

# Bilinear Prediction Using Low-Rank Models

Inderjit S. Dhillon  
Dept of Computer Science  
UT Austin

*26th International Conference on Algorithmic Learning Theory*  
Banff, Canada  
Oct 6, 2015

Joint work with C-J. Hsieh, P. Jain, N. Natarajan, H. Yu and K. Zhong

# Outline

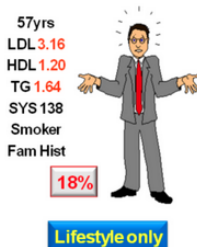
- Multi-Target Prediction
- Features on Targets: Bilinear Prediction
- Inductive Matrix Completion
  - 1 Algorithms
  - 2 Positive-Unlabeled Matrix Completion
  - 3 Recovery Guarantees
- Experimental Results

# Sample Prediction Problems

## Predicting stock prices

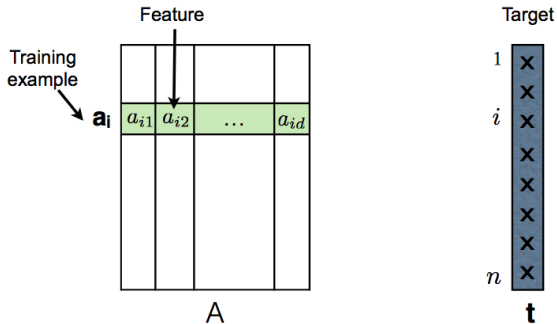


## Predicting risk factors in healthcare



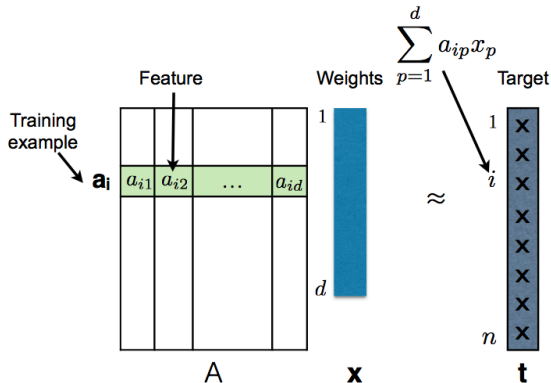
# Regression

- Real-valued responses (target)  $\mathbf{t}$
- Predict response for given input data (features)  $\mathbf{a}$



# Linear Regression

- Estimate target by a linear function of given data  $\mathbf{a}$ , i.e.  $\mathbf{t} \approx \hat{\mathbf{t}} = \mathbf{a}^T \mathbf{x}$ .

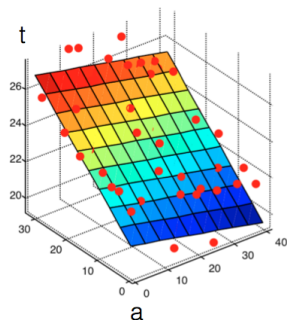
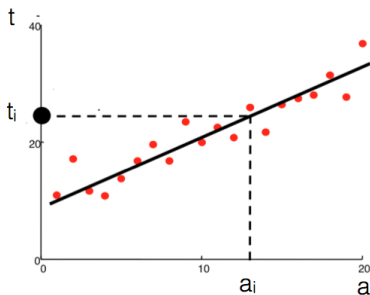


# Linear Regression: Least Squares

- Choose  $\mathbf{x}$  that minimizes

$$J_{\mathbf{x}} = \frac{1}{2} \sum_{i=1}^n (\mathbf{a}_i^T \mathbf{x} - t_i)^2$$

- Closed-form solution:  $\mathbf{x}^* = (A^T A)^{-1} A^T \mathbf{t}$ .



# Prediction Problems: Classification

## Spam detection

Gmail ▾

COMPOSE

Inbox (8,439)

Starred

Important

Sent Mail

Drafts

Notes

Less ▾

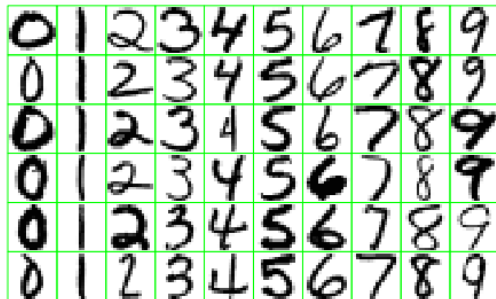
Chats

All Mail

Spam (298)

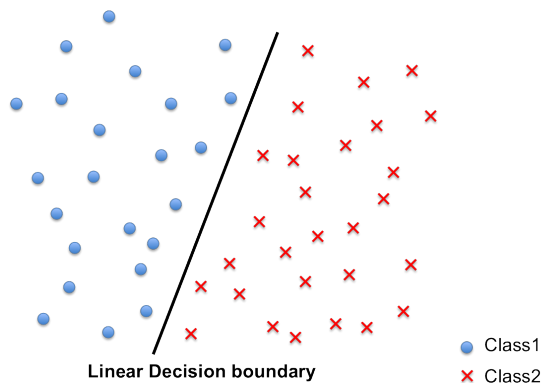
Trash

## Character Recognition



# Binary Classification

- Categorical responses (target)  $\mathbf{t}$
- Predict response for given input data (features)  $\mathbf{a}$
- Linear methods — decision boundary is a linear surface or hyperplane





# Linear Methods for Prediction Problems

Regression:

- Ridge Regression:  $J_{\mathbf{x}} = \frac{1}{2} \sum_{i=1}^n (\mathbf{a}_i^T \mathbf{x} - t_i)^2 + \lambda \|\mathbf{x}\|_2^2$ .
- Lasso:  $J_{\mathbf{x}} = \frac{1}{2} \sum_{i=1}^n (\mathbf{a}_i^T \mathbf{x} - t_i)^2 + \lambda \|\mathbf{x}\|_1$ .

Classification:

- Linear Support Vector Machines

$$J_{\mathbf{x}} = \frac{1}{2} \sum_{i=1}^n \max(0, 1 - t_i \mathbf{a}_i^T \mathbf{x}) + \lambda \|\mathbf{x}\|_2^2.$$

- Logistic Regression

$$J_{\mathbf{x}} = \frac{1}{2} \sum_{i=1}^n \log(1 + \exp(-t_i \mathbf{a}_i^T \mathbf{x})) + \lambda \|\mathbf{x}\|_2^2.$$

# Linear Prediction

Springer Series in Statistics

Trevor Hastie  
Robert Tibshirani  
Jerome Friedman

## The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Second Edition

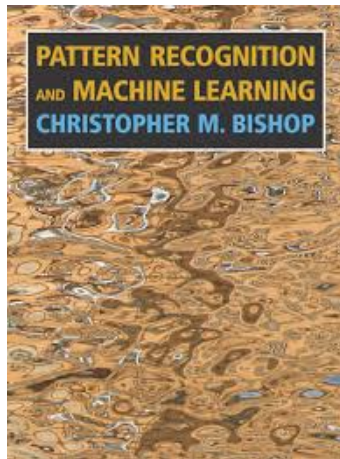
 Springer

### 3 Linear Methods for Regression

- 3.1 Introduction . . . . .
- 3.2 Linear Regression Models and Least Squares . . . . .
  - 3.2.1 Example: Prostate Cancer . . . . .
  - 3.2.2 The Gauss–Markov Theorem . . . . .
  - 3.2.3 Multiple Regression  
from Simple Univariate Regression . . . . .
  - 3.2.4 Multiple Outputs . . . . .
- 3.3 Subset Selection . . . . .
  - 3.3.1 Best-Subset Selection . . . . .

### 4 Linear Methods for Classification

- 4.1 Introduction . . . . .
- 4.2 Linear Regression of an Indicator Matrix . . . . .
- 4.3 Linear Discriminant Analysis . . . . .
  - 4.3.1 Regularized Discriminant Analysis . . . . .
  - 4.3.2 Computations for LDA . . . . .
  - 4.3.3 Reduced-Rank Linear Discriminant Analysis . . . . .
- 4.4 Logistic Regression . . . . .
  - 4.4.1 Fitting Logistic Regression Models . . . . .



