Multi-Output Prediction: Theory and Practice

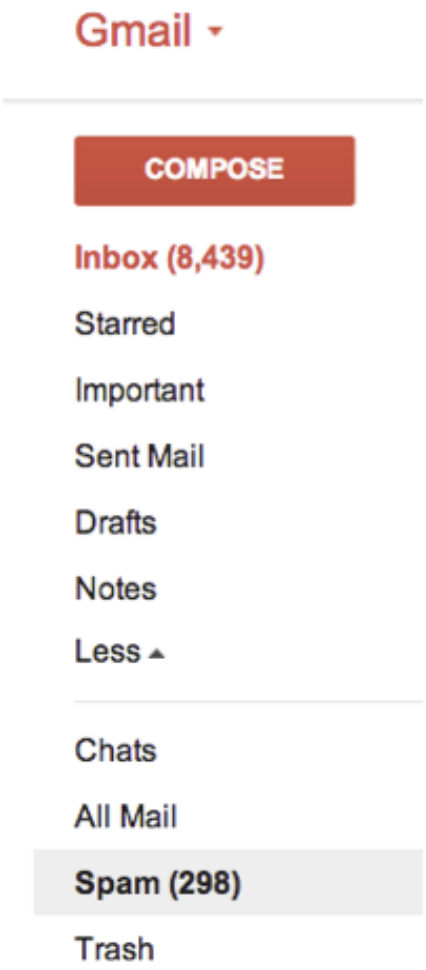
Inderjit S. Dhillon
UT Austin & Amazon

PAKDD 2020
Singapore
May 13, 2020

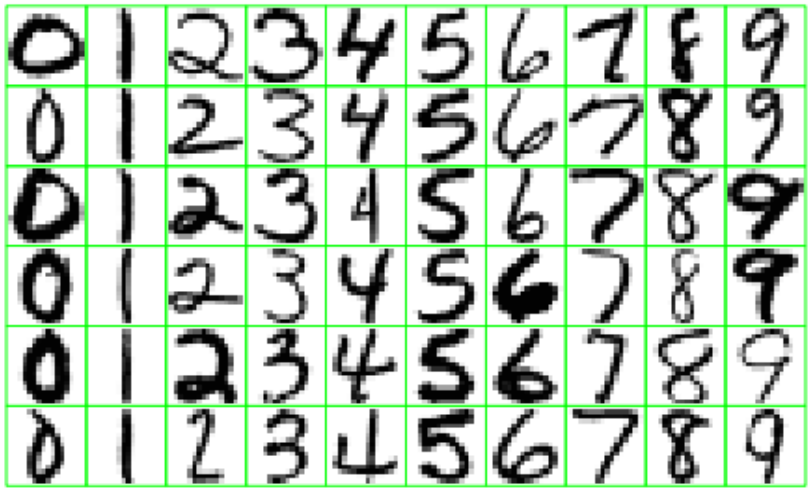
Outline

Modern Prediction Problems

