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

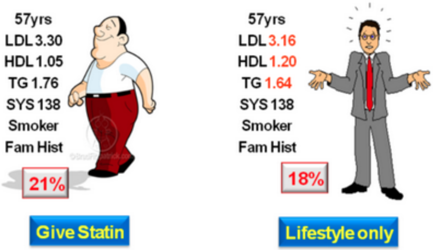
Outline

- 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

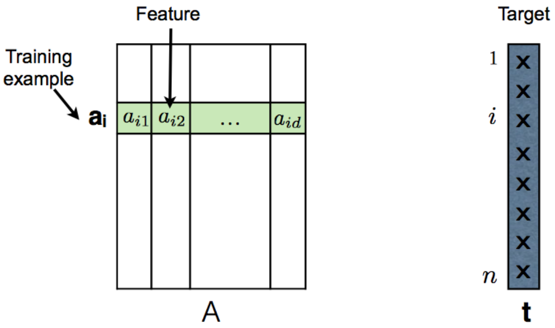
Predicting stock prices

Predicting risk factors in healthcare



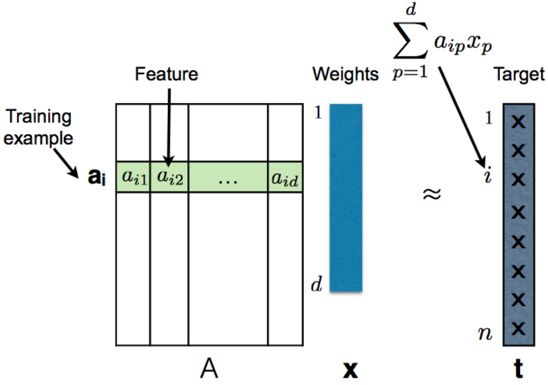
Regression

- Real-valued responses (target) **t**
- Predict response for given input data (features) **a**



Linear Regression

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

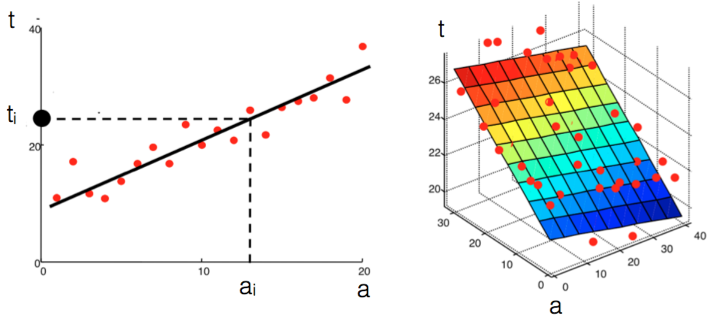


Linear Regression: Least Squares

- Choose **x** that minimizes

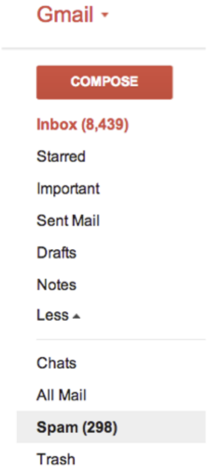
$$J_{\mathbf{x}} = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{a}_i^T\mathbf{x} - t_i)^2$$

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

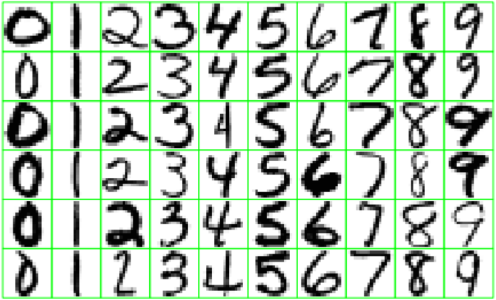


Prediction Problems: Classification

Spam detection

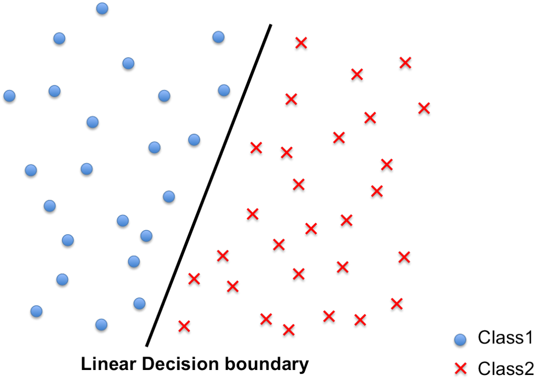


Character Recognition



Binary Classification

- Categorical responses (target) **t**
- Predict response for given input data (features) **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:

