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

![Graph showing stock prices]

Predicting risk factors in healthcare

![Cartoon showing risk factor comparison]
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 $a$, i.e. $t \approx \hat{t} = a^T x$. 

![Diagram of linear regression](image)
Linear Regression: Least Squares

- Choose $\mathbf{x}$ that minimizes

$$J_x = \frac{1}{2} \sum_{i=1}^{n} (a_i^T \mathbf{x} - t_i)^2$$

- 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: \[ J_x = \frac{1}{2} \sum_{i=1}^{n} (a_i^T x - t_i)^2 + \lambda \|x\|_2^2. \]
- Lasso: \[ J_x = \frac{1}{2} \sum_{i=1}^{n} (a_i^T x - t_i)^2 + \lambda \|x\|_1. \]

Classification:
- Linear Support Vector Machines

\[ J_x = \frac{1}{2} \sum_{i=1}^{n} \max(0, 1 - t_i a_i^T x) + \lambda \|x\|_2^2. \]

- Logistic Regression

\[ J_x = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \exp(-t_i a_i^T x)) + \lambda \|x\|_2^2. \]
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

- Laura Yoga Studio: (646) 702-4596
  - www.lauarayoga.com
  - Great for beginners. Get the first 3 classes free! Call now.

- Youth Yoga Classes
  - www.yogakids.com
  - Yoga for all ages! We offer modern facilities and reasonable rates
  - Yoga Kids Inc. – 610 McKerzie Boul. Denver, CO

- Yoga Accessories
  - www.yogaaaccessories.com
  - Experts or beginners, we have everything you need for yoga.

- Yoga Yoga Denver
  - 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.yogabasicguide.com

- Hot Yoga Classes
  - www.yogabears.com/hotyoga
  - Dynamic, fun and cost effective!
  - Special: 10 classes for $100

- Yoga for beginners
  - www.vinashiyoga.com
  - Burn calories and find peace.
  - Small classes. First week free!
  - (354) 555-0111 - Directions

- Lilac Yoga Studio
  - www.lilacyogadenver.com
  - Try our popular yoga sessions
  - Limited time $100 for 10!
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

**Candidates**
1. AQP1
2. AQP6
3. AQP5
4. MIP
...
40. MYBL2

*Gene–Phenotype*
*Gene–Gene*
*Candidate link*
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 $a_i$ is associated with $m$ targets, $t_i = (t_i^{(1)}, t_i^{(2)}, \ldots, t_i^{(m)})$
Multi-target Linear Prediction

- Basic model: Treat targets independently
- Estimate regression coefficients $x_j$ for each target $j$
Multi-target Linear Prediction

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

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

- 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 $t^{(j)}$ are independent
- Linear predictive model: $t_i \approx a_i^T X$

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)
Exploit correlations between targets $T$, where $T \approx AX$

**Reduced-Rank Regression** [A.J. Izenman, 1974] — model the coefficient matrix $X$ as *low-rank*

---

Multi-target Linear Prediction: Low-rank Model

- \( X \) is rank-\( k \)
- Linear predictive model: \( t_i \approx a_i^T X \)
Multi-target Linear Prediction: Low-rank Model

- \( X \) is rank-\( k \)
- Linear predictive model: \( t_i \approx a_i^T X \)

Low-rank multi-target regression problem has a closed-form solution:

\[
X^* = \min_{X: \text{rank}(X) \leq k} \| T - AX \|_F^2
\]

\[
= \begin{cases} 
V_A \Sigma_A^{-1} U_A^T T_k & \text{if } A \text{ is full row rank,} \\
V_A \Sigma_A^{-1} M_k & \text{otherwise,}
\end{cases}
\]

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.