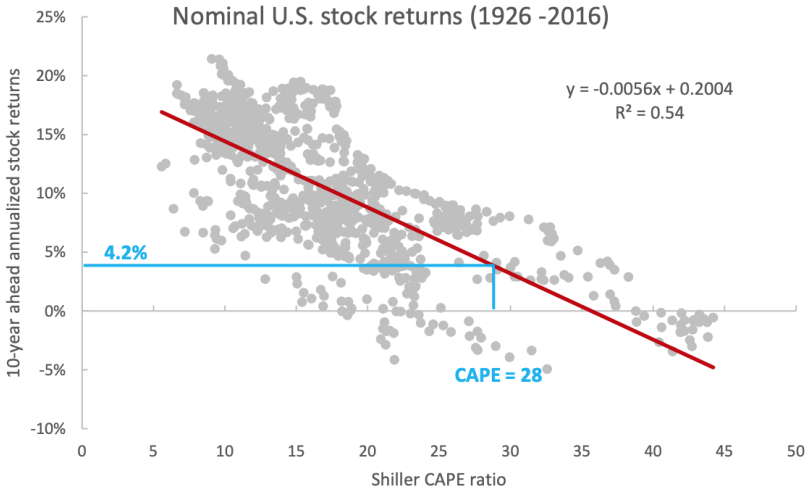
Spam detection



Character Recognition

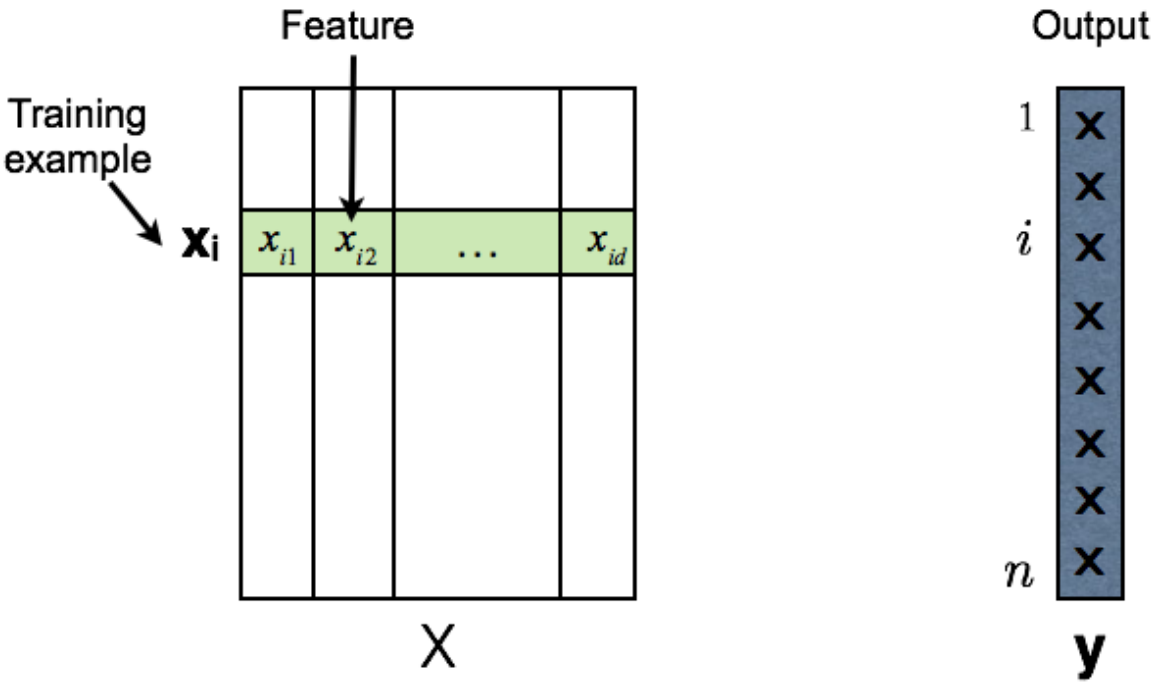


Regression Analysis



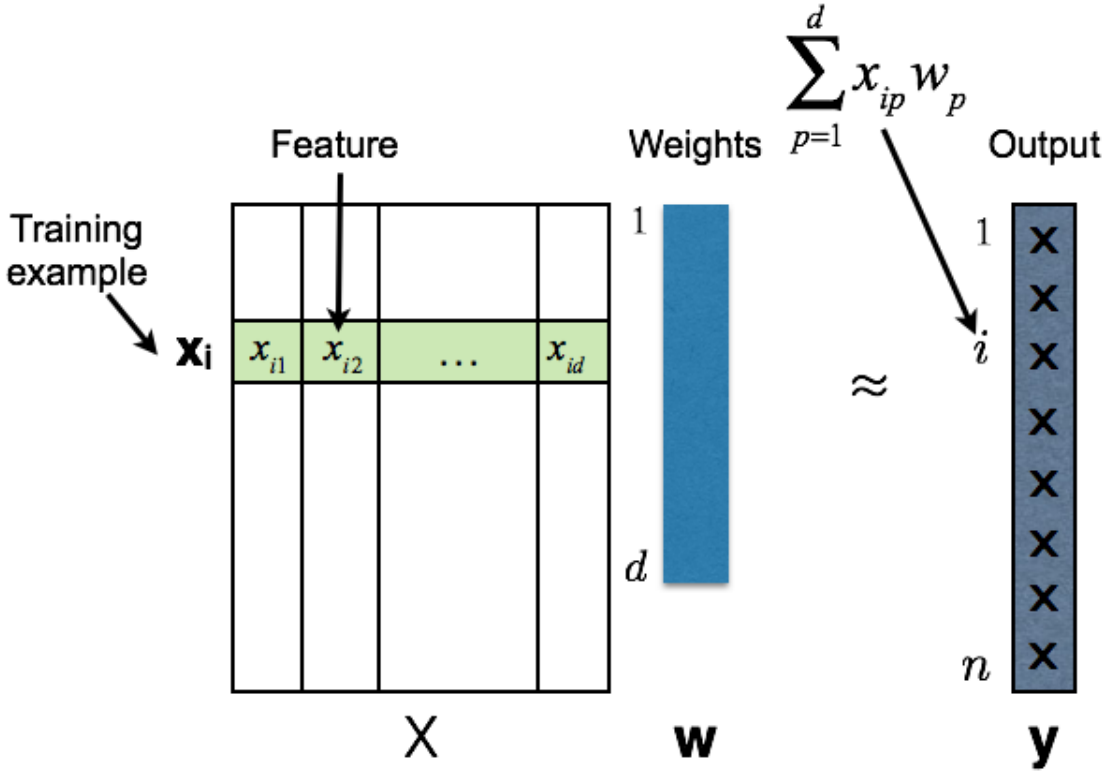
Single-Output Regression

- Real-valued response (output) **y**
- Predict response for given input data (features) **x**



Linear Regression

- Estimate output by a linear function of given data $\mathbf{x}$, i.e. $\mathbf{y} \approx \hat{\mathbf{y}} = \mathbf{x}^\top \mathbf{w}$.