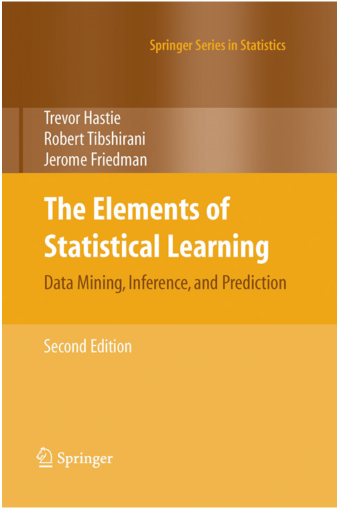
- Linear Support Vector Machines

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

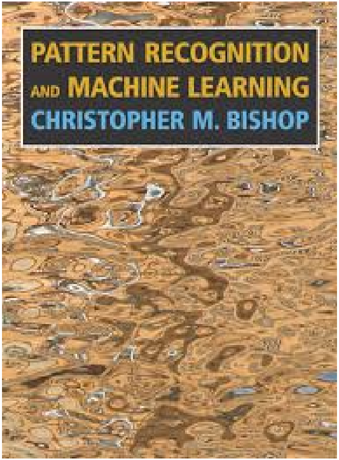
- Logistic Regression

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

Linear Prediction



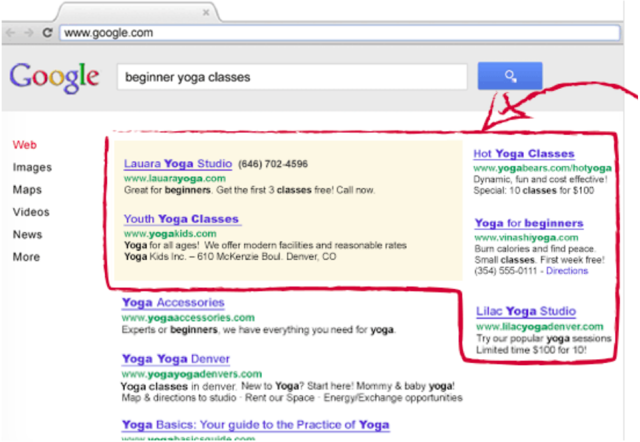
Linear Prediction



Multi-Target Prediction

Modern Prediction Problems in Machine Learning

Ad-word Recommendation



Modern Prediction Problems in Machine Learning

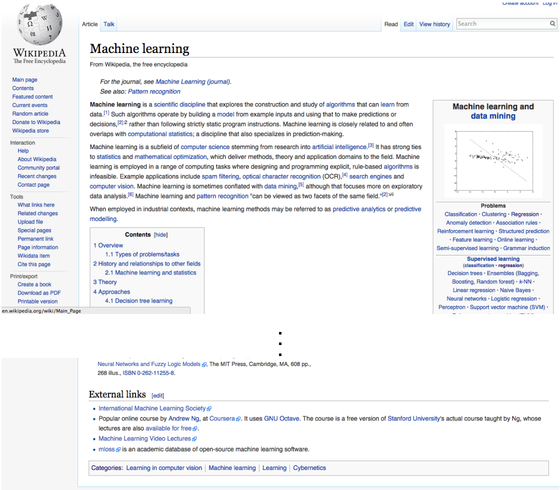
Ad-word Recommendation

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

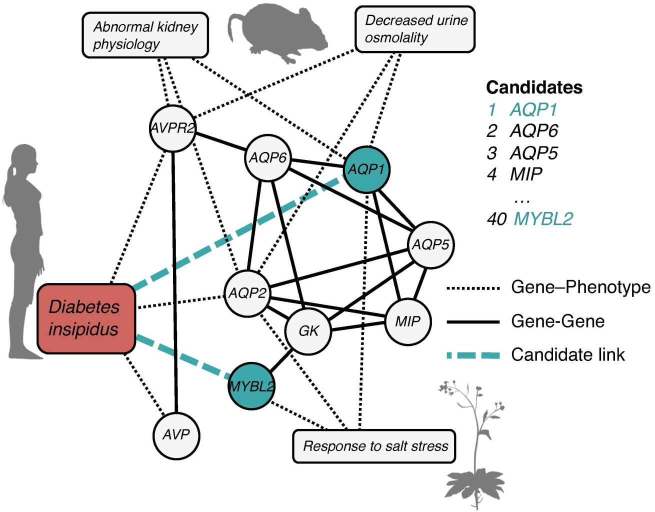
Modern Prediction Problems in Machine Learning

