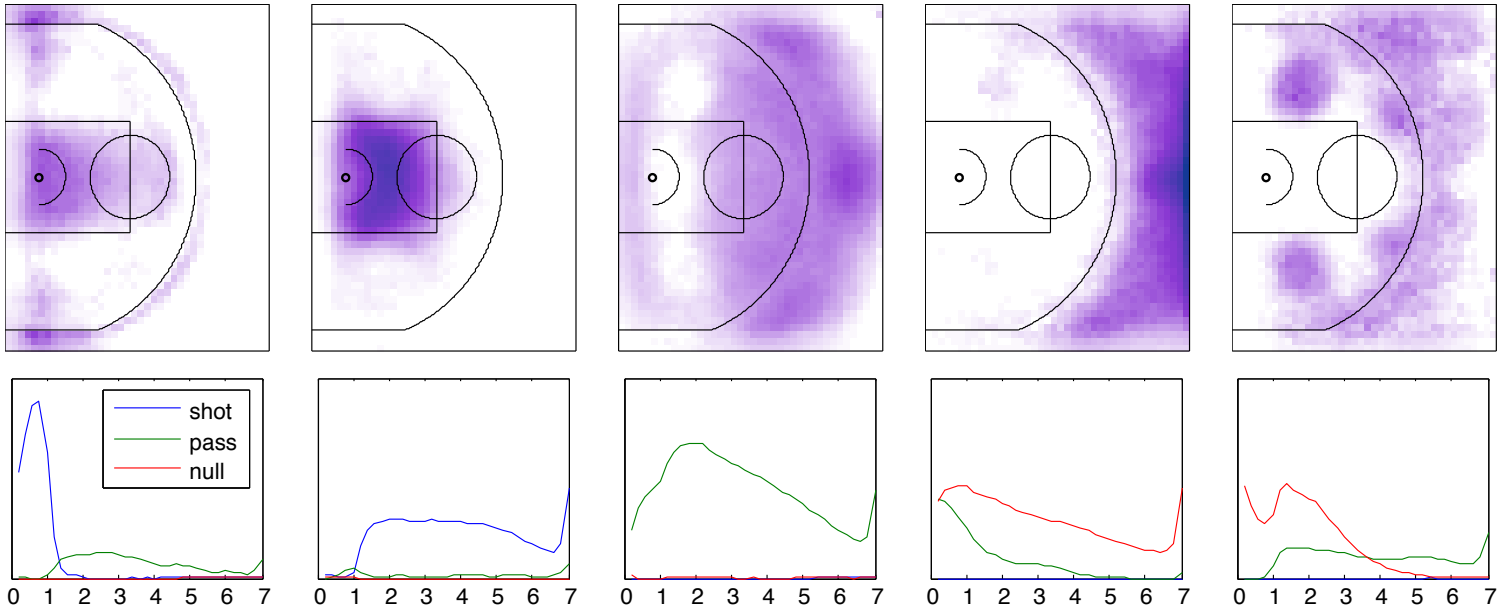




http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# Visualizing Time-of-Possession Factors




How long the ball handler has held on to the ball.

http://www.yisongyue.com/publications/icdm2014_bball_predict.pdf

# Recap: Latent Factor Spatial Models

- Multiclass prediction — E.g., shot, pass, etc.
- Spatial Model — Discretize into fine-grained cells (regularize neighbors to be similar) (compose different spatial factors)
  - Captures spatial structure
- Multitask prediction — One task per player
- Interpretable Model — Latent factor model; Shared representation across players; Non-negative coefficients
- Gradient Descent — Requires good initialization (NNMF of full model)

# Latent Collaborative Clustering

# Motivation: Richer User Interfaces


Information Bottleneck

We can only learn as much as interface permits

(We can only help user as much as interface permits)


**“ReGroup: Interactive Machine Learning for On-Demand Group Creation in Social Networks”**
**[**Ameershi et al., CHI 2012]

Search

Pinterest

Yisong

# Sweet Somethings

Unfollow Board | Send Board | 73 Pins | 101 Followers

Cinnamon Cake with Cinnamon-Cream Cheese Frosting - Recipes, Dinner Ideas, Healthy Recipes & Food Guide