## 3 Linear Models for Regression

- 3.1 Linear Basis Function Models . . . . .
  - 3.1.1 Maximum likelihood and least squares . . . . .
  - 3.1.2 Geometry of least squares . . . . .
  - 3.1.3 Sequential learning . . . . .
  - 3.1.4 Regularized least squares . . . . .
  - 3.1.5 Multiple outputs . . . . .
- 3.2 The Bias-Variance Decomposition . . . . .

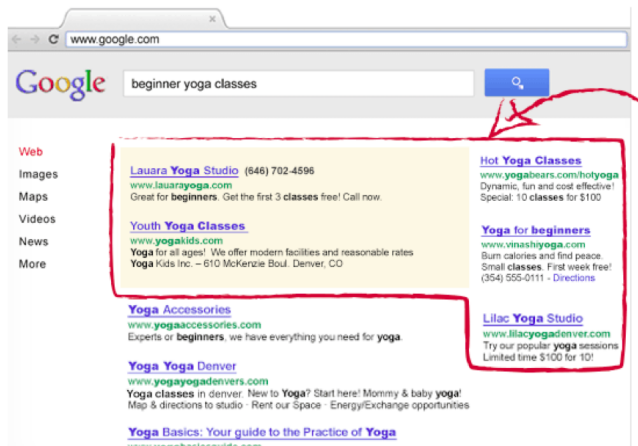
## 4 Linear Models for Classification

- 4.1 Discriminant Functions . . . . .
  - 4.1.1 Two classes . . . . .
  - 4.1.2 Multiple classes . . . . .
  - 4.1.3 Least squares for classification . . . . .
  - 4.1.4 Fisher's linear discriminant . . . . .
  - 4.1.5 Relation to least squares . . . . .
  - 4.1.6 Fisher's discriminant for multiple classes . . . . .
  - 4.1.7 The perceptron algorithm . . . . .
- 4.2 Probabilistic Generative Models . . . . .

# Multi-Target Prediction

# Modern Prediction Problems in Machine Learning

## Ad-word Recommendation



A screenshot of a Google search for "beginner yoga classes". The search results page shows several organic results and advertisements. A red hand-drawn box highlights a group of advertisements on the right side of the page. A red arrow points from the search button area towards the highlighted ads.

Search results for "beginner yoga classes":

- Web**
- Images
- Maps
- Videos
- News
- More

Advertisements (highlighted in red box):

- Laura Yoga Studio** (646) 702-4596  
[www.laurayoga.com](http://www.laurayoga.com)  
Great for **beginners**. Get the first 3 **classes** free! Call now.
- Youth Yoga Classes**  
[www.yogakids.com](http://www.yogakids.com)  
**Yoga** for all ages! We offer modern facilities and reasonable rates  
**Yoga** Kids Inc. - 610 McKenzie Boul. Denver, CO
- Hot Yoga Classes**  
[www.yogabears.com/hotyoga](http://www.yogabears.com/hotyoga)  
Dynamic, fun and cost effective!  
Special: 10 **classes** for \$100
- Yoga for beginners**  
[www.vinashiyoga.com](http://www.vinashiyoga.com)  
Burn calories and find peace.  
Small **classes**. First week free!  
(354) 555-0111 - [Directions](#)
- Lilac Yoga Studio**  
[www.lilacyogadenver.com](http://www.lilacyogadenver.com)  
Try our popular **yoga** sessions  
Limited time \$100 for 10!

Organic search results (below the highlighted ads):

- Yoga Accessories**  
[www.yogaaccessories.com](http://www.yogaaccessories.com)  
Experts or **beginners**, we have everything you need for **yoga**.
- Yoga Yoga Denver**  
[www.yogayogadenvers.com](http://www.yogayogadenvers.com)  
**Yoga classes** in denver. New to **Yoga**? Start here! Mommy & baby **yoga**!  
Map & directions to studio - Rent our Space - Energy/Exchange opportunities
- Yoga Basics: Your guide to the Practice of Yoga**  
[www.holisticnida.com](http://www.holisticnida.com)

# Modern Prediction Problems in Machine Learning

## Ad-word Recommendation

- geico auto insurance
- geico car insurance
- car insurance
- geico insurance
- need cheap auto insurance
- geico com
- car insurance coupon code



# Modern Prediction Problems in Machine Learning

## Wikipedia Tag Recommendation

- Learning in computer vision
- Machine learning
- Learning
- Cybernetics



The screenshot shows the Wikipedia article for "Machine learning". The article text includes a definition: "Machine learning is a scientific discipline that explores the construction and study of algorithms that can learn from data." It also mentions its application in various fields like computer science, statistics, and optimization. A sidebar on the right, titled "Machine learning and data mining", features a scatter plot and a list of related topics such as Classification, Clustering, Regression, and Support vector machine (SVM). A tag recommendation sidebar on the left lists "Learning in computer vision", "Machine learning", "Learning", and "Cybernetics".

**Machine learning**

From Wikipedia, the free encyclopedia

For the journal, see *Machine Learning (journal)*.  
See also: *Pattern recognition*

**Machine learning** is a scientific discipline that explores the construction and study of algorithms that can learn from data.<sup>[1]</sup> Such algorithms operate by building a model from example inputs and using that to make predictions or decisions,<sup>[2]</sup> rather than following strictly static program instructions. Machine learning is closely related to and often overlaps with *computational statistics*, a discipline that also specializes in prediction-making.

Machine learning is a subfield of computer science stemming from research into *artificial intelligence*.<sup>[3]</sup> It has strong ties to statistics and mathematical optimization, which deliver methods, theory and application domains to the field. Machine learning is employed in a range of computing tasks where designing and programming explicit, rule-based algorithms is infeasible. Example applications include *spam filtering*, *optical character recognition (OCR)*,<sup>[4]</sup> *search engines* and *computer vision*. Machine learning is sometimes conflated with *data mining*,<sup>[5]</sup> although that focuses more on exploratory data analysis.<sup>[6]</sup> Machine learning and *pattern recognition* can be viewed as two facets of the same field.<sup>[7][8]</sup>

When employed in industrial contexts, machine learning methods may be referred to as *predictive analytics* or *predictive modelling*.

**Contents** [hide]

- Overview
  - 1.1 Types of problems/tasks
  - 2 History and relationships to other fields
    - 2.1 Machine learning and statistics
  - 3 Theory
  - 4 Approaches
    - 4.1 Decision tree learning

en.wikipedia.org/wiki/Main\_Page

Neural Networks and Fuzzy Logic Models [\[edit\]](#), The MIT Press, Cambridge, MA, 608 pp., 288 illus., ISBN 0-262-11252-6.

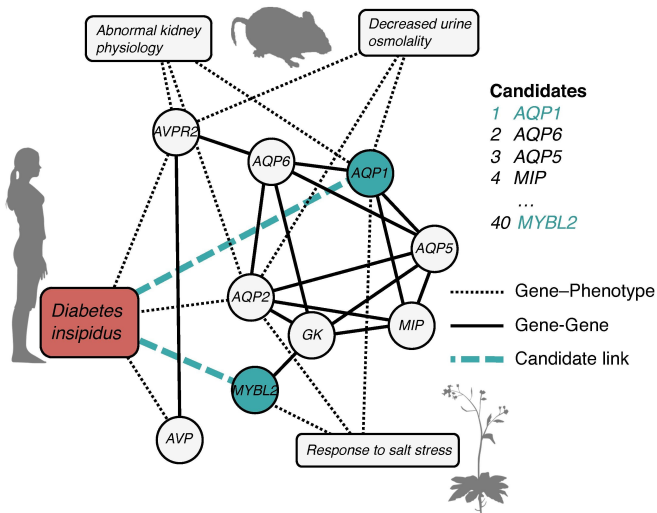
**External links** [edit]

- International Machine Learning Society
- Popular online course by Andrew Ng, at [Coursera](#); It uses GNU Octave. The course is a free version of [Stanford University's](#) actual course taught by Ng, whose lectures are also available for free.
- Machine Learning Video Lectures
- mlclass* is an academic database of open-source machine learning software.