Wikipedia Tag Recommendation

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



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*

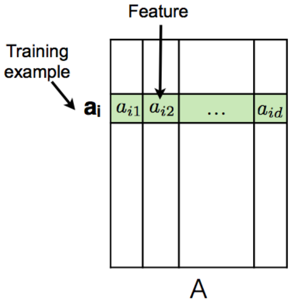
Prediction with Multiple Targets

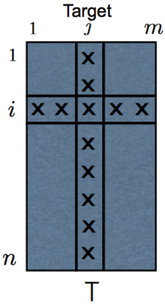
Applications

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

Prediction with Multiple Targets

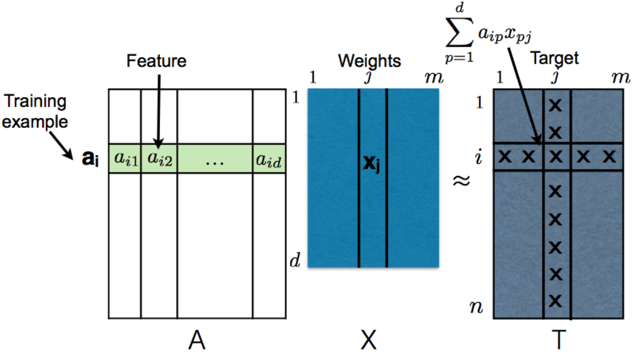
- Input data $\mathbf{a}_i$ is associated with m targets, $\mathbf{t}_i = (t_i^{(1)}, t_i^{(2)}, \ldots, t_i^{(m)})$





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:

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

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:

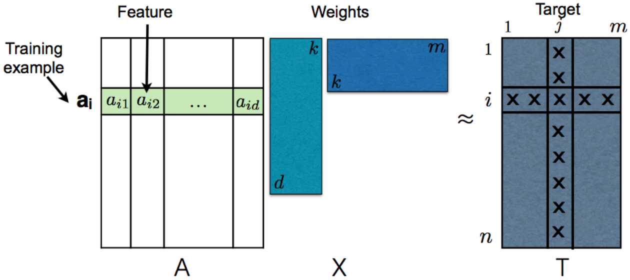
$$V_A \Sigma_A^{-1} U_A^\top T = \arg\min_X \; \|T - AX\|_F^2$$

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:

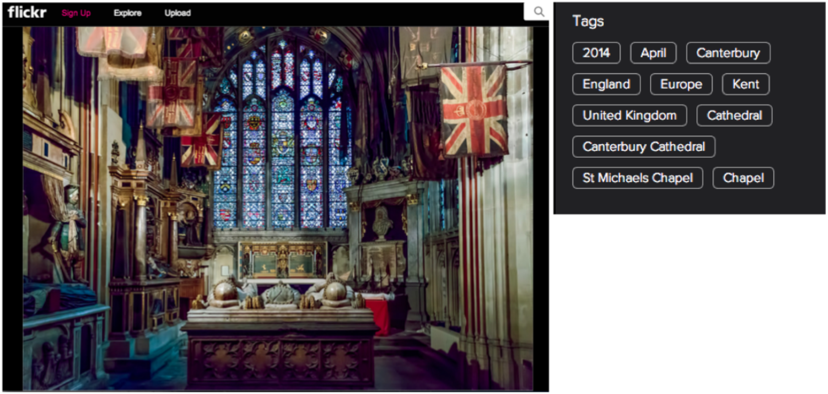
$$X^* = \min_{X: rank(X) \leq k} \| T - AX \|_F^2$$
$$= \begin{cases} V_A \Sigma_A^{-1} U_A^\top T_k & \text{if } A \text{ is full row rank,} \\ V_A \Sigma_A^{-1} M_k & \text{otherwise,} \end{cases}$$