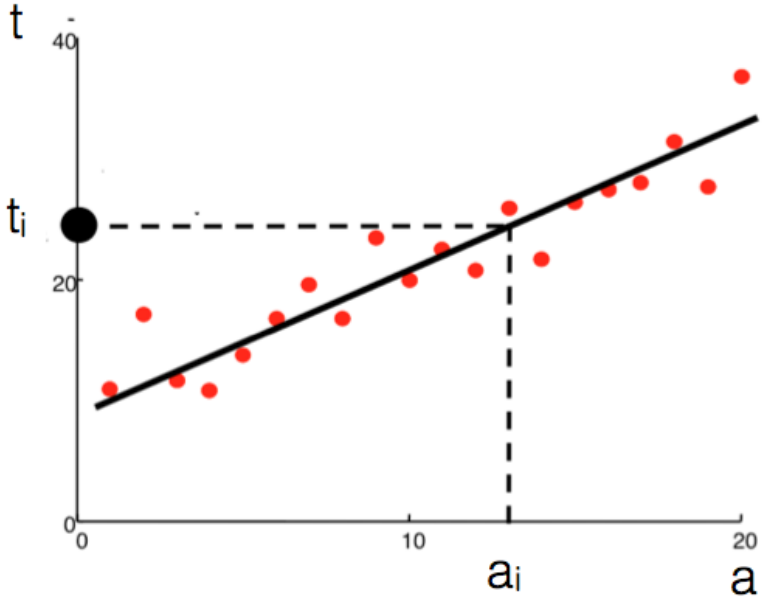


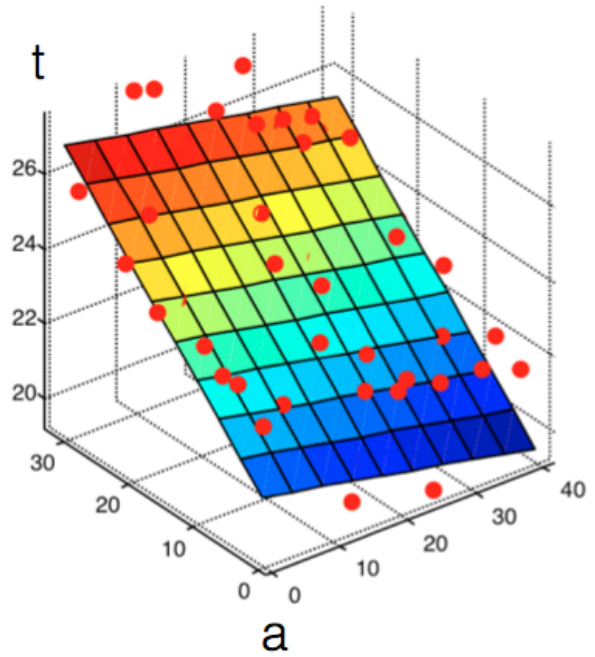
Linear Regression: Least Squares

- Choose $\mathbf{w}$ that minimizes

$$J_{\mathbf{w}} = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i^\top \mathbf{w} - y_i)^2$$

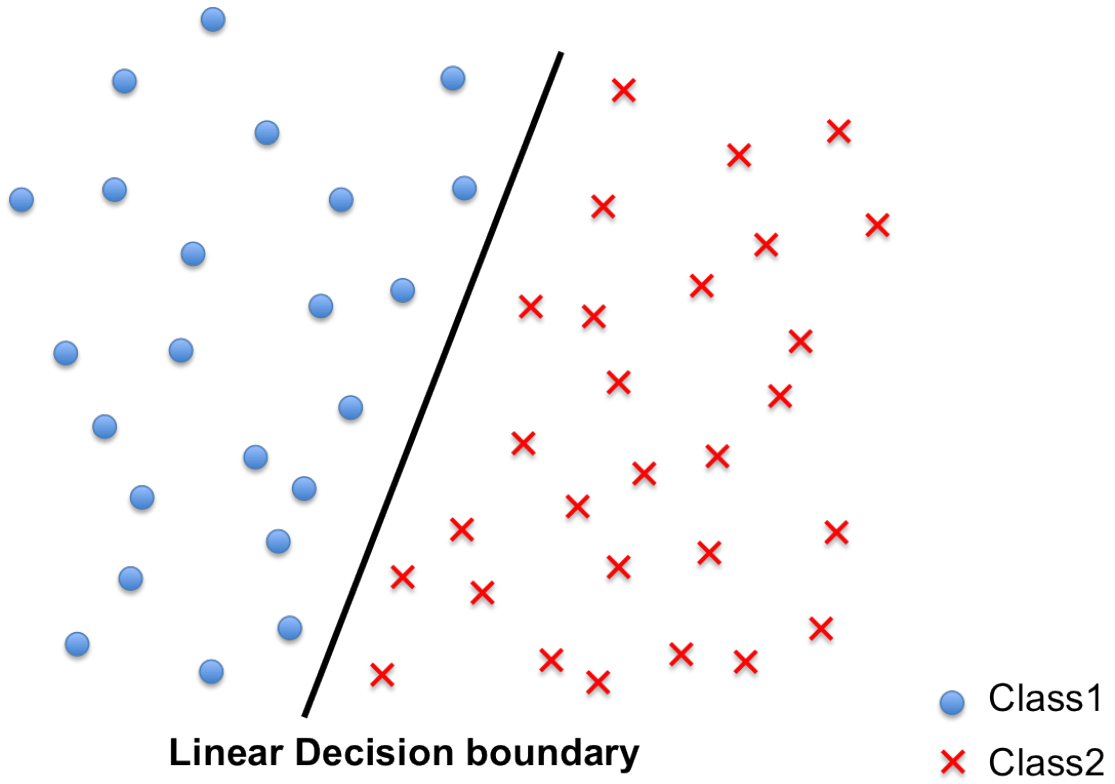
- Closed-form solution: $\mathbf{w}^* = (X^\top X)^{-1} X^\top \mathbf{y}$.