Pinned from
piarecipes.com


from The Girl Who Ate Everything

**Cookies and Cream Cheesecakes**

7

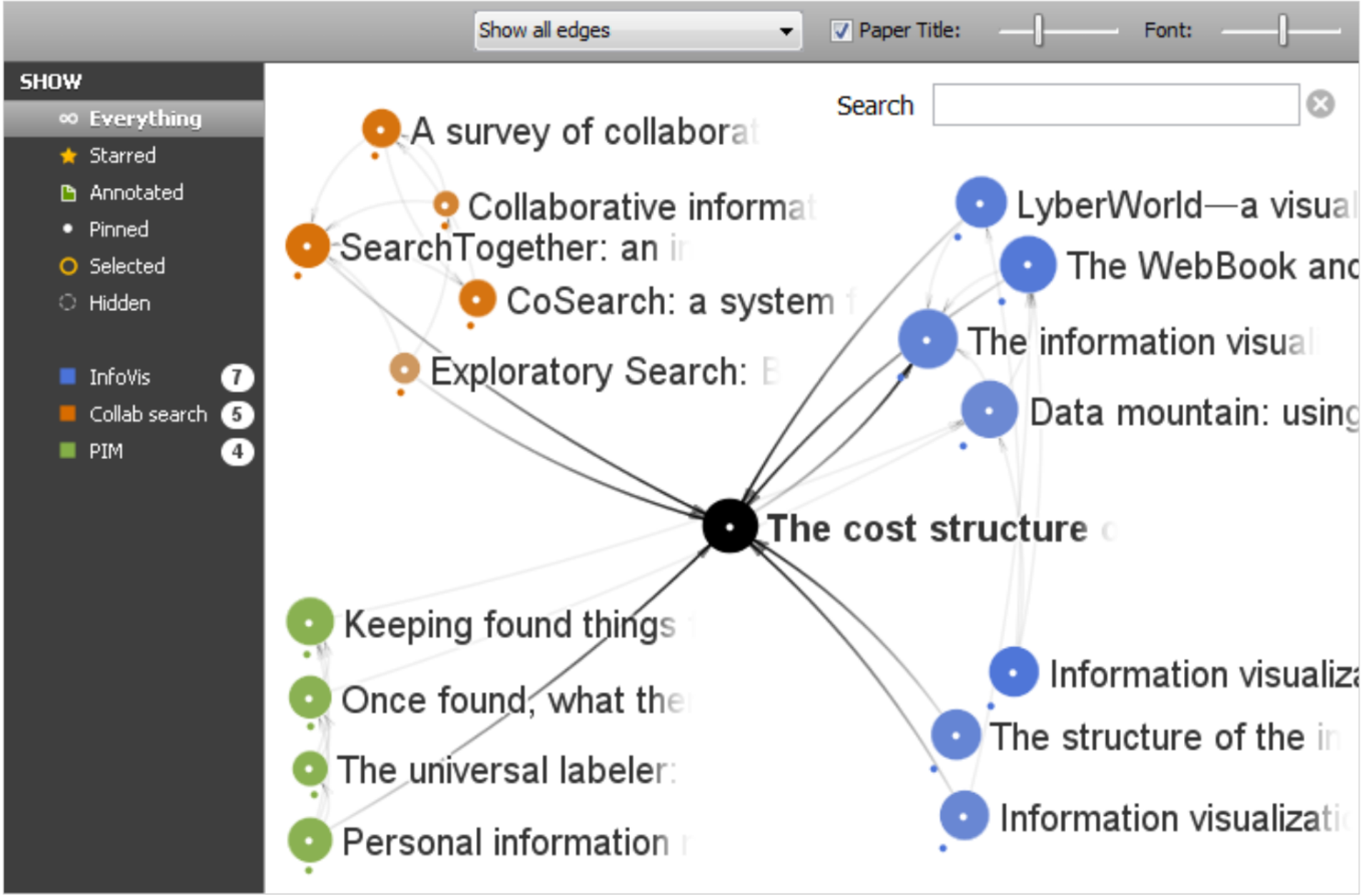
Pinned from
the-girl-who-ate-everything.com

Great visual aid on piping tips and their result for cake decorating.

2

Pinned from
media-cache-ak0.pinimg.com

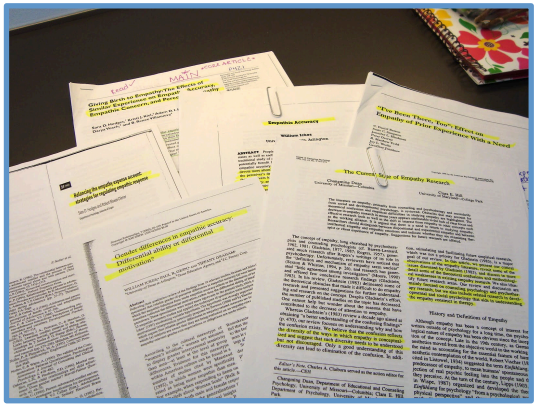
http://www.pinterest.com




**"Apolo: Making Sense of Large Network Data by Combining Rich User Interaction and Machine Learning"**
[Chau et al., CHI 2011]

# Demo
## (Goal Oriented Data Browsing)





**Planning a vacation**

**Planning meals for the week**

**Literature review for grant proposal**

# Recap: Collaborative Filtering

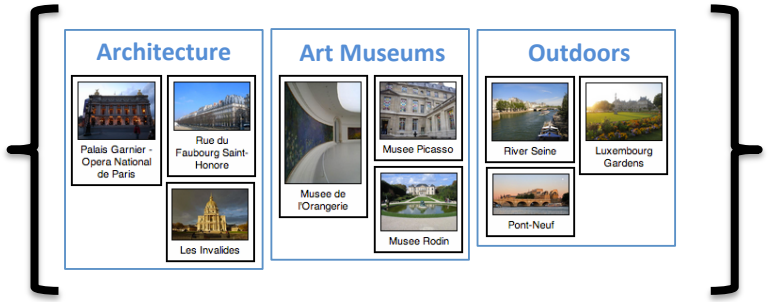
- **Training Data:** N items, M users
- $Y = \{Y_1 \dots Y_M\}$
- $Y_m = \left\{ \left( \text{Palais Garnier - Opera National de Paris}, \text{👍} \right), \left( \text{Promenade plantee,}, \text{👎} \right), \left( \text{Les Invalides}, \text{👍} \right), \dots \right\}$
- $Y_k = \left\{ \left( \text{River Seine}, \text{👍} \right), \left( \text{Napoleon's Tomb,}, \text{👎} \right), \left( \text{Promenade plantee,}, \text{👍} \right), \dots \right\}$

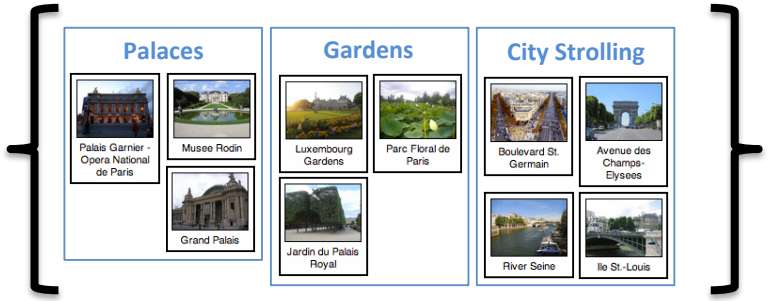
Each user rates a a subset of items

# Recap: Collaborative Filtering

- **Learning Goal:** $F(m,i) \approx y_{mi}$ (+1 or -1)
- $Y_m = \{ (\text{Palais Garnier - Opera National de Paris}, 👍), (\text{Promenade plantee}, 👎), (\text{Les Invalides}, 👍), \ldots \}$
- $F(m, \text{Palais Garnier - Opera National de Paris}) \approx +1$
- $F(m, \text{Promenade plantee}) \approx -1$
- **Prediction goal:** F(m,i) for new items