Categories: Learning in computer vision | Machine learning | Learning | Cybernetics

# Modern Prediction Problems in Machine Learning

## Predicting causal disease genes





# Prediction with Multiple Targets

- In many domains, goal is to *simultaneously* predict multiple target variables
- **Multi-target regression**: targets are *real-valued*
- **Multi-label classification**: targets are *binary*

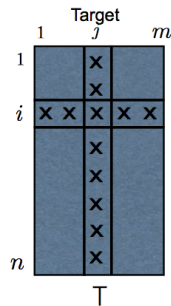
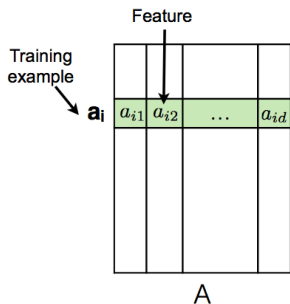
# Prediction with Multiple Targets

## Applications

- Bid word recommendation
- Tag recommendation
- Disease-gene linkage prediction
- Medical diagnoses
- Ecological modeling
- ...

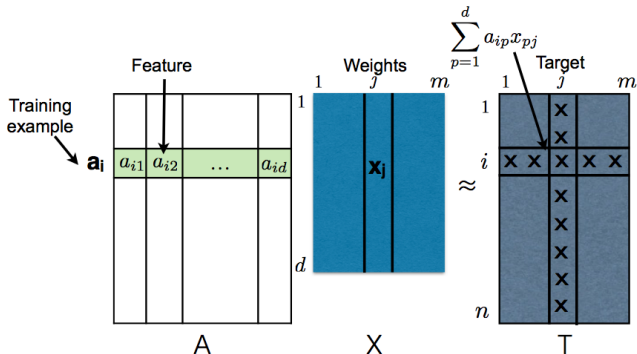
# Prediction with Multiple Targets

- Input data  $\mathbf{a}_i$  is associated with  $m$  targets,  $\mathbf{t}_i = (t_i^{(1)}, t_i^{(2)}, \dots, t_i^{(m)})$



# Multi-target Linear Prediction

- Basic model: Treat targets independently
- Estimate regression coefficients  $\mathbf{x}_j$  for each target  $j$



# Multi-target Linear Prediction

- Assume targets  $\mathbf{t}^{(j)}$  are independent
- Linear predictive model:  $\mathbf{t}_i \approx \mathbf{a}_i^T X$

# Multi-target Linear Prediction

- Assume targets  $\mathbf{t}^{(j)}$  are independent
- Linear predictive model:  $\mathbf{t}_i \approx \mathbf{a}_i^T X$
- Multi-target regression problem has a closed-form solution:

$$V_A \Sigma_A^{-1} U_A^T T = \arg \min_X \|T - AX\|_F^2$$

where  $A = U_A \Sigma_A V_A^T$  is the thin SVD of  $A$

# Multi-target Linear Prediction

- Assume targets  $\mathbf{t}^{(j)}$  are independent
- Linear predictive model:  $\mathbf{t}_i \approx \mathbf{a}_i^T X$
- Multi-target regression problem has a closed-form solution:

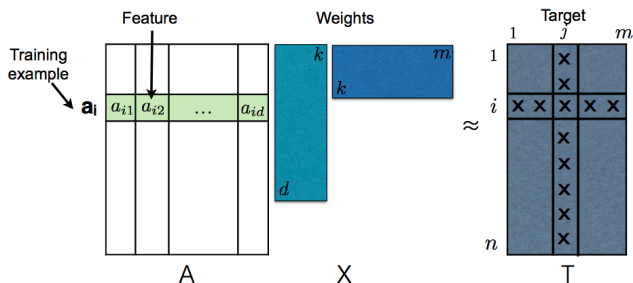
$$V_A \Sigma_A^{-1} U_A^T T = \arg \min_X \|T - AX\|_F^2$$

where  $A = U_A \Sigma_A V_A^T$  is the thin SVD of  $A$

In multi-label classification: **Binary Relevance** (independent binary classifier for each label)

# Multi-target Linear Prediction: Low-rank Model

- Exploit correlations between targets  $T$ , where  $T \approx AX$
- **Reduced-Rank Regression** [A.J. Izenman, 1974] — model the coefficient matrix  $X$  as *low-rank*



A. J. Izenman. *Reduced-rank regression for the multivariate linear model*. Journal of Multivariate Analysis 5.2 (1975): 248-264.



# Multi-target Linear Prediction: Low-rank Model

- $X$  is rank- $k$
- Linear predictive model:  $\mathbf{t}_i \approx \mathbf{a}_i^T X$

# Multi-target Linear Prediction: Low-rank Model

- $X$  is rank- $k$
- Linear predictive model:  $\mathbf{t}_i \approx \mathbf{a}_i^T X$
- Low-rank multi-target regression problem has a closed-form solution:

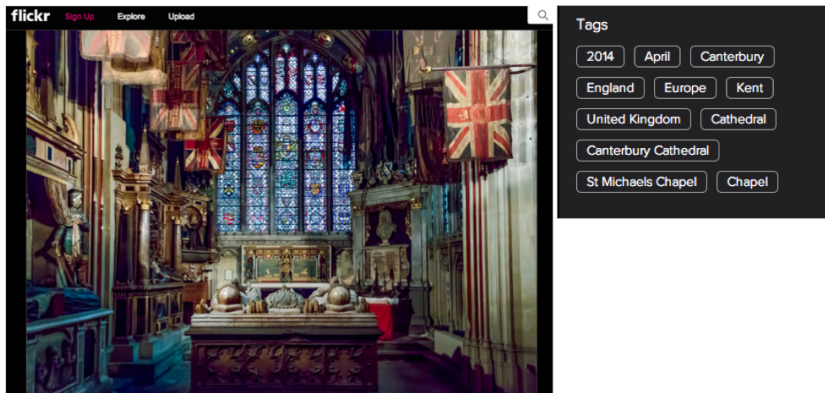
$$\begin{aligned} X^* &= \min_{X: \text{rank}(X) \leq k} \|T - AX\|_F^2 \\ &= \begin{cases} V_A \Sigma_A^{-1} U_A^T T_k & \text{if } A \text{ is full row rank,} \\ V_A \Sigma_A^{-1} M_k & \text{otherwise,} \end{cases} \end{aligned}$$

where  $A = U_A \Sigma_A V_A^T$  is the thin SVD of  $A$ ,  $M = U_A^T T$ , and  $T_k, M_k$  are the best rank- $k$  approximations of  $T$  and  $M$  respectively.

# Modern Challenges

# Multi-target Prediction with Missing Values

- In many applications, several observations (targets) may be *missing*
- E.g. Recommending tags for images and wikipedia articles



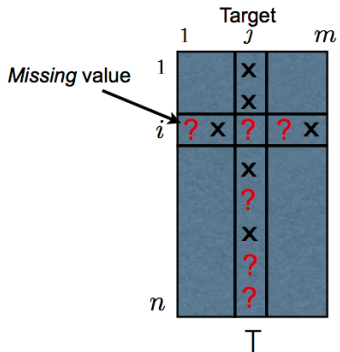
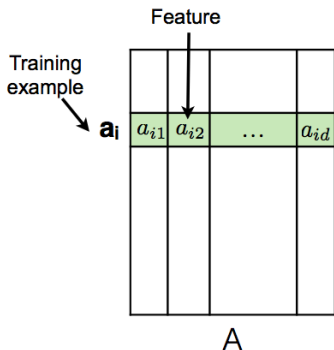
# Modern Prediction Problems in Machine Learning

## Ad-word Recommendation

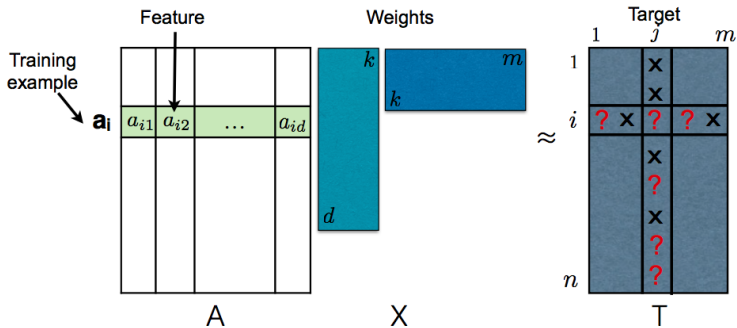
- geico auto insurance
- geico car insurance
- car insurance
- geico insurance
- need cheap auto insurance
- geico com
- car insurance coupon code