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

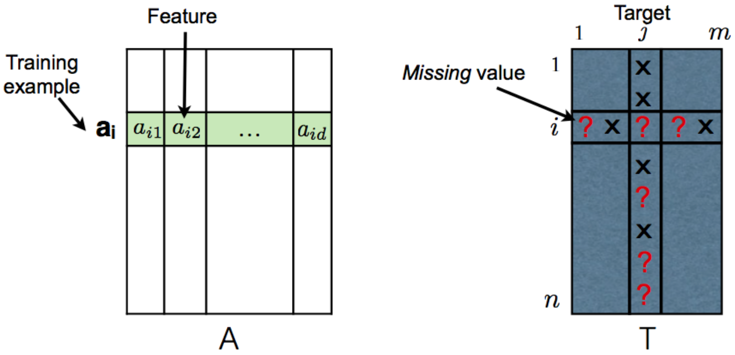


Modern Prediction Problems in Machine Learning

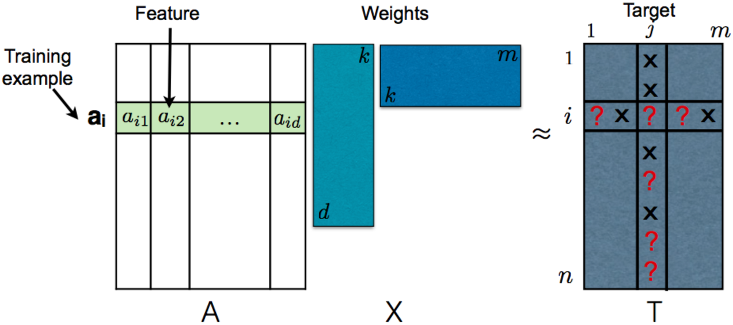
Ad-word Recommendation

- 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

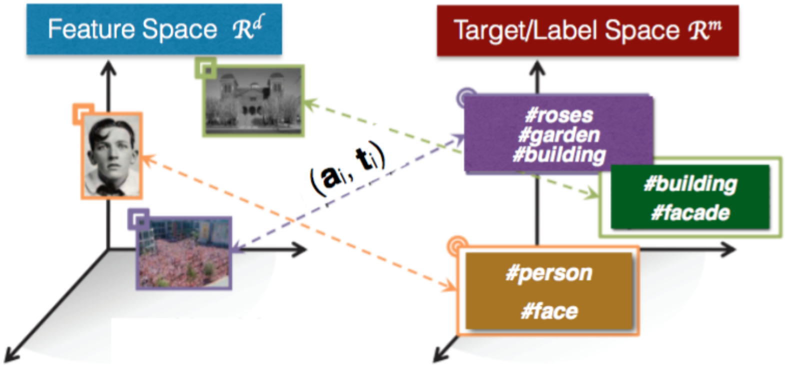


Multi-target Prediction with Missing Values



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

Canonical Correlation Analysis

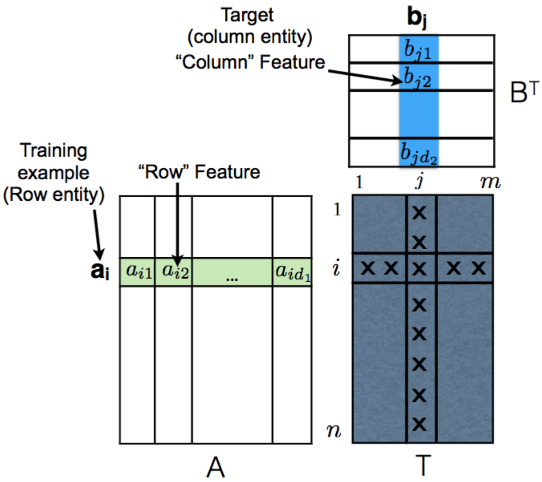


Bilinear Prediction

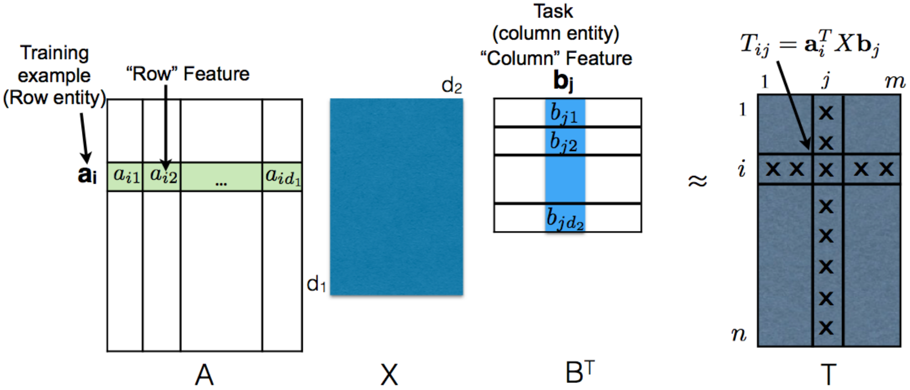
Bilinear Prediction

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

Bilinear Prediction

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

- Corresponding regression problem has a closed-form solution:

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

where $A = U_A \Sigma_A V_A^\top$, $B = U_B \Sigma_B V_B^\top$ 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:

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

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

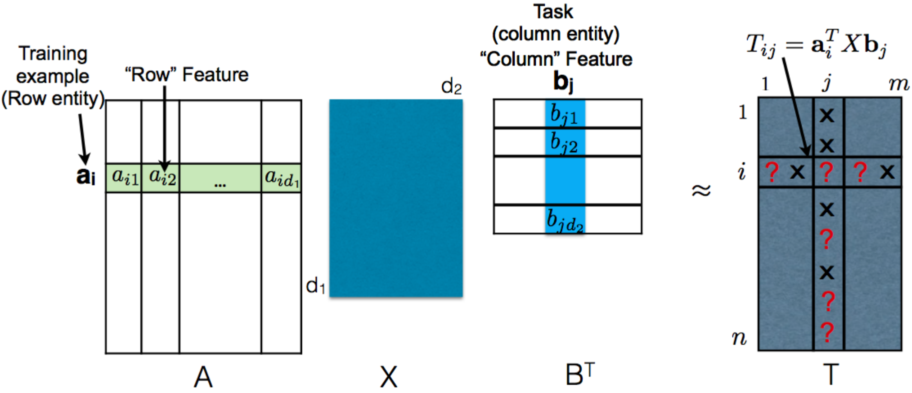
Modern Challenges in Multi-Target Prediction

- Millions of targets
- Correlations among targets
- Missing values

Modern Challenges in Multi-Target Prediction

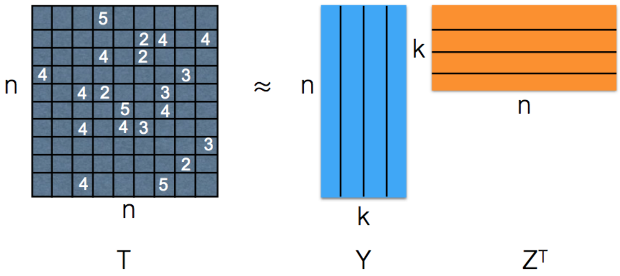
- Millions of targets (Scalable)
- Correlations among targets (Low-rank)
- Missing values (Inductive Matrix Completion)

Bilinear Prediction with Missing Values



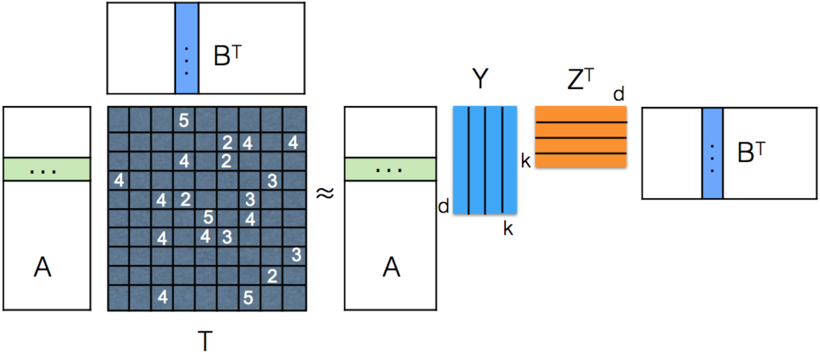
Matrix Completion

- 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:

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

- 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 Ω 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

$$\textbf{C1.} \ \max_{i \in [n]} \|\mathbf{a}_i\|_2^2 \leq \frac{\mu d}{n}, \ \max_{j \in [n]} \|\mathbf{b}_j\|_2^2 \leq \frac{\mu d}{n}$$

$$\textbf{C2.} \ \max_{i \in [n]} \|U_*^T \mathbf{a}_i\|_2^2 \leq \frac{\mu_0 k}{n}, \ \max_{j \in [n]} \|V_*^T \mathbf{b}_j\|_2^2 \leq \frac{\mu_0 k}{n}$$

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: $O(|\Omega|k^2 d \times (nnz(A) + nnz(B))/n + k^3 d^3)$.
2. Vanilla conjugate gradient: $O(|\Omega|k \times (nnz(A) + nnz(B))/n)$ 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 $O((nnz(A) + nnz(B) + |\Omega|) \times k)$ per iteration.