# Collaborative Clustering

- **Training Data:** N items, M users
- $Y = \{Y_1 \dots Y_M\}$
- $Y_m =$ 

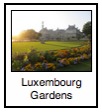
- $Y_k =$ 


Each user clusters a subset of items (partial clustering)

http://www.yisongyue.com/publications/www2014_collab_cluster.pdf

# Collaborative Clustering

- **Learning goal:** $F(m,i,j) \approx y_{mij}$ (+1 or -1)
- $Y_m =$


- $F(m,$  $,$  $) \approx +1$
- $F(m,$  $,$  $) \approx -1$

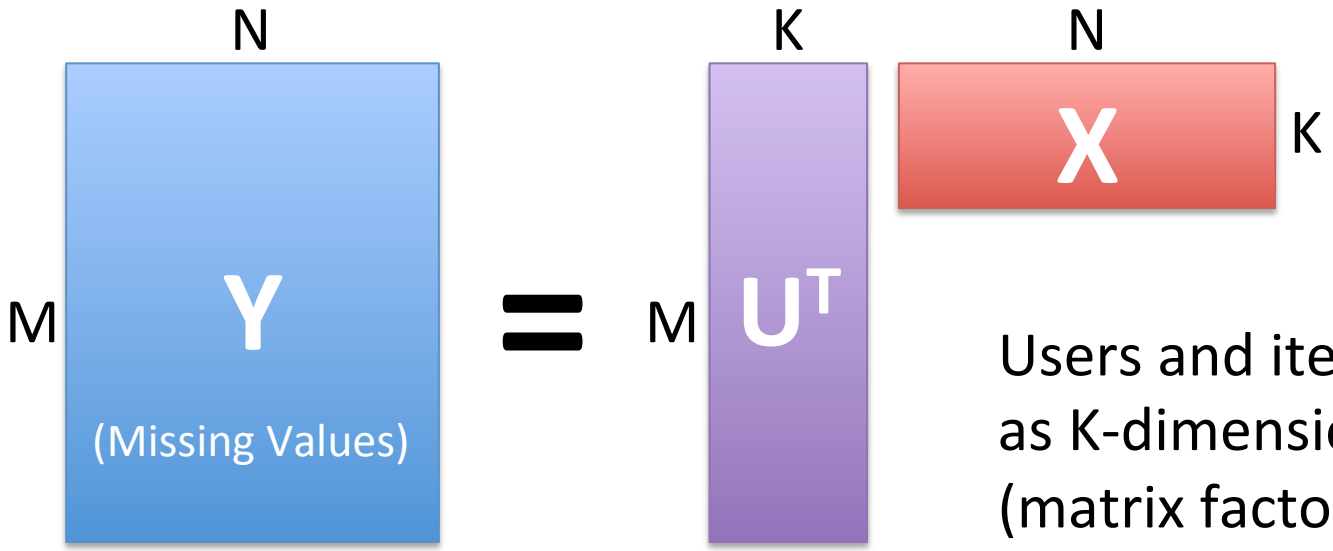
Clustering analogue to Collaborative Filtering

- **Prediction goal:** F(m,i,j) for new items

http://www.yisongyue.com/publications/www2014_collab_cluster.pdf

# Recap: Latent Factor Models

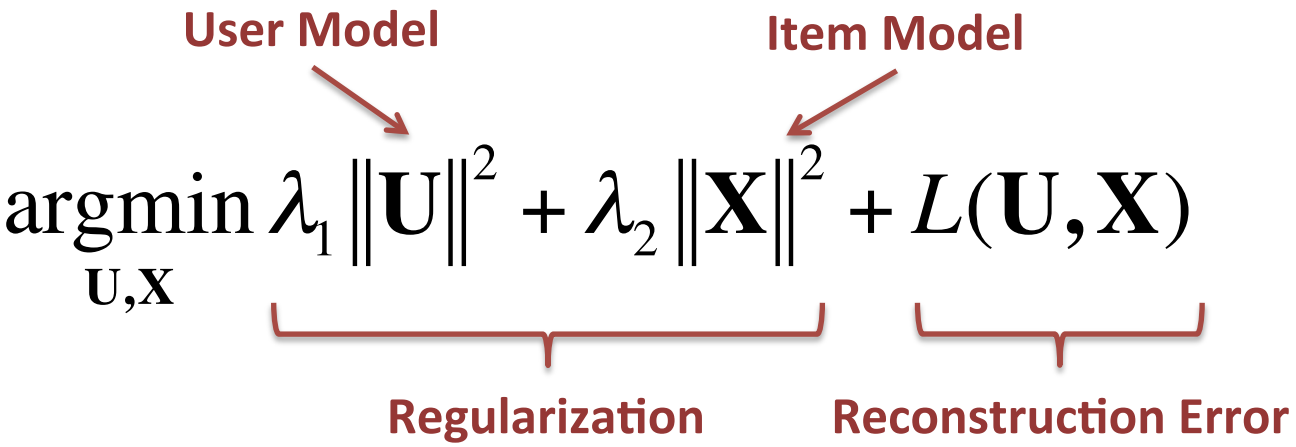
- **Collaborative Filtering:** N items, M users
- Users rate items ($y_{mi}$)




Users and items modeled as K-dimensional factors. (matrix factorization)