# Multi-target Prediction with Missing Values

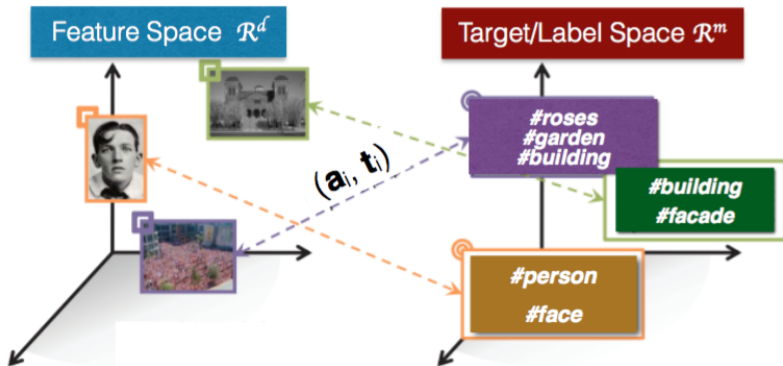


# Multi-target Prediction with Missing Values



- Low-rank model:  $\mathbf{t}_i = \mathbf{a}_i^T X$  where  $X$  is low-rank

# Canonical Correlation Analysis



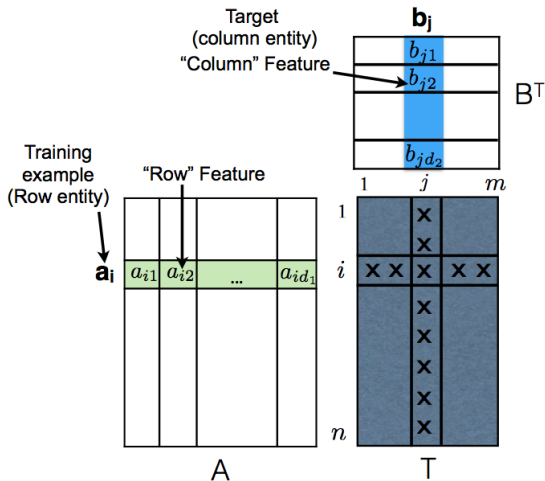


# Bilinear Prediction

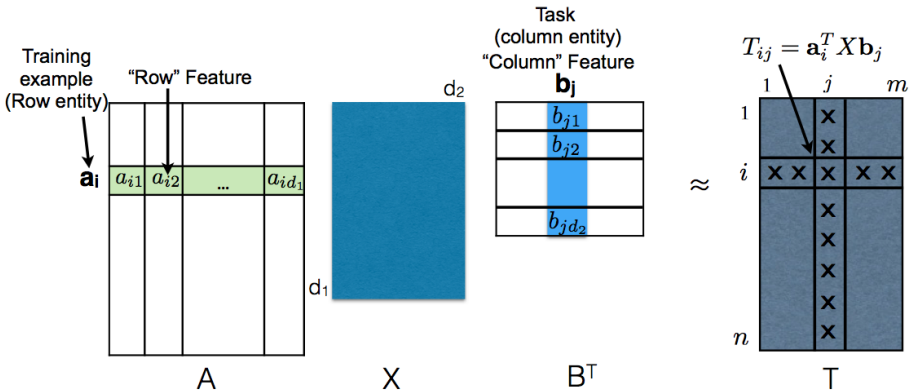
# Bilinear Prediction

- Augment multi-target prediction with *features* on targets as well
- Motivated by modern applications of machine learning — bioinformatics, auto-tagging articles
- Need to model *dyadic* or *pairwise* interactions
- Move from linear models to *bilinear* models — linear in input features *as well as* target features

# Bilinear Prediction



# Bilinear Prediction



# Bilinear Prediction

- Bilinear predictive model:  $T_{ij} \approx \mathbf{a}_i^T \mathbf{X} \mathbf{b}_j$

# Bilinear Prediction

- Bilinear predictive model:  $T_{ij} \approx \mathbf{a}_i^T X \mathbf{b}_j$
- Corresponding regression problem has a closed-form solution:

$$V_A \Sigma_A^{-1} U_A^T T U_B \Sigma_B^{-1} V_B^T = \arg \min_X \|T - AXB^T\|_F^2$$

where  $A = U_A \Sigma_A V_A^T$ ,  $B = U_B \Sigma_B V_B^T$  are the thin SVDs of  $A$  and  $B$

# Bilinear Prediction: Low-rank Model

- $X$  is rank- $k$
- Bilinear predictive model:  $T_{ij} \approx \mathbf{a}_i^T X \mathbf{b}_j$

# Bilinear Prediction: Low-rank Model

- $X$  is rank- $k$
- Bilinear predictive model:  $T_{ij} \approx \mathbf{a}_i^T X \mathbf{b}_j$
- Corresponding regression problem has a closed-form solution:

$$\begin{aligned} X^* &= \min_{X: \text{rank}(X) \leq k} \|T - AXB^T\|_F^2 \\ &= \begin{cases} V_A \Sigma_A^{-1} U_A^T T_k U_B \Sigma_B^{-1} V_B^T & \text{if } A, B \text{ are full row rank,} \\ V_A \Sigma_A^{-1} M_k \Sigma_B^{-1} V_B^T & \text{otherwise,} \end{cases} \end{aligned}$$

where  $A = U_A \Sigma_A V_A^T$ ,  $B = U_B \Sigma_B V_B^T$  are the thin SVDs of  $A$  and  $B$ ,  $M = U_A^T T U_B$ , and  $T_k$ ,  $M_k$  are the best rank- $k$  approximations of  $T$  and  $M$



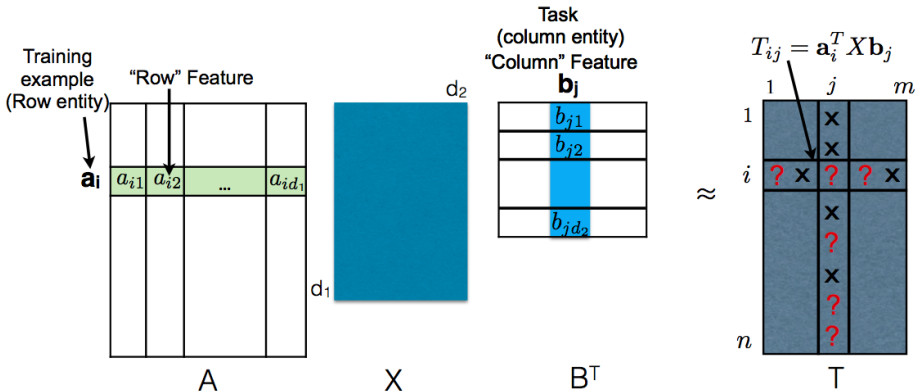
# Modern Challenges in Multi-Target Prediction

- Millions of targets
- Correlations among targets
- Missing values

# Modern Challenges in Multi-Target Prediction

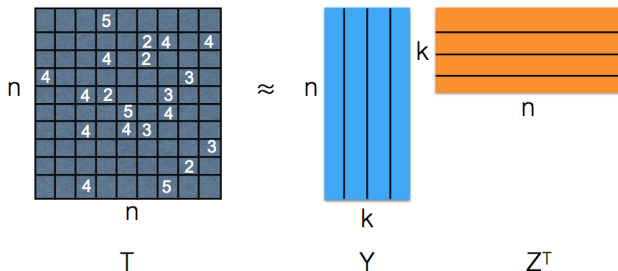
- Millions of targets (**Scalable**)
- Correlations among targets (**Low-rank**)
- Missing values (**Inductive Matrix Completion**)

# Bilinear Prediction with Missing Values



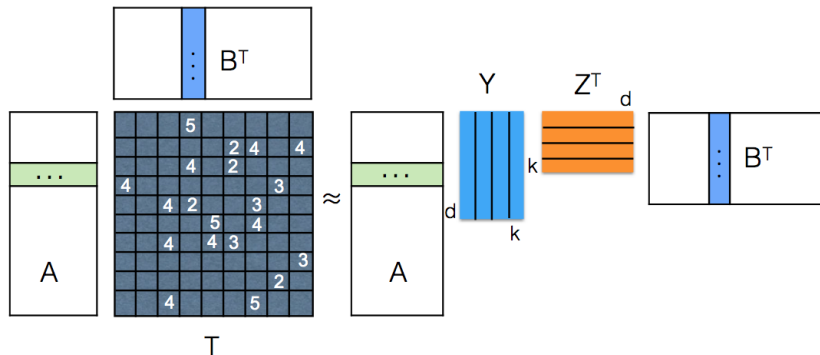
# Matrix Completion

- Missing Value Estimation Problem
  - Matrix Completion: Recover a low-rank matrix from observed entries
- Matrix Completion: exact recovery requires  $O(kn \log^2(n))$  samples, under the assumptions of:
  - 1 Uniform sampling
  - 2 Incoherence



# Inductive Matrix Completion

- Inductive Matrix Completion: Bilinear low-rank prediction with missing values
- Degrees of freedom in  $X$  are  $O(kd)$
- Can we get better sample complexity (than  $O(kn)$ )?