Algorithm 2: Alternating Least Squares

Theorem (Convergence Guarantees for ALS)

Let X_* be a rank-k matrix with condition number β, and $T = AX_*B^T$. Assume A, B are orthogonal w.l.o.g. and satisfy the incoherence conditions. Then if Ω 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:

$$\textbf{C1.} \ \max_{i\in[n]} \|\mathbf{a}_i\|_2^2 \leq \frac{\mu d}{n}, \ \max_{j\in[n]} \|b_j\|_2^2 \leq \frac{\mu d}{n}$$

$$\textbf{C2'.} \ \max_{i\in[n]} \|Y_h^T \mathbf{a}_i\|_2^2 \leq \frac{\mu_0 k}{n}, \ \max_{j\in[n]} \|Z_h^T b_j\|_2^2 \leq \frac{\mu_0 k}{n},$$

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

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

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 β is the condition number of X

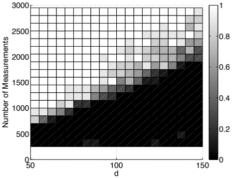
- In most cases, $n \gg d$
- Incoherence conditions on A, B are required
  - Satisfied e.g. when A, B are Gaussian (no assumption on X needed)

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

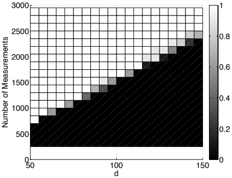
M. Hardt. *Understanding alternating minimization for matrix completion.* Foundations of Computer Science (FOCS), IEEE 55th Annual Symposium, pp. 651-660 (2014).

Inductive Matrix Completion: Sample Complexity Results

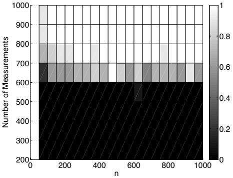
- All matrices are sampled from Gaussian random distribution.
- Left two figures: fix $k = 5$, $n = 1000$ and change d.
- Right two figures: fix $k = 5$, $d = 50$ and change n.
- Darkness of the shading is proportional to the number of failures (repeated 10 times).



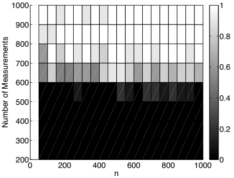
$|\Omega|$ vs. d (ALS)



$|\Omega|$ vs. d (Nuclear)



$|\Omega|$ vs. n (ALS)



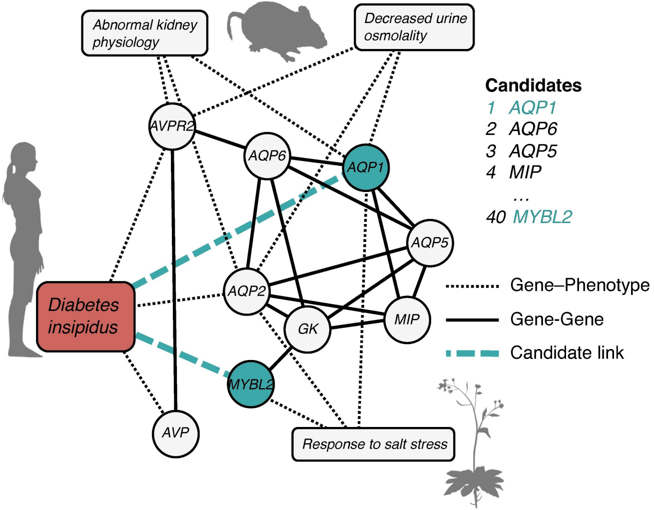
$|\Omega|$ vs. n (Nuclear)

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

Positive-Unlabeled Learning

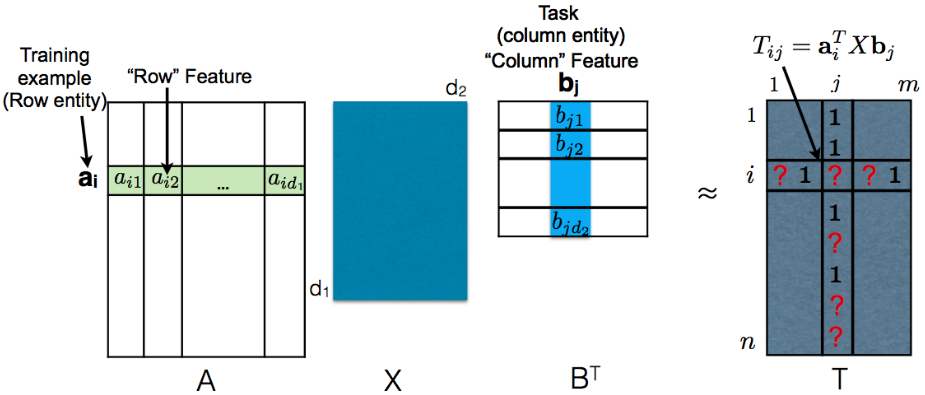
Modern Prediction Problems in Machine Learning