![Image of a church interior with a list of tags on the right side.](image-url)
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

Inderjit S. Dhillon  Dept of Computer Science UT Austin

Low-Rank Bilinear Prediction
Multi-target Prediction with Missing Values

Low-rank model: \( \mathbf{t}_i = \mathbf{a}_i^T \mathbf{X} \) where \( \mathbf{X} \) is low-rank.
Canonical Correlation Analysis

Feature Space $\mathcal{R}^d$

Target/Label Space $\mathcal{R}^m$

$(a_i, t_i)$

#roses
#garden
#building

#building
#facade

#person
#face
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

- Target (column entity) “Column” Feature
- “Row” Feature
- Training example (Row entity)

\[ A \]

\[ \begin{bmatrix} a_{i1} & a_{i2} & \ldots & a_{id_1} \end{bmatrix} \]

\[ \begin{bmatrix} b_{j1} \\ b_{j2} \\ b_{jd_2} \end{bmatrix} \]

\[ B^T \]

\[ \begin{bmatrix} x & x \\ x & x \\ \vdots & \vdots \\ x & x \end{bmatrix} \]
Bilinear Prediction

Task (column entity)
"Column" Feature

\[ T_{ij} = a_i^T X b_j \]
Bilinear Prediction

Bilinear predictive model: \( T_{ij} \approx a_i^T X b_j \)
Bilinear Prediction

- Bilinear predictive model: $T_{ij} \approx a_i^T X b_j$

- 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: $T_{ij} \approx a_i^T X b_j$
Bilinear Prediction: Low-rank Model

- $X$ is rank-$k$
- Bilinear predictive model: $T_{ij} \approx a_i^T X b_j$

Corresponding regression problem has a closed-form solution:

$$X^* = \min_{X: \text{rank}(X) \leq k} \| T - AXB^\top \|_F^2$$

$$= \begin{cases} 
V_A \Sigma_A^{-1} U_A^\top T_k U_B \Sigma_B^{-1} V_B^\top & \text{if } A, B \text{ are full row rank,} \\
V_A \Sigma_A^{-1} M_k \Sigma_B^{-1} V_B^\top & \text{otherwise,}
\end{cases}$$

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

Task (column entity) 
"Column" Feature

\[ T_{ij} = a_i^T X b_j \]
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 $O(kn)$)?

\[
\text{Low-Rank Bilinear Prediction}
\]
Algorithm 1: Convex Relaxation

Nuclear-norm Minimization:

\[
\min \|X\|_* \\
\text{s.t. } a_i^T X b_j = T_{ij}, (i, j) \in \Omega
\]

- 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
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

\[\begin{align*}
\text{C1.} & \quad \max_{i \in [n]} \|a_i\|_2^2 \leq \frac{\mu d}{n}, \quad \max_{j \in [n]} \|b_j\|_2^2 \leq \frac{\mu d}{n} \\
\text{C2.} & \quad \max_{i \in [n]} \|U_*^T a_i\|_2^2 \leq \frac{\mu_0 k}{n}, \quad \max_{j \in [n]} \|V_*^T b_j\|_2^2 \leq \frac{\mu_0 k}{n}
\end{align*}\]
Algorithm 2: Alternating Least Squares

Alternating Least Squares (ALS):

\[
\begin{align*}
\min_{Y \in \mathbb{R}^{d_1 \times k}, Z \in \mathbb{R}^{d_2 \times k}} \sum_{(i,j) \in \Omega} (a_i^T Y Z^T b_j - T_{ij})^2
\end{align*}
\]

- 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 $Z_{h+1}$:

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

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

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

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

where $D$ is a sparse matrix with $D_{ji} = \tilde{a}_j^T Z^T b_j$, $(i,j) \in \Omega$, 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:

1. $\max_{i \in [n]} \|a_i\|_2^2 \leq \frac{\mu d}{n}$, $\max_{j \in [n]} \|b_j\|_2^2 \leq \frac{\mu d}{n}$

2. $\max_{i \in [n]} \|Y_h^T a_i\|_2^2 \leq \frac{\mu_0 k}{n}$, $\max_{j \in [n]} \|Z_h^T b_j\|_2^2 \leq \frac{\mu_0 k}{n}$

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 $k = 1$:

$$
\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (a_i^T y z^T b_j - T_{ij})^2
$$
Algorithm 2: Alternating Least Squares

- Proof sketch for rank-1 ALS

\[
\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (a_i^T y z^T b_j - T_{ij})^2
\]

(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.