Binary Classification

- Categorical response (output) **y**
- Predict response for given input data (features) **x**
- Linear methods — decision boundary is a linear surface or hyperplane



Linear Methods for Prediction Problems

Regression:

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

Classification:

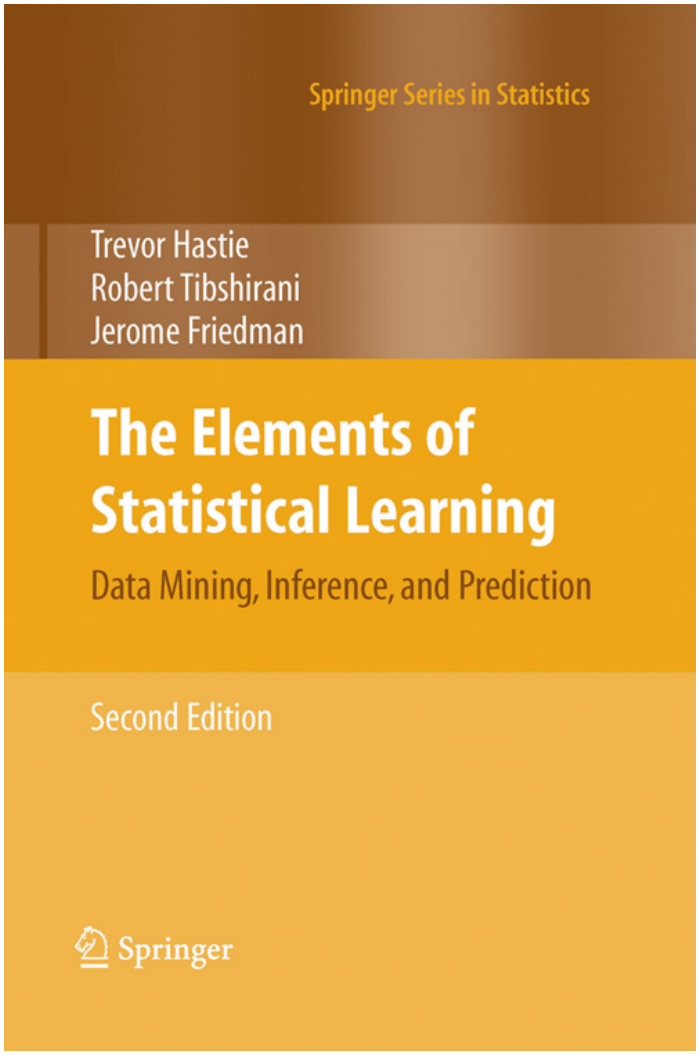
- Linear Support Vector Machines

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

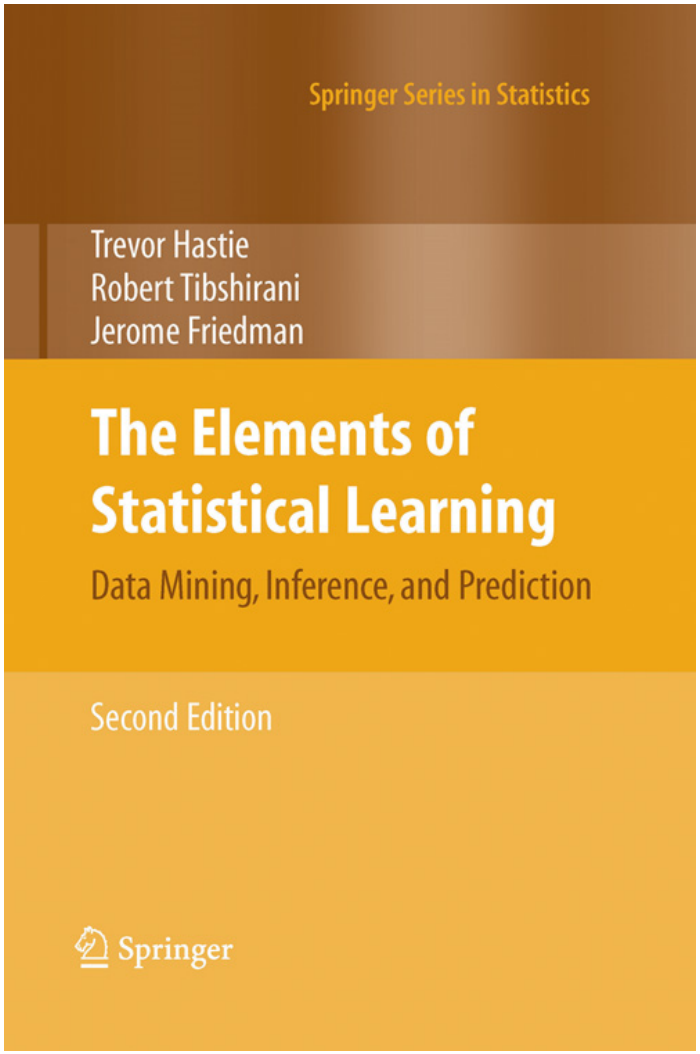