Predicting causal disease genes



Bilinear Prediction: PU Learning

In many applications, only "positive" labels are observed



PU Learning

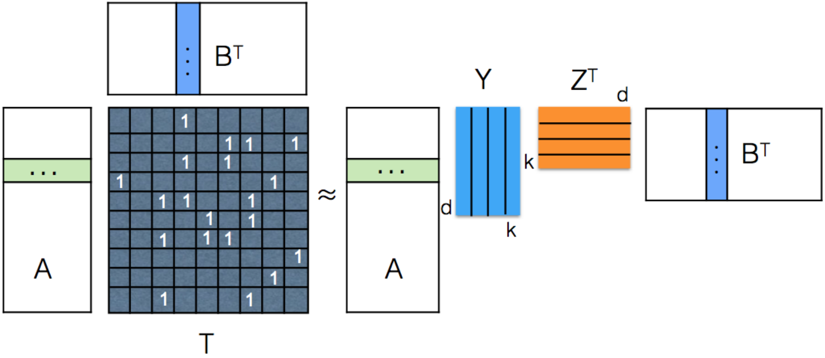
Learning Task	"Positives"	"Negatives"	"Unlabeled"
Supervised	✓	✓	
Semi-supervised	✓	✓	✓
Positive-Unlabeled (PU)	✓		✓
Unsupervised			✓

- No observations of the "negative" class available

$(X, Y) \sim D$ Training data

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

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

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

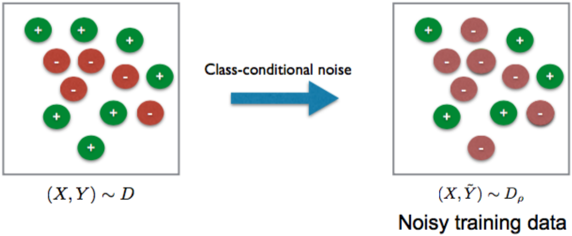
V. Sindhwani, S. S. Bucak, J. Hu, A. Mojsilovic. *One-class matrix completion with low-density factorizations*. ICDM, pp. 1055-1060. 2010.

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

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

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

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

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

- Rationale:
  (a) Fix $\alpha = (1+\rho)/2$ and define $\widehat{T}_{ij} = I\left[(A\widehat{X}B^T)_{ij} > 0.5\right]$
  (b) The above problem is equivalent to:

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

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

  (c) Minimizing ℓ_α loss is equivalent to minimizing the true error, i.e.

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

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 $\widehat{T}_{ij} = I\left[(A\widehat{X}B^T)_{ij} > 0.5\right]$, $\mathcal{A} = \max_i \|\mathbf{a}_i\|$ and $\mathcal{B} = \max_i \|\mathbf{b}_i\|$. If $\alpha = \frac{1+\rho}{2}$, then with probability at least $1-\delta$,

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

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).

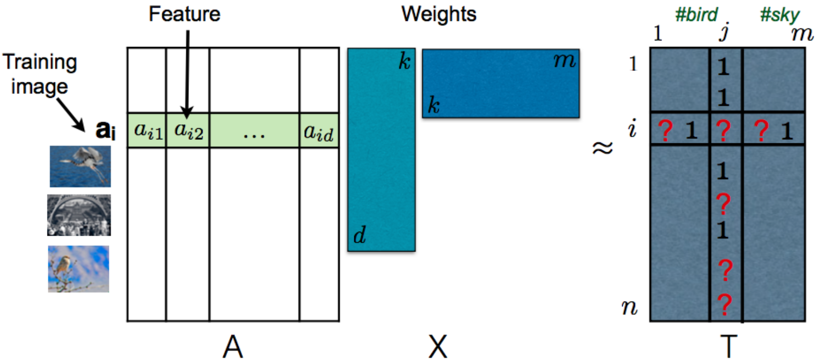
Experimental Results

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)	**574**	**20.71**	**15.96**	**0.7741**
WSABIE	4,705	14.58	11.37	0.7658