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

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

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

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

\[
\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (a_i^T y z^T b_j - T_{ij})^2
\]

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

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

where \( N = \sum_{(i,j) \in \Omega} b_j a_i^T y_h y_h^T a_i b_j \), \( \tilde{N} = \sum_{(i,j) \in \Omega} b_j a_i^T y_h y_*^T a_i b_j \).

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

\[
\|g\|_2 \leq \frac{1}{99} \sqrt{1 - (y_h^T y_*)^2}
\]
Algorithm 2: Alternating Least Squares

- Proof sketch for rank-1 ALS

\[
\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (a_i^T y z^T b_j - T_{ij})^2
\]

(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

\[
\min_{y \in \mathbb{R}^{d_1}, z \in \mathbb{R}^{d_2}} \sum_{(i,j) \in \Omega} (a_i^T y z^T b_j - T_{ij})^2
\]

(e) Given \textbf{C1} and \textbf{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.
Sample complexity of Inductive Matrix Completion (IMC) and Matrix Completion (MC).

<table>
<thead>
<tr>
<th>methods</th>
<th>IMC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nuclear-norm</td>
<td>$O(kd \log n \log d)$</td>
<td>$kn \log^2 n$ (Recht, 2011)</td>
</tr>
<tr>
<td>ALS</td>
<td>$O(k^4 \beta^2 d \log d)$</td>
<td>$k^3 \beta^2 n \log n$ (Hardt, 2014)</td>
</tr>
</tbody>
</table>

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)


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
Predicting causal disease genes

**Diabetes insipidus**

**Candidates**
1. AQP1
2. AQP6
3. AQP5
4. MIP

**Response to salt stress**

**Abnormal kidney physiology**

**Decreased urine osmolality**

**Gene–Phenotype**

**Gene-Gene**

**Candidate link**
Bilinear Prediction: PU Learning

In many applications, only “positive” labels are observed.

\[
T_{ij} = a_i^T X b_j
\]

\(A\) is the training example (row entity), \(a_i\) is the row feature with \(d_1\) features, and \(a_{i1}, a_{i2}, ..., a_{id_1}\) are features. \(X\) is another matrix with \(d_2\) columns. \(B^T\) is a matrix with \(m\) columns. \(b_j\) is the column entity feature with \(d_2\) features, and \(b_{j1}, b_{j2}, ..., b_{jd_2}\) are features. \(T\) is the target matrix with \(n \times m\) entries, and \(T_{ij}\) is the element at row \(i\) and column \(j\) with \(1\) if the label is observed, or \(?\) if the label is not observed.
## PU Learning

<table>
<thead>
<tr>
<th>Learning Task</th>
<th>“Positives”</th>
<th>“Negatives”</th>
<th>“Unlabeled”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Positive-Unlabeled (PU)</td>
<td>✔</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Unsupervised</td>
<td></td>
<td></td>
<td>✔</td>
</tr>
</tbody>
</table>

- 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")
Inductive Matrix Completion:

\[
\min_{X: \|X\|_* \leq t} \sum_{(i,j) \in \Omega} (a_i^T X b_j - T_{ij})^2
\]

Commonly used PU strategy: Biased Matrix Completion

\[
\min_{X: \|X\|_* \leq t} \alpha \sum_{(i,j) \in \Omega} (a_i^T X b_j - T_{ij})^2 + (1 - \alpha) \sum_{(i,j) \notin \Omega} (a_i^T X b_j - 0)^2
\]

Typically, \(\alpha > 1 - \alpha\) (\(\alpha \approx 0.9\)).
PU Inductive Matrix Completion

- Inductive Matrix Completion:

\[
\min_{X : \|X\|_* \leq t} \sum_{(i,j) \in \Omega} (a_i^T X b_j - T_{ij})^2
\]

- Commonly used PU strategy: Biased Matrix Completion

\[
\min_{X : \|X\|_* \leq t} \alpha \sum_{(i,j) \in \Omega} (a_i^T X b_j - T_{ij})^2 + (1 - \alpha) \sum_{(i,j) \notin \Omega} (a_i^T X b_j - 0)^2
\]