- Logistic Regression

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

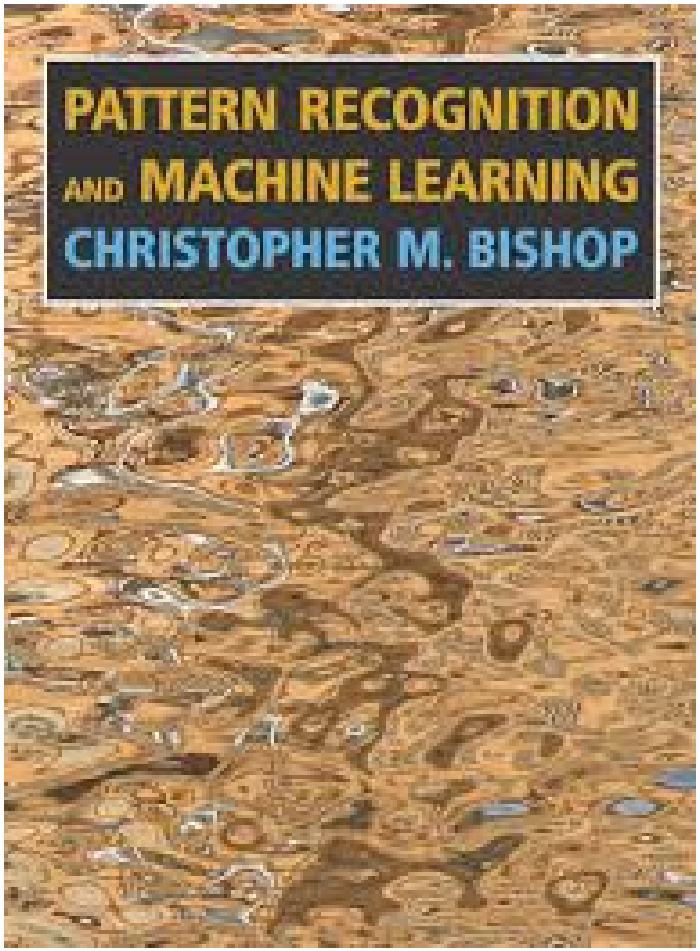
Linear Prediction



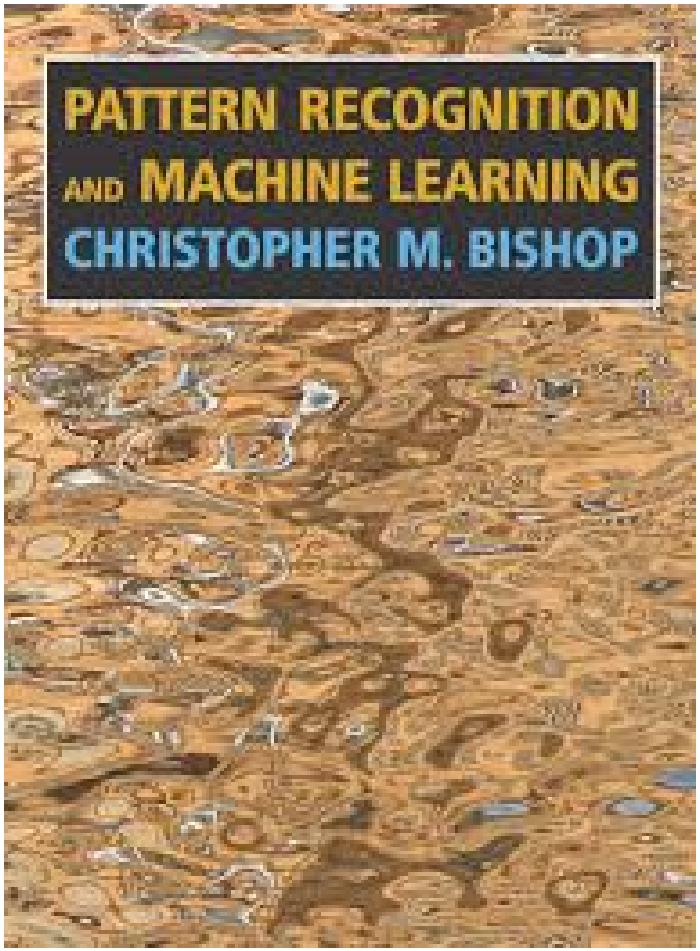
Linear Prediction



Linear Prediction



Linear Prediction



Multi-Output Prediction

Modern Prediction Problems

Wikipedia Tag Recommendation

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

Modern Prediction Problems

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

Recommending tags for images