# Algorithm 1: Convex Relaxation

## 1 Nuclear-norm Minimization:

$$\begin{aligned} \min \quad & \|X\|_* \\ \text{s.t.} \quad & \mathbf{a}_i^T X \mathbf{b}_j = T_{ij}, (i,j) \in \Omega \end{aligned}$$

- Computationally expensive
- Sample complexity for exact recovery:  $O(kd \log d \log n)$
- Conditions for exact recovery:
  - **C1.** Incoherence on  $A, B$ .
  - **C2.** Incoherence on  $AU_*$  and  $BV_*$ , where  $X_* = U_* \Sigma_* V_*^T$  is the SVD of the ground truth  $X_*$
- **C1** and **C2** are satisfied with high probability when  $A, B$  are Gaussian

# Algorithm 1: Convex Relaxation

## Theorem (Recovery Guarantees for Nuclear-norm Minimization)

Let  $X_* = U_* \Sigma_* V_*^T \in \mathbb{R}^{d \times d}$  be the SVD of  $X_*$  with rank  $k$ . Assume  $A, B$  are orthonormal matrices w.l.o.g., satisfying the incoherence conditions. Then if  $\Omega$  is uniformly observed with

$$|\Omega| \geq O(kd \log d \log n),$$

the solution of nuclear-norm minimization problem is unique and equal to  $X_*$  with high probability.

The incoherence conditions are

$$\begin{aligned} \text{C1. } & \max_{i \in [n]} \|\mathbf{a}_i\|_2^2 \leq \frac{\mu d}{n}, \quad \max_{j \in [n]} \|\mathbf{b}_j\|_2^2 \leq \frac{\mu d}{n} \\ \text{C2. } & \max_{i \in [n]} \|U_*^T \mathbf{a}_i\|_2^2 \leq \frac{\mu_0 k}{n}, \quad \max_{j \in [n]} \|V_*^T \mathbf{b}_j\|_2^2 \leq \frac{\mu_0 k}{n} \end{aligned}$$

## Algorithm 2: Alternating Least Squares

- Alternating Least Squares (ALS):

$$\min_{Y \in \mathbb{R}^{d_1 \times k} Z \in \mathbb{R}^{d_2 \times k}} \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T Y Z^T \mathbf{b}_j - T_{ij})^2$$

- Non-convex optimization
- Alternately minimize w.r.t.  $Y$  and  $Z$



## Algorithm 2: Alternating Least Squares

- Computational complexity of ALS.
  - At  $h$ -th iteration, fixing  $Y_h$ , solve the least squares problem for  $Z_{h+1}$ :

$$\sum_{(i,j) \in \Omega} (\tilde{\mathbf{a}}_i^T Z_{h+1}^T \mathbf{b}_j) \mathbf{b}_j \tilde{\mathbf{a}}_i^T = \sum_{(i,j) \in \Omega} T_{ij} \mathbf{b}_j \tilde{\mathbf{a}}_i^T$$

where  $\tilde{\mathbf{a}}_i = Y_h^T \mathbf{a}_i$ . Similarly solve for  $Y_h$  when fixing  $Z_h$ .

- 1 Closed form:  $O(|\Omega|k^2d \times (\text{nnz}(A) + \text{nnz}(B))/n + k^3d^3)$ .
- 2 Vanilla conjugate gradient:  $O(|\Omega|k \times (\text{nnz}(A) + \text{nnz}(B))/n)$  per iteration.
- 3 Exploit the structure for conjugate gradient:

$$\sum_{(i,j) \in \Omega} (\tilde{\mathbf{a}}_i^T Z^T \mathbf{b}_j) \mathbf{b}_j \tilde{\mathbf{a}}_i^T = B^T D \tilde{A}$$

where  $D$  is a sparse matrix with  $D_{ji} = \tilde{\mathbf{a}}_i^T Z^T \mathbf{b}_j$ ,  $(i,j) \in \Omega$ , and  $\tilde{A} = AY_h$ . Only  $O((\text{nnz}(A) + \text{nnz}(B) + |\Omega|) \times k)$  per iteration.

## Algorithm 2: Alternating Least Squares

### Theorem (Convergence Guarantees for ALS)

Let  $X_*$  be a rank- $k$  matrix with condition number  $\beta$ , and  $T = AX_*B^T$ . Assume  $A, B$  are orthogonal w.l.o.g. and satisfy the incoherence conditions. Then if  $\Omega$  is uniformly sampled with

$$|\Omega| \geq O(k^4 \beta^2 d \log d),$$

then after  $H$  iterations of ALS,  $\|Y_H Z_{H+1}^T - X_*\|_2 \leq \epsilon$ , where  $H = O(\log(\|X_*\|_F/\epsilon))$ .

The incoherence conditions are:

$$\begin{aligned} \text{C1. } & \max_{i \in [n]} \|\mathbf{a}_i\|_2^2 \leq \frac{\mu d}{n}, \quad \max_{j \in [n]} \|b_j\|_2^2 \leq \frac{\mu d}{n} \\ \text{C2'. } & \max_{i \in [n]} \|Y_h^T \mathbf{a}_i\|_2^2 \leq \frac{\mu_0 k}{n}, \quad \max_{j \in [n]} \|Z_h^T b_j\|_2^2 \leq \frac{\mu_0 k}{n}, \end{aligned}$$

for all the  $Y_h$ 's and  $Z_h$ 's generated from ALS.

## Algorithm 2: Alternating Least Squares

- Proof sketch for ALS
  - Consider the case when the rank  $k = 1$ :

$$\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T y z^T \mathbf{b}_j - T_{ij})^2$$

## Algorithm 2: Alternating Least Squares

- Proof sketch for rank-1 ALS

$$\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T y z^T \mathbf{b}_j - T_{ij})^2$$

- (a) Let  $X_* = \sigma_* y_* z_*^T$  be the thin SVD of  $X_*$  and assume  $A$  and  $B$  are orthogonal w.l.o.g.
- (b) In the absence of missing values, ALS = Power method.

$$\frac{\partial \|A y_h z^T B^T - T\|_F^2}{\partial z} = 2B^T (B z y_h^T A^T - T^T) A y_h = 2(z \|y_h\|^2 - B^T T^T A y_h)$$

$$z_{h+1} \leftarrow (A^T T B)^T y_h ; \text{normalize } z_{h+1}$$

$$y_{h+1} \leftarrow (A^T T B) z_{h+1} ; \text{normalize } y_{h+1}$$

Note that  $A^T T B = A^T A X_* B^T B = X_*$  and the power method converges to the optimal.

## Algorithm 2: Alternating Least Squares

- Proof sketch for rank-1 ALS

$$\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T y z^T \mathbf{b}_j - T_{ij})^2$$

- (c) With missing values, ALS is a variant of power method with noise in each iteration

$$z_{h+1} \leftarrow QR\left( \underbrace{X_*^T y_h}_{\text{power method}} - \underbrace{\sigma_* N^{-1}((y_*^T y_h)N - \tilde{N})z_*}_{\text{noise term } \mathbf{g}} \right)$$

where  $N = \sum_{(i,j) \in \Omega} \mathbf{b}_j \mathbf{a}_i^T y_h y_h^T \mathbf{a}_i \mathbf{b}_j^T$ ,  $\tilde{N} = \sum_{(i,j) \in \Omega} \mathbf{b}_j \mathbf{a}_i^T y_h y_*^T \mathbf{a}_i \mathbf{b}_j^T$ .