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

- We can show guarantees for the biased formulation

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 \( \rho_{-1} = 0 \)

\( (X, Y) \sim D \)

Class-conditional noise

\( (X, \tilde{Y}) \sim D_\rho \)

Noisy training data

A deterministic PU learning model

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

\[
\begin{array}{cccc}
0.2 & 0.1 & 0 & 0.8 \\
0 & 0.6 & 0.1 & 0.9 \\
0 & 0 & 0.8 & 0.1 \\
0.9 & 0 & 0.2 & 0.1 \\
0 & 0.6 & 0 & 1
\end{array}
\]

\[
\begin{array}{cccc}
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1
\end{array}
\]
A deterministic PU learning model

\[ P(\tilde{T}_{ij} = 0 | T_{ij} = 1) = \rho \text{ and } P(\tilde{T}_{ij} = 0 | T_{ij} = 0) = 1. \]

- 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

\[ \hat{X} = \min_{X: \|X\|_* \leq t} \alpha \sum_{(i,j) \in \Omega} (a_i^T X b_j - 1)^2 + (1 - \alpha) \sum_{(i,j) \notin \Omega} (a_i^T X b_j - 0)^2 \]

Rationale:
(a) Fix \( \alpha = (1 + \rho)/2 \) and define \( \tilde{T}_{ij} = I[(A\hat{X}B^T)_{ij} > 0.5] \)
(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 $\hat{T}_{ij} = I[(A\hat{X}B^T)_{ij} > 0.5]$, $A = \max_i \|a_i\|$ and $B = \max_i \|b_i\|$. If $\alpha = \frac{1+\rho}{2}$, then with probability at least $1 - \delta$,

$$\frac{1}{n^2} \| T - \hat{T} \|_F^2 = O\left(\frac{\eta \sqrt{\log(2/\delta)}}{n(1 - \rho)} + \frac{\eta \ tAB \sqrt{\log2d}}{(1 - \rho)n^{3/2}}\right)$$

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

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

## Multi-target Prediction: Image Tag Recommendation

- **Low-rank Model with** \( k = 50 \):
  
<table>
<thead>
<tr>
<th></th>
<th>time(s)</th>
<th>prec@1</th>
<th>prec@3</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEML(ALS)</td>
<td>574</td>
<td>20.71</td>
<td>15.96</td>
<td>0.7741</td>
</tr>
<tr>
<td>WSABIE</td>
<td>4,705</td>
<td>14.58</td>
<td>11.37</td>
<td>0.7658</td>
</tr>
</tbody>
</table>

- **Low-rank Model with** \( k = 100 \):
  
<table>
<thead>
<tr>
<th></th>
<th>time(s)</th>
<th>prec@1</th>
<th>prec@3</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEML(ALS)</td>
<td>1,097</td>
<td>20.76</td>
<td>16.00</td>
<td>0.7718</td>
</tr>
<tr>
<td>WSABIE</td>
<td>6,880</td>
<td>12.46</td>
<td>10.21</td>
<td>0.7597</td>
</tr>
</tbody>
</table>

---

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 $k = 250$:

<table>
<thead>
<tr>
<th></th>
<th>time(s)</th>
<th>prec@1</th>
<th>prec@3</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEML(ALS)</td>
<td>9,932</td>
<td>19.56</td>
<td>14.43</td>
<td>0.9086</td>
</tr>
<tr>
<td>WSABIE</td>
<td>79,086</td>
<td>18.91</td>
<td>14.65</td>
<td>0.9020</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>time(s)</th>
<th>prec@1</th>
<th>prec@3</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEML(ALS)</td>
<td>18,072</td>
<td>22.83</td>
<td>17.30</td>
<td>0.9374</td>
</tr>
<tr>
<td>WSABIE</td>
<td>139,290</td>
<td>19.20</td>
<td>15.66</td>
<td>0.9058</td>
</tr>
</tbody>
</table>


Predicting genes for diseases with no training associations.

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