Multi-Output Prediction: Modern Challenges

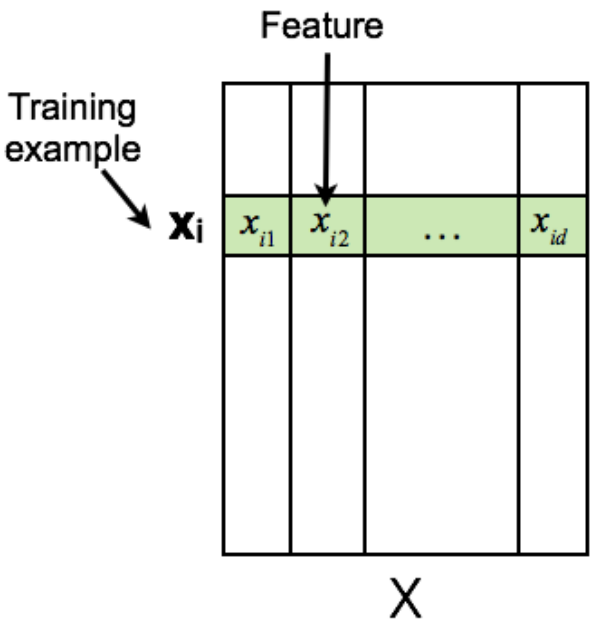
- Correlated outputs, and missing output values
- Outputs have features
- Positive-unlabeled (PU) Learning
- Scaling Up - Millions of Correlated Outputs

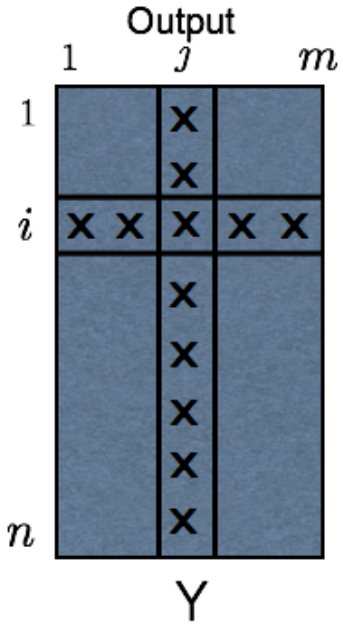
Multi-Output Prediction: Modern Challenges

- Correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features
- Positive-unlabeled (PU) Learning
- Scaling Up - Millions of Correlated Outputs

Prediction with Multiple Outputs