- (d) Given **C1** and **C2'**, the noise term  $\mathbf{g} = \sigma_* N^{-1}((y_*^T y_h)N - \tilde{N})z_*$  becomes smaller as the iterate gets close to the optimal:

$$\|\mathbf{g}\|_2 \leq \frac{1}{99} \sqrt{1 - (y_h^T y_*)^2}$$

## Algorithm 2: Alternating Least Squares

- Proof sketch for rank-1 ALS

$$\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T y z^T \mathbf{b}_j - T_{ij})^2$$

- (e) Given **C1** and **C2'**, the first iterate  $y_0$  is well initialized, i.e.  $y_0^T y_* \geq 0.9$ , which guarantees the initial noise is small enough
- (f) The iterates can then be shown to linearly converge to the optimal:

$$1 - (z_{h+1}^T z_*)^2 \leq \frac{1}{2} (1 - (y_h^T z_*)^2)$$
$$1 - (y_{h+1}^T y_*)^2 \leq \frac{1}{2} (1 - (z_{h+1}^T y_*)^2)$$

## Algorithm 2: Alternating Least Squares

- Proof sketch for rank-1 ALS

$$\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T y z^T \mathbf{b}_j - T_{ij})^2$$

- (e) Given **C1** and **C2'**, the first iterate  $y_0$  is well initialized, i.e.  $y_0^T y_* \geq 0.9$ , which guarantees the initial noise is small enough
- (f) The iterates can then be shown to linearly converge to the optimal:

$$1 - (z_{h+1}^T z_*)^2 \leq \frac{1}{2} (1 - (y_h^T z_*)^2)$$
$$1 - (y_{h+1}^T y_*)^2 \leq \frac{1}{2} (1 - (z_{h+1}^T y_*)^2)$$

- Similarly, the rank- $k$  case can be proved.

# Inductive Matrix Completion: Sample Complexity

- Sample complexity of Inductive Matrix Completion (IMC) and Matrix Completion (MC).

methods	IMC	MC
Nuclear-norm	$O(kd \log n \log d)$	$kn \log^2 n$ (Recht, 2011)
ALS	$O(k^4 \beta^2 d \log d)$	$k^3 \beta^2 n \log n$ (Hardt, 2014)

where  $\beta$  is the condition number of  $X$

- In most cases,  $n \gg d$
- Incoherence conditions on  $A, B$  are required
  - Satisfied e.g. when  $A, B$  are Gaussian (no assumption on  $X$  needed)

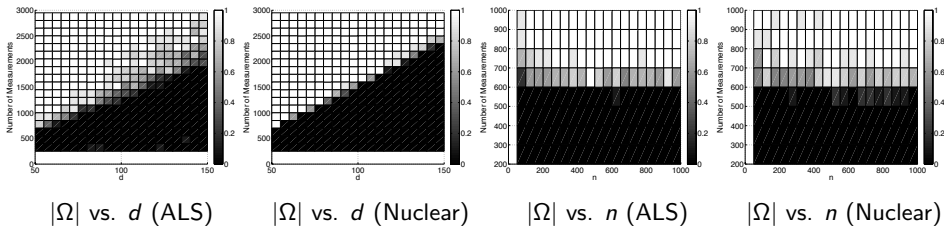
B. Recht. [A simpler approach to matrix completion](#). The Journal of Machine Learning Research 12 : 3413-3430 (2011).

M. Hardt. [Understanding alternating minimization for matrix completion](#). Foundations of Computer Science (FOCS), IEEE 55th Annual Symposium, pp. 651-660 (2014).



# Inductive Matrix Completion: Sample Complexity Results

- All matrices are sampled from Gaussian random distribution.
- Left two figures: fix  $k = 5$ ,  $n = 1000$  and change  $d$ .
- Right two figures: fix  $k = 5$ ,  $d = 50$  and change  $n$ .
- Darkness of the shading is proportional to the number of failures (repeated 10 times).

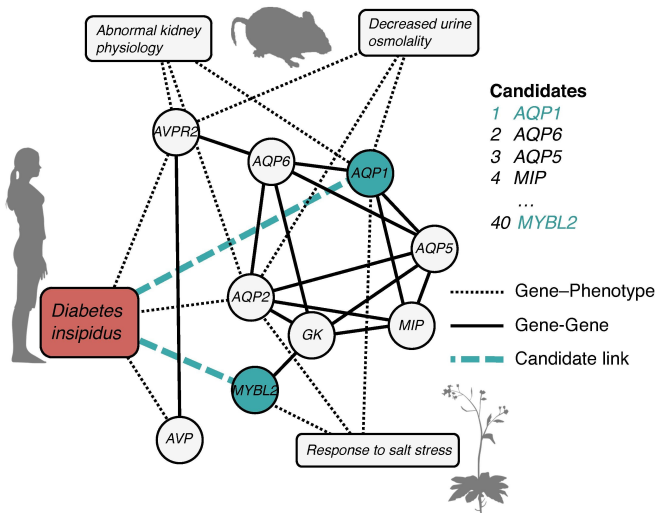


- Sample complexity is proportional to  $d$  while almost independent of  $n$  for both Nuclear-norm and ALS methods.

# Positive-Unlabeled Learning

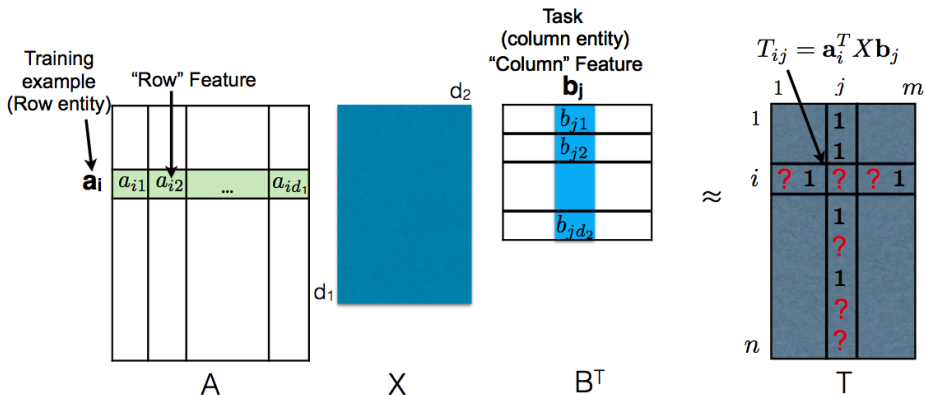
# Modern Prediction Problems in Machine Learning

## Predicting causal disease genes



# Bilinear Prediction: PU Learning

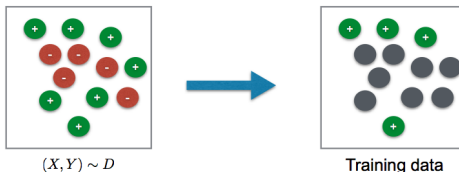
In many applications, only “positive” labels are observed



# PU Learning

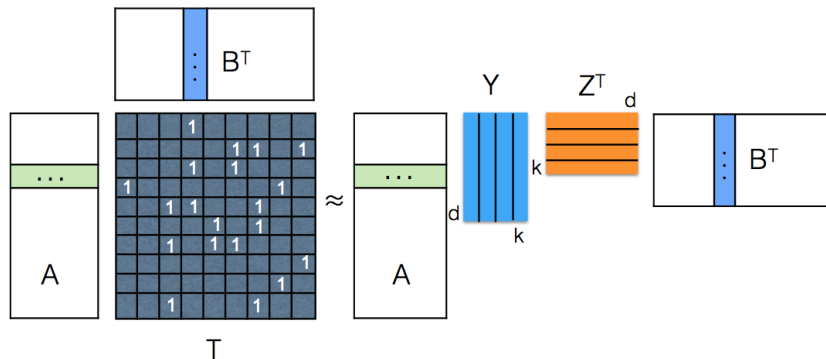
Learning Task	"Positives"	"Negatives"	"Unlabeled"
Supervised	✓	✓	
Semi-supervised	✓	✓	✓
Positive-Unlabeled (PU)	✓		✓
Unsupervised			✓

- No observations of the "negative" class available



# PU Inductive Matrix Completion

- Guarantees so far assume observations are sampled uniformly
- What can we say about the case when observations are all 1's ("positives")?
- Typically, 99% entries are missing ("unlabeled")



# PU Inductive Matrix Completion

- Inductive Matrix Completion:

$$\min_{X: \|X\|_* \leq t} \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T X \mathbf{b}_j - T_{ij})^2$$

- Commonly used PU strategy: Biased Matrix Completion

$$\min_{X: \|X\|_* \leq t} \alpha \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T X \mathbf{b}_j - T_{ij})^2 + (1 - \alpha) \sum_{(i,j) \notin \Omega} (\mathbf{a}_i^T X \mathbf{b}_j - 0)^2$$

Typically,  $\alpha > 1 - \alpha$  ( $\alpha \approx 0.9$ ).