- **Prediction:** $F(m,i) = \langle u_m, x_i \rangle$

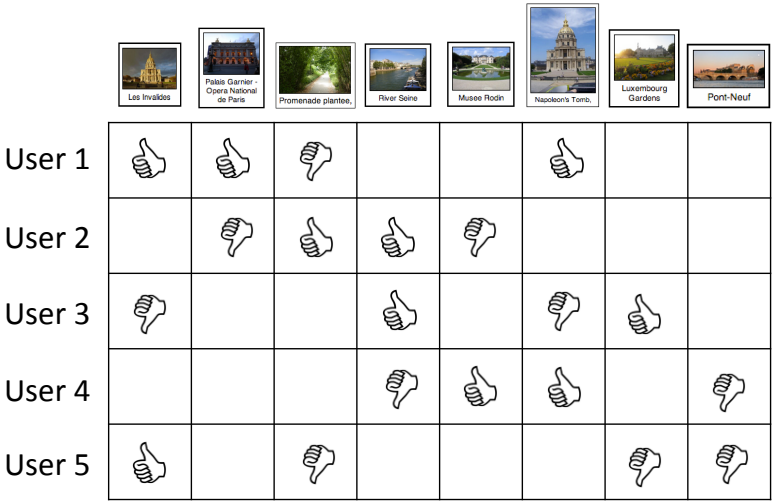
# Learning for Collaborative Filtering




$$L(\mathbf{U},\mathbf{X}) = \sum_m \sum_{i \in Y_m} \left( F(m,i) - y_{mi} \right)^2$$