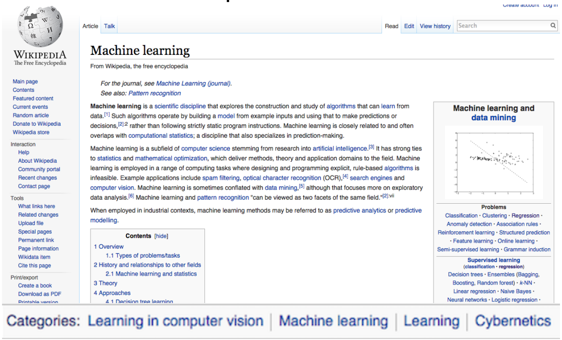
- Low-rank Model with $k = 100$:

	time(s)	prec@1	prec@3	AUC
LEML(ALS)	**1,097**	**20.76**	**16.00**	**0.7718**
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



- 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 $k = 250$:

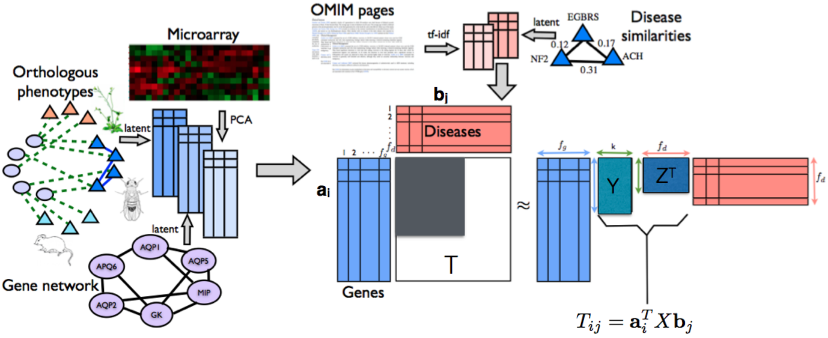
	time(s)	prec@1	prec@3	AUC
LEML(ALS)	**9,932**	**19.56**	14.43	**0.9086**
WSABIE	79,086	18.91	**14.65**	0.9020

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

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

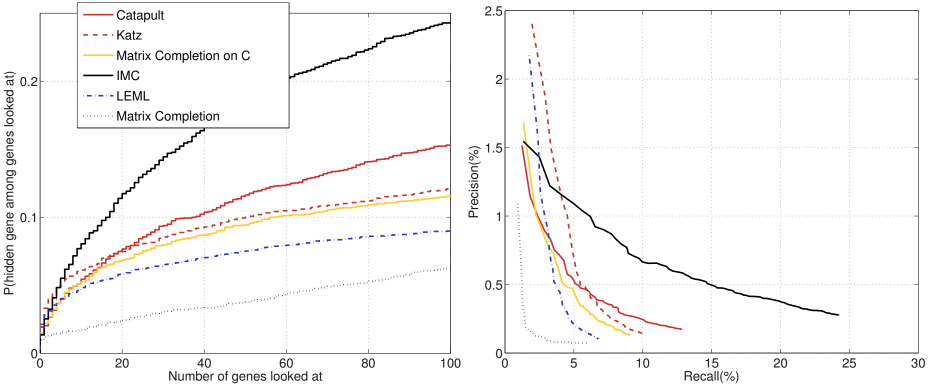
H. F. Yu, P. Jain, P. Kar, and I. S. Dhillon. *Large-scale Multi-label Learning with Missing Labels*. In Proceedings of The 31st International Conference on Machine Learning, pp. 593-601 (2014).

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).

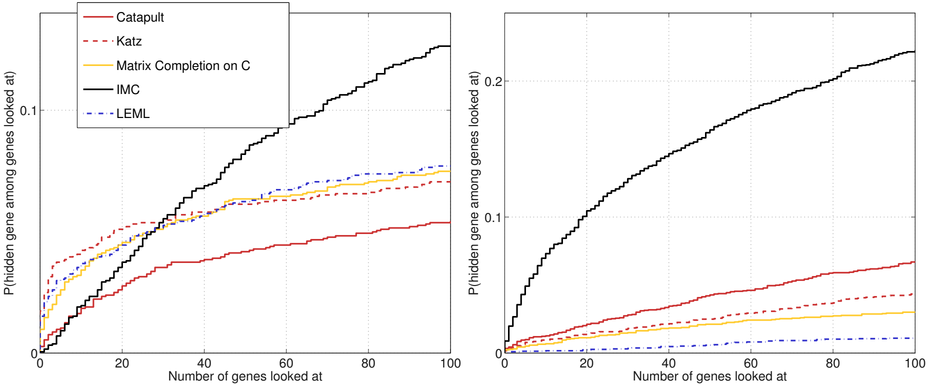
PU Inductive Matrix Completion: Gene-Disease Prediction



Predicting gene-disease associations in the OMIM data set (`www.omim.org`).

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).