# PU Inductive Matrix Completion

- Inductive Matrix Completion:

$$\min_{X: \|X\|_* \leq t} \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T X \mathbf{b}_j - T_{ij})^2$$

- Commonly used PU strategy: Biased Matrix Completion

$$\min_{X: \|X\|_* \leq t} \alpha \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T X \mathbf{b}_j - T_{ij})^2 + (1 - \alpha) \sum_{(i,j) \notin \Omega} (\mathbf{a}_i^T X \mathbf{b}_j - 0)^2$$

Typically,  $\alpha > 1 - \alpha$  ( $\alpha \approx 0.9$ ).

- We can show guarantees for the biased formulation

V. Sindhwani, S. S. Bucak, J. Hu, A. Mojsilovic. *One-class matrix completion with low-density factorizations*. ICDM, pp. 1055-1060. 2010.

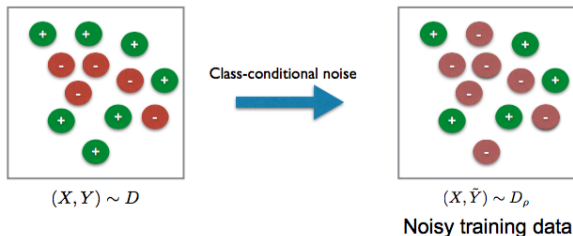


# PU Learning: Random Noise Model

- Can be formulated as learning with “class-conditional” noise

$$P(\tilde{Y} = -1|Y = +1) = \rho_{+1}$$
$$P(\tilde{Y} = +1|Y = -1) = \rho_{-1}$$

Becomes PU learning  
when  $\rho_{-1} = 0$



N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari. *Learning with Noisy Labels*. In Advances in Neural Information Processing Systems, pp. 1196-1204. 2013.

# PU Inductive Matrix Completion

A deterministic PU learning model

$M$				$T$			
0.2	0.1	0	0.8	0	0	0	1
0	0.6	0.1	0.9	0	1	0	1
0	0	0.8	0.1	0	0	1	0
0.9	0	0.2	0.1	1	0	0	0
0	0.6	0	1	0	1	0	1

$$T_{ij} = \begin{cases} 1 & \text{if } M_{ij} > 0.5, \\ 0 & \text{if } M_{ij} \leq 0.5 \end{cases}$$

# PU Inductive Matrix Completion

A deterministic PU learning model

$M$

0.2	0.1	0	0.8
0	0.6	0.1	0.9
0	0	0.8	0.1
0.9	0	0.2	0.1
0	0.6	0	1



$T$

0	0	0	1
0	1	0	1
0	0	1	0
1	0	0	0
0	1	0	1



$\tilde{T}$

?	?	?	1
?	1	?	?
?	?	1	?
1	?	?	?
?	?	?	1

- $P(\tilde{T}_{ij} = 0 | T_{ij} = 1) = \rho$  and  $P(\tilde{T}_{ij} = 0 | T_{ij} = 0) = 1$ .
- We are given *only*  $\tilde{T}$  but *not*  $T$  or  $M$
- Goal: Recover  $T$  given  $\tilde{T}$  (recovering  $M$  is not possible!)

# Algorithm 1: Biased Inductive Matrix Completion

$$\hat{X} = \min_{X: \|X\|_* \leq t} \alpha \sum_{(i,j) \in \Omega} (\mathbf{a}_i^T X \mathbf{b}_j - 1)^2 + (1 - \alpha) \sum_{(i,j) \notin \Omega} (\mathbf{a}_i^T X \mathbf{b}_j - 0)^2$$

- Rationale:

(a) Fix  $\alpha = (1 + \rho)/2$  and define  $\hat{T}_{ij} = I[(A\hat{X}B^T)_{ij} > 0.5]$

(b) The above problem is equivalent to:

$$\hat{X} = \min_{X: \|X\|_* \leq t} \sum_{i,j} \ell_\alpha((AXB^T)_{ij}, \tilde{T}_{ij})$$

where  $\ell_\alpha(x, \tilde{T}_{ij}) = \alpha \tilde{T}_{ij} (x - 1)^2 + (1 - \alpha)(1 - \tilde{T}_{ij})x^2$

(c) Minimizing  $\ell_\alpha$  loss is equivalent to minimizing the true error, i.e.

$$\frac{1}{mn} \sum_{ij} \ell_\alpha((AXB^T)_{ij}, \tilde{T}_{ij}) = C_1 \frac{1}{mn} \|\hat{T} - T\|_F^2 + C_2$$

# Algorithm 1: Biased Inductive Matrix Completion

## Theorem (Error Bound for Biased IMC)

Assume ground-truth  $X$  satisfies  $\|X\|_* \leq t$  (where  $M = AXB^T$ ). Define  $\hat{T}_{ij} = I[(A\hat{X}B^T)_{ij} > 0.5]$ ,  $\mathcal{A} = \max_i \|\mathbf{a}_i\|$  and  $\mathcal{B} = \max_i \|\mathbf{b}_i\|$ . If  $\alpha = \frac{1+\rho}{2}$ , then with probability at least  $1 - \delta$ ,

$$\frac{1}{n^2} \|T - \hat{T}\|_F^2 = O\left(\frac{\eta \sqrt{\log(2/\delta)}}{n(1-\rho)} + \frac{\eta t \mathcal{A} \mathcal{B} \sqrt{\log 2d}}{(1-\rho)n^{3/2}}\right)$$

where  $\eta = 4(1 + 2\rho)$ .

C-J. Hsieh, N. Natarajan, and I. S. Dhillon. *PU Learning for Matrix Completion*. In Proceedings of The 32nd International Conference on Machine Learning, pp. 2445-2453 (2015).

# Experimental Results

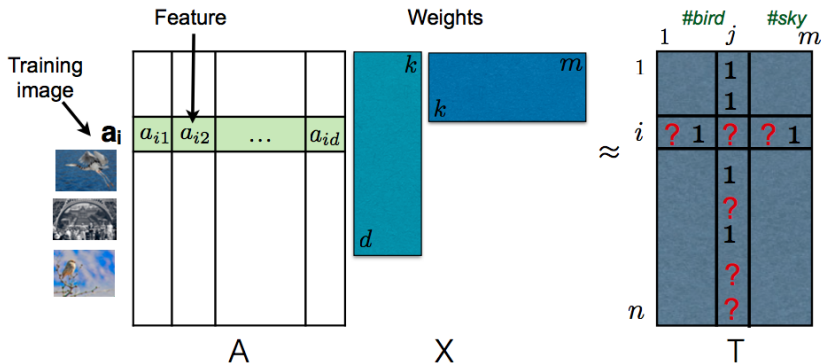
# Multi-target Prediction: Image Tag Recommendation

## NUS-Wide Image Dataset



- 161,780 training images
- 107,879 test images
- 1,134 features
- 1,000 tags

# Multi-target Prediction: Image Tag Recommendation



H. F. Yu, P. Jain, P. Kar, and I. S. Dhillon. *Large-scale Multi-label Learning with Missing Labels*. In Proceedings of The 31st International Conference on Machine Learning, pp. 593-601 (2014).



# Multi-target Prediction: Image Tag Recommendation

- Low-rank Model with  $k = 50$ :

	time(s)	prec@1	prec@3	AUC
LEML(ALS)	<b>574</b>	<b>20.71</b>	<b>15.96</b>	<b>0.7741</b>
WSABIE	4,705	14.58	11.37	0.7658

- Low-rank Model with  $k = 100$ :

	time(s)	prec@1	prec@3	AUC
LEML(ALS)	<b>1,097</b>	<b>20.76</b>	<b>16.00</b>	<b>0.7718</b>
WSABIE	6,880	12.46	10.21	0.7597

H. F. Yu, P. Jain, P. Kar, and I. S. Dhillon. [Large-scale Multi-label Learning with Missing Labels](#). In Proceedings of The 31st International Conference on Machine Learning, pp. 593-601 (2014).