$$F(m,i) = x_i^T u_m$$

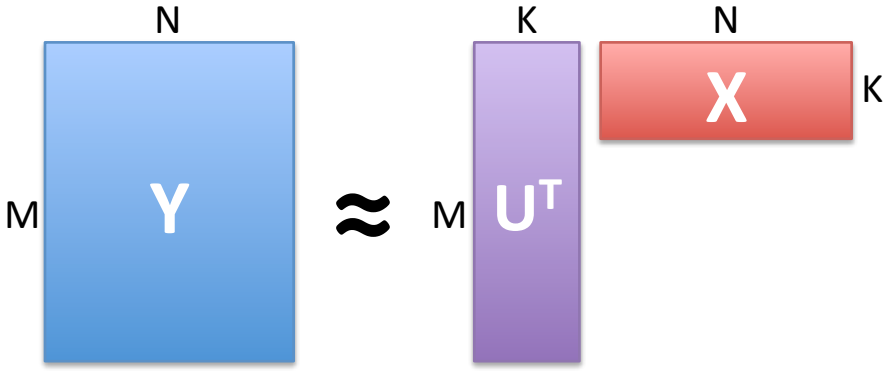
# Recap: Collaborative Filtering



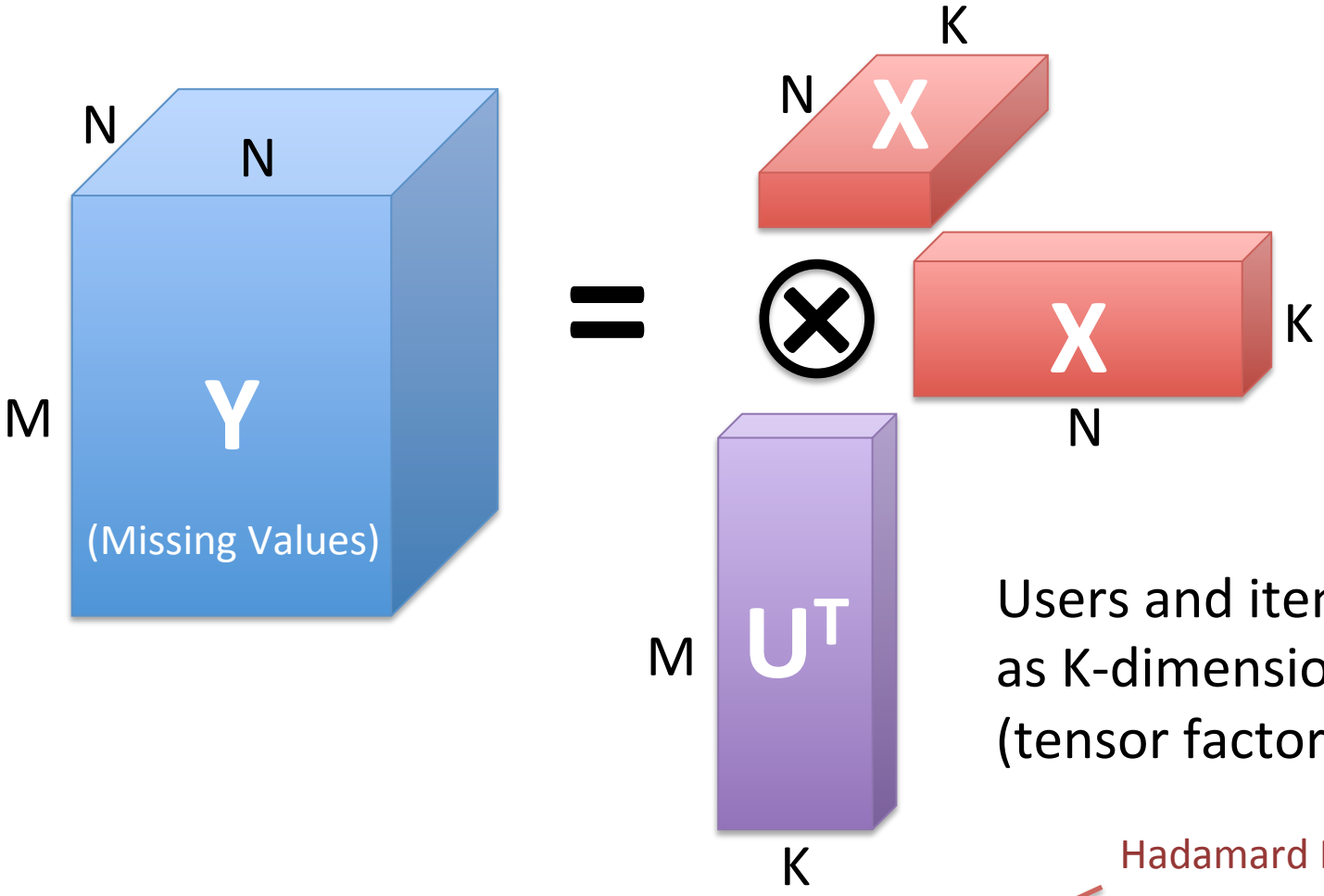

Can now predict missing values

$$F(m,i) = x_i^T u_m$$

Also known as matrix completion



# Latent Collaborative Clustering




Users and items modeled as K-dimensional factors. (tensor factorization)

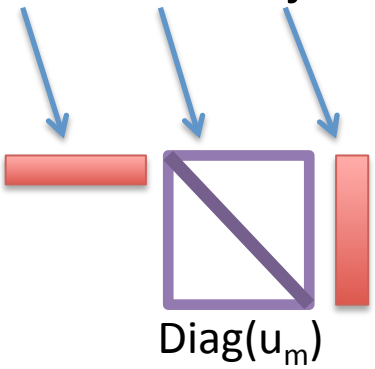
Hadamard Product

**Prediction:** $F(m,i,j) = x_i \circ u_m \circ x_j$

Actual model slightly more complicated

# Latent Collaborative Clustering

- $F(m,i,j) = x_i \circ u_m \circ x_j$
- $F(m,i,j) =$




$$F(m,i,j) = x_i^T \text{Diag}(u_m) x_j$$
$$= \langle x_i, x_j \rangle_{\text{Diag}(u_m)}$$

- $\text{Diag}(u_m)$ = user-specific transform
  - Metric learning (i.e., Mahanalobis)

http://www.yisongyue.com/publications/www2014_collab_cluster.pdf

Actual model slightly more complicated

# Learning

User Transform

Item Factor

$$\arg\min_{\mathbf{U}\geq\mathbf{0},\mathbf{X}} \lambda_1 \|\mathbf{U}\|^2 + \lambda_2 \|\mathbf{X}\|^2 + L(\mathbf{U},\mathbf{X})$$

Regularization

Reconstruction Error

Enforces Metric

$$L(\mathbf{U},\mathbf{X}) = \sum_m \sum_{i,j\in Y_m} \beta_{mij} \left(F(m,i,j) - y_{mij}\right)^2$$

$$F(m,i,j) = x_i^T Diag(u_m) x_j$$

Weight positive & negative feedback differently

(optimization details in paper)

http://www.yisongyue.com/publications/www2014_collab_cluster.pdf

# Learning

User Transform

Item Factor

$$\arg\min_{\mathbf{U}\geq\mathbf{0},\mathbf{X}} \lambda_1 \|\mathbf{U}\|^2 + \lambda_2 \|\mathbf{X}\|^2 + L(\mathbf{U},\mathbf{X})$$

- Alternating least squares optimization
  - (convex in **U** and each x, but not jointly)
- Iterative closed form solutions
  - (use ADMM to solve **U**≥0)

(optimization details in paper)

http://www.yisongyue.com/publications/www2014_collab_cluster.pdf

# Relationship to Metric Learning

- Metric Learning Learns a Transformation U:
  - Such that transforming features x by U minimizes loss:

Normal Metric Learning:

$$\underset{\mathbf{U} \geq \mathbf{0}}{\operatorname{argmin}} \lambda_1 \|\mathbf{U}\|^2 + L(\mathbf{U}, \mathbf{X})$$

Transformed Inner Product:

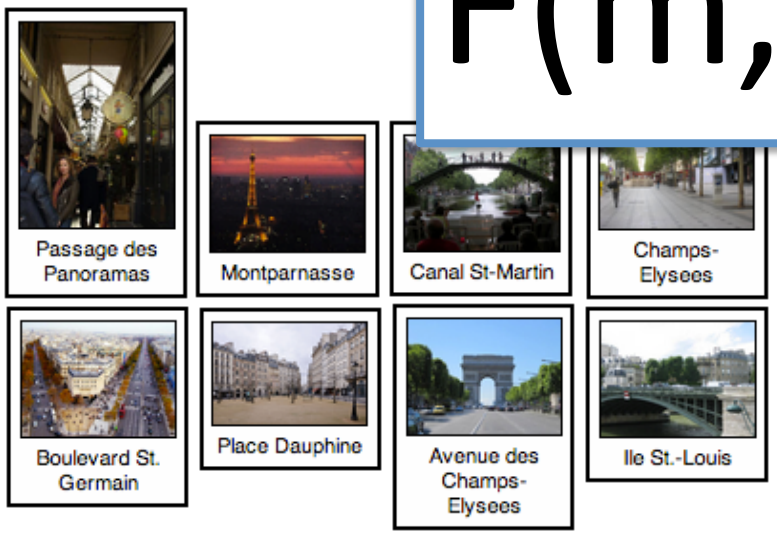
$$\langle x_i, x_j \rangle_U = x_i^T U x_j$$

- Latent Collaborate Clustering Learns both U and Features x
  - Useful if there many tasks, and few examples per task
  - Or if don't trust raw features x

# Input Clustering


# Input Clustering


F(m, River Seine, ?)




F(m, River Seine, ?)

# Data Collection

**New User Study:**

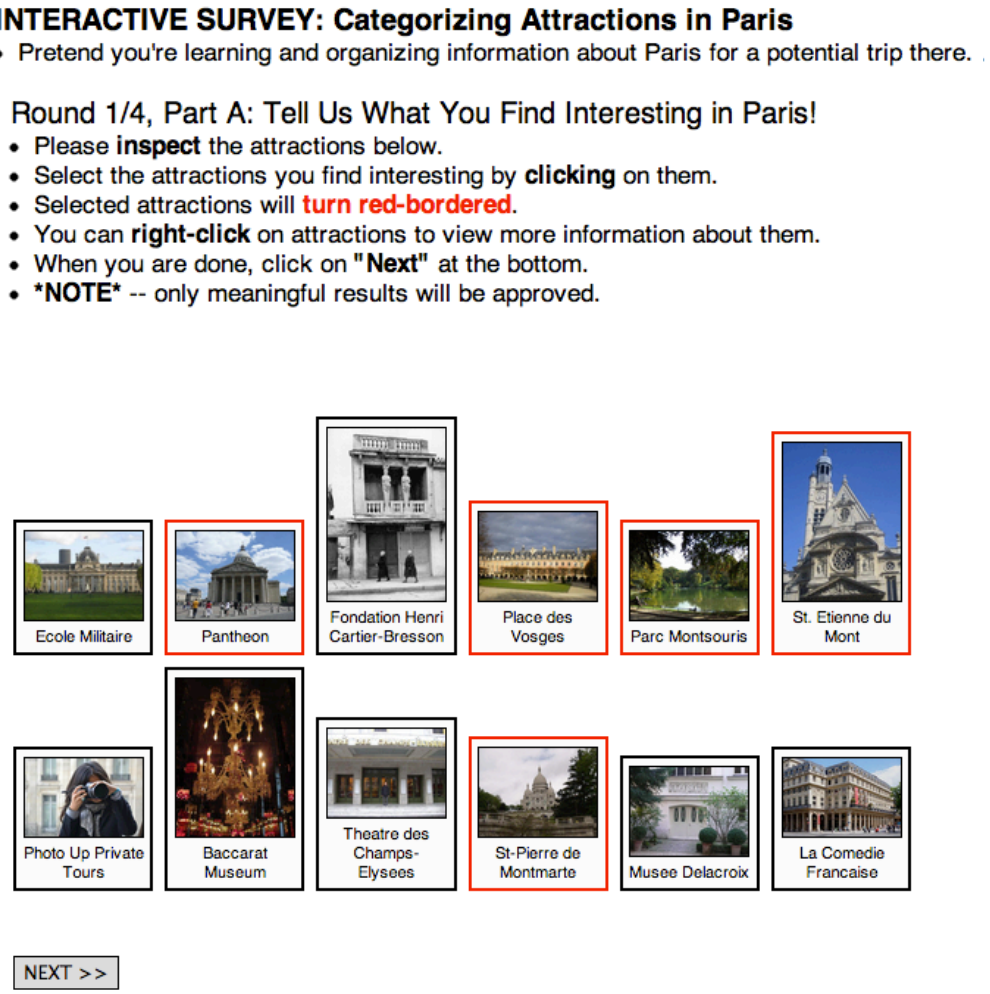
**Hypothetical trip to Paris**

**Mechanical Turk**

**Cluster interesting attractions**

**250 attractions (from TripAdvisor)**

**218 users**
**18.7 items per user**
**4.5 clusters per user**



# Data Collection

**New User Study:**

**Hypothetical trip to Paris**

**Mechanical Turk**

**Cluster interesting attractions**

**250 attractions (from TripAdvisor)**

**218 users**
**18.7 items per user**
**4.5 clusters per user**

# Data Collection

***"Very interesting - I feel like I learned a bit about Paris!"***

***"I have never been to Paris but now I want to go even more."***

***"We would just about sell our souls to get there! Thank you for letting me travel there vicariously through this HIT!!!"***
(HIT refers to "Human Intelligence Task")

***"Great format. Have been to Paris several times but still learned about many great new places to visit"***

# Features for Attractions

**2 sets of features**

**Wikipedia tf-idf**
(~3.5K features)

**Mechanical Turk Tagging (right)**
(~40 features)

**Used for baselines**

# Evaluation

- **For each test user:**
  - Hold out some attractions
  - Predict cluster membership (or new cluster)

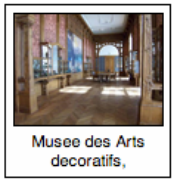
- **Feature-based baselines:**
  - Diagonal metric (per user)

    [Wagstaff & Cardie, 2000] [Xing et al., 2002] [Schultz & Joachims, 2003] [Davis et al., 2007] [Parameswaran & Weinberger, 2010]
  - Latent feature transform (significantly slower)

    [Blitzer & Weston, 2012]

# Prediction Tasks


- **Predict cluster membership**
  - Existing cluster:

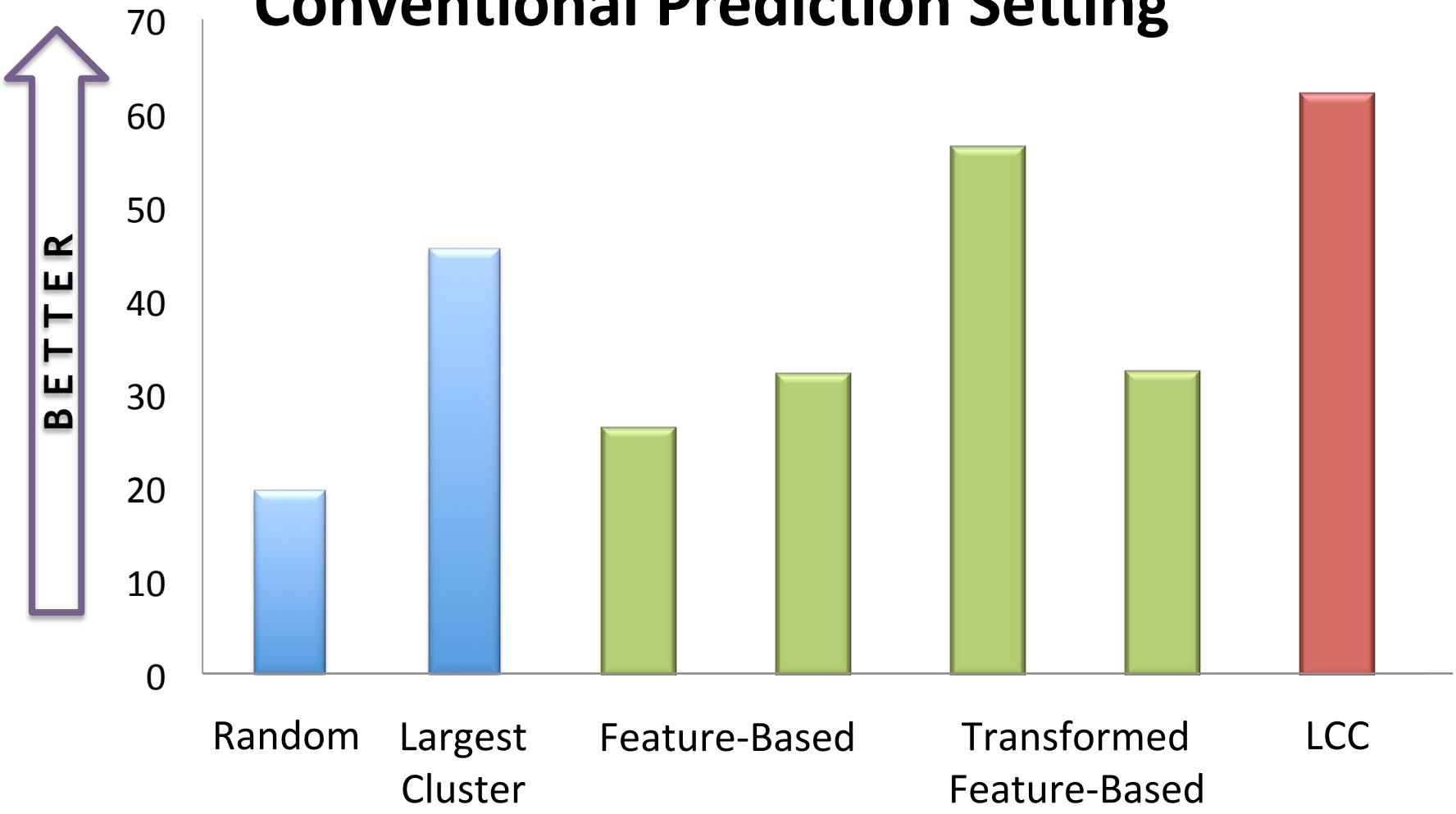



Cluster with highest average F(m,i,j). **(conventional classification)**

  - New cluster:

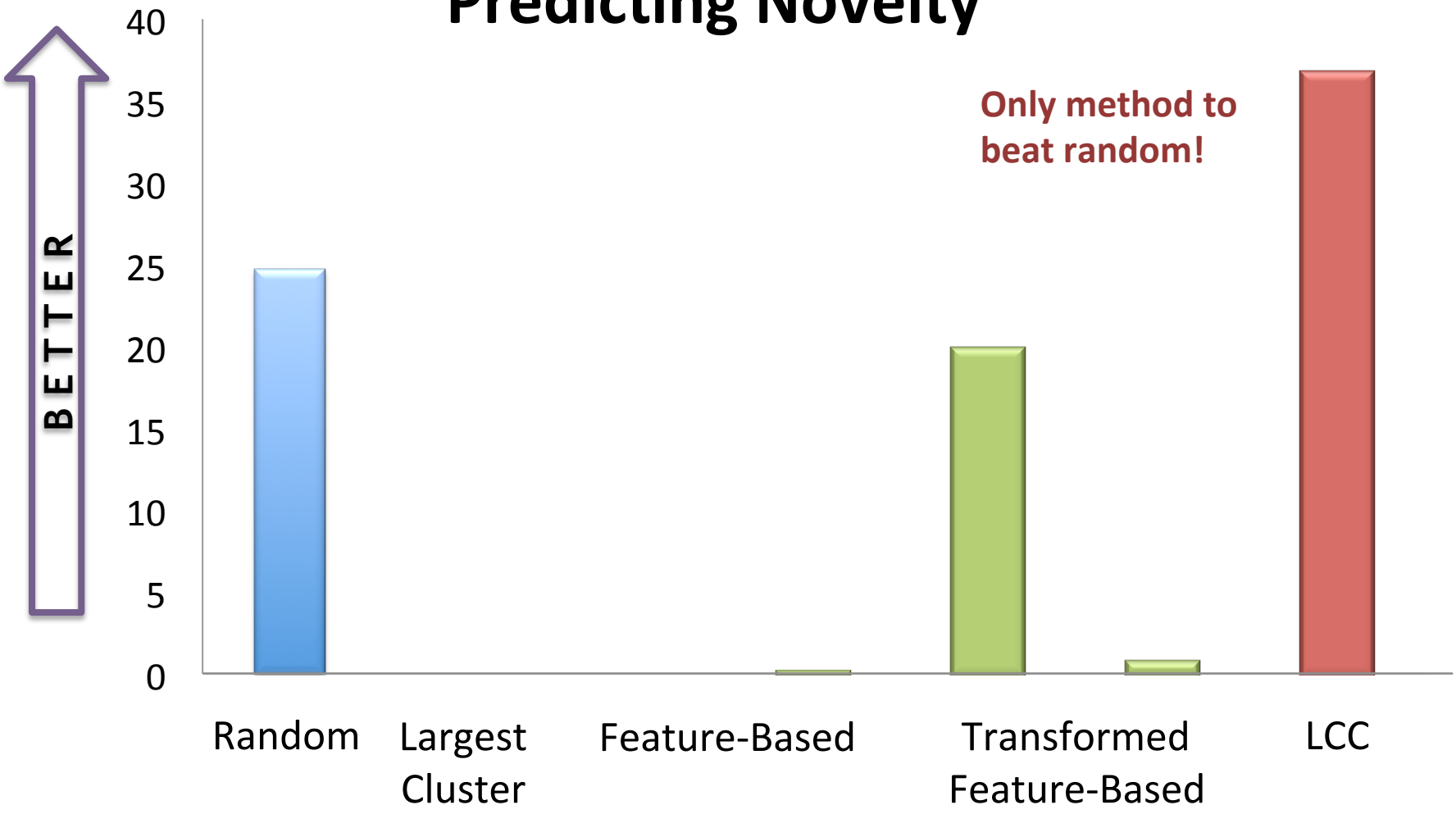

All clusters have negative F(m,i,j). **(predicting novelty)**

# Conventional Prediction Setting

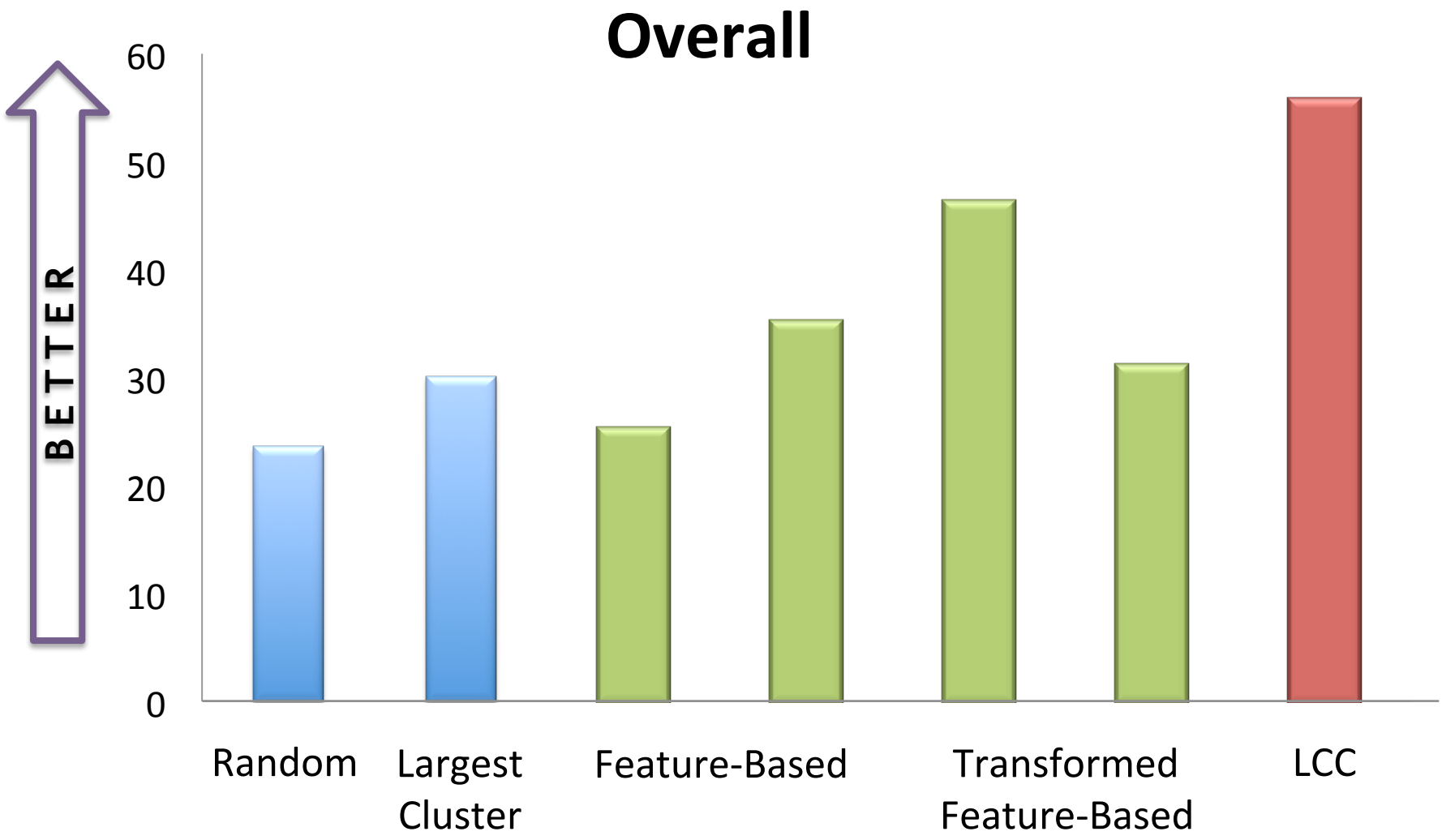



Hold 25% from each cluster (no empty clusters)
Predict cluster membership (conventional classification)

# Predicting Novelty




Hold one entire cluster
Predict new cluster (predict novelty)




Hold 50% of attractions at random (some clusters empty)
Predict cluster membership (or new cluster)

# Recap: Latent Collaborative Clustering

- Multitask Metric Learning & Feature Learning
  - Trained on partial clusterings created by individual users
- Individual factors x hard to visualize
  - Maybe easier if enforced non-negativity
- Maybe better served as an embedding model:

$$F(m,i,j) = \left\| x_i - x_j \right\|^2_{\mathrm{Diag}(u_m)}$$

# Recap: Latent Factor Models

- Great way to compactly represent data
  - Share across many tasks
- Can be very interpretable
  - At least the simpler versions
  - Tradeoff between interpretability and accuracy
- Particularly useful if you don't trust (or don't have) raw features
- **Next Week: Deep Learning**