# Multi-target Prediction: Wikipedia Tag Recommendation

## Wikipedia Dataset

The screenshot shows the Wikipedia article for "Machine learning". At the top, there's a navigation bar with "Article" and "Talk" tabs, and a search box. The main heading is "Machine learning". Below it, there's a sub-heading "From Wikipedia, the free encyclopedia". The article text starts with "For the journal, see Machine Learning (journal). See also: Pattern recognition". The main body of text defines machine learning as a scientific discipline that explores the construction and study of algorithms that can learn from data. It mentions applications like spam filtering, optical character recognition (OCR), search engines, and computer vision. A section titled "Machine learning and data mining" contains a scatter plot showing a positive correlation between two variables. Below the plot, there's a list of "Problems" including Classification, Clustering, Regression, Anomaly detection, Association rules, Reinforcement learning, Structured prediction, Feature learning, Online learning, and Semi-supervised learning. Grammar induction is also listed. Under "Supervised learning", there's a list of classification and regression methods: Decision trees, Ensembles (Bagging, Boosting, Random forest), k-NN, Linear regression, Naive Bayes, and Neural networks. At the bottom, there are navigation icons and a list of categories: Learning in computer vision, Machine learning, Learning, and Cybernetics.

- 881,805 training wiki pages
- 10,000 test wiki pages
- 366,932 features
- 213,707 tags

# Multi-target Prediction: Wikipedia Tag Recommendation

- Low-rank Model with  $k = 250$ :

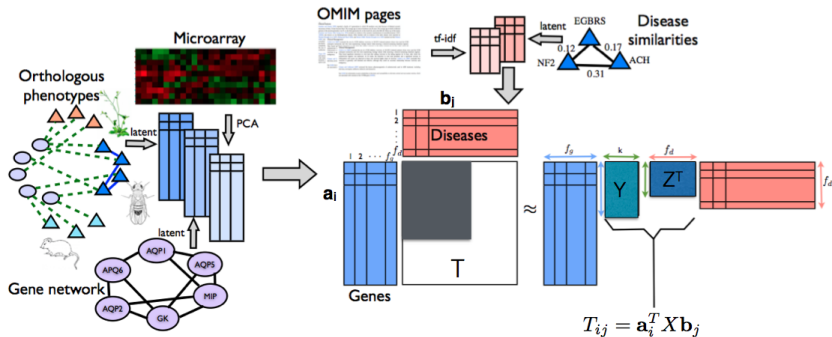
	time(s)	prec@1	prec@3	AUC
LEML(ALS)	<b>9,932</b>	<b>19.56</b>	14.43	<b>0.9086</b>
WSABIE	79,086	18.91	<b>14.65</b>	0.9020

- Low-rank Model with  $k = 500$ :

	time(s)	prec@1	prec@3	AUC
LEML(ALS)	<b>18,072</b>	<b>22.83</b>	<b>17.30</b>	<b>0.9374</b>
WSABIE	139,290	19.20	15.66	0.9058

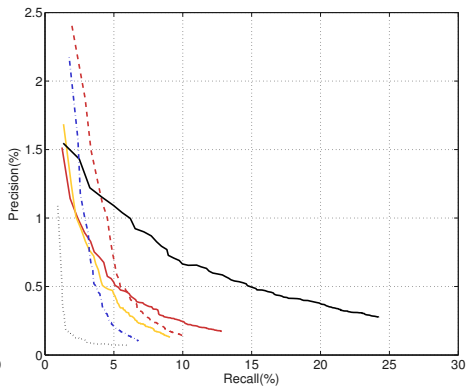
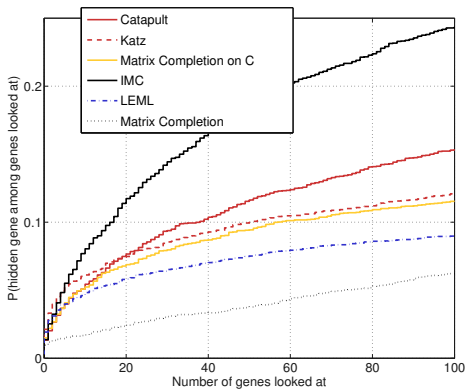
H. F. Yu, P. Jain, P. Kar, and I. S. Dhillon. *Large-scale Multi-label Learning with Missing Labels*. In Proceedings of The 31st International Conference on Machine Learning, pp. 593-601 (2014).

# PU Inductive Matrix Completion: Gene-Disease Prediction



N. Natarajan, and I. S. Dhillon. *Inductive matrix completion for predicting gene disease associations*. Bioinformatics, 30(12), i60-i68 (2014).

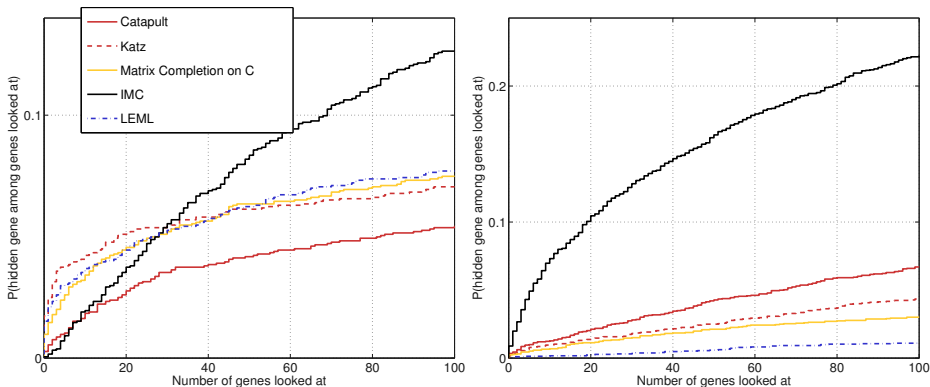
# PU Inductive Matrix Completion: Gene-Disease Prediction



Predicting gene-disease associations in the OMIM data set ([www.omim.org](http://www.omim.org)).

N. Natarajan, and I. S. Dhillon. *Inductive matrix completion for predicting gene disease associations*. Bioinformatics, 30(12), i60-i68 (2014).

# PU Inductive Matrix Completion: Gene-Disease Prediction



Predicting genes for diseases with *no* training associations.

N. Natarajan, and I. S. Dhillon. [Inductive matrix completion for predicting gene disease associations](#). Bioinformatics, 30(12), i60-i68 (2014).

# Conclusions and Future Work

- Inductive Matrix Completion:
  - Scales to millions of targets
  - Captures correlations among targets
  - Overcomes missing values
  - Extension to PU learning
- Much work to do:
  - Other structures: low-rank+sparse, low-rank+column-sparse (outliers)?
  - Different loss functions?
  - Handling “time” as one of the dimensions — incorporating smoothness through graph regularization?
  - Incorporating non-linearities?
  - Efficient (parallel) implementations?
  - Improved recovery guarantees?

# References

- [1] P. Jain, and I. S. Dhillon. *Provable inductive matrix completion*. arXiv preprint arXiv:1306.0626 (2013).
- [2] K. Zhong, P. Jain, I. S. Dhillon. *Efficient Matrix Sensing Using Rank-1 Gaussian Measurements*. In Proceedings of The 26th Conference on Algorithmic Learning Theory (2015).
- [3] N. Natarajan, and I. S. Dhillon. *Inductive matrix completion for predicting gene disease associations*. Bioinformatics, 30(12), i60-i68 (2014).
- [4] H. F. Yu, P. Jain, P. Kar, and I. S. Dhillon. *Large-scale Multi-label Learning with Missing Labels*. In Proceedings of The 31st International Conference on Machine Learning, pp. 593-601 (2014).
- [5] C-J. Hsieh, N. Natarajan, and I. S. Dhillon. *PU Learning for Matrix Completion*. In Proceedings of The 32nd International Conference on Machine Learning, pp. 2445-2453 (2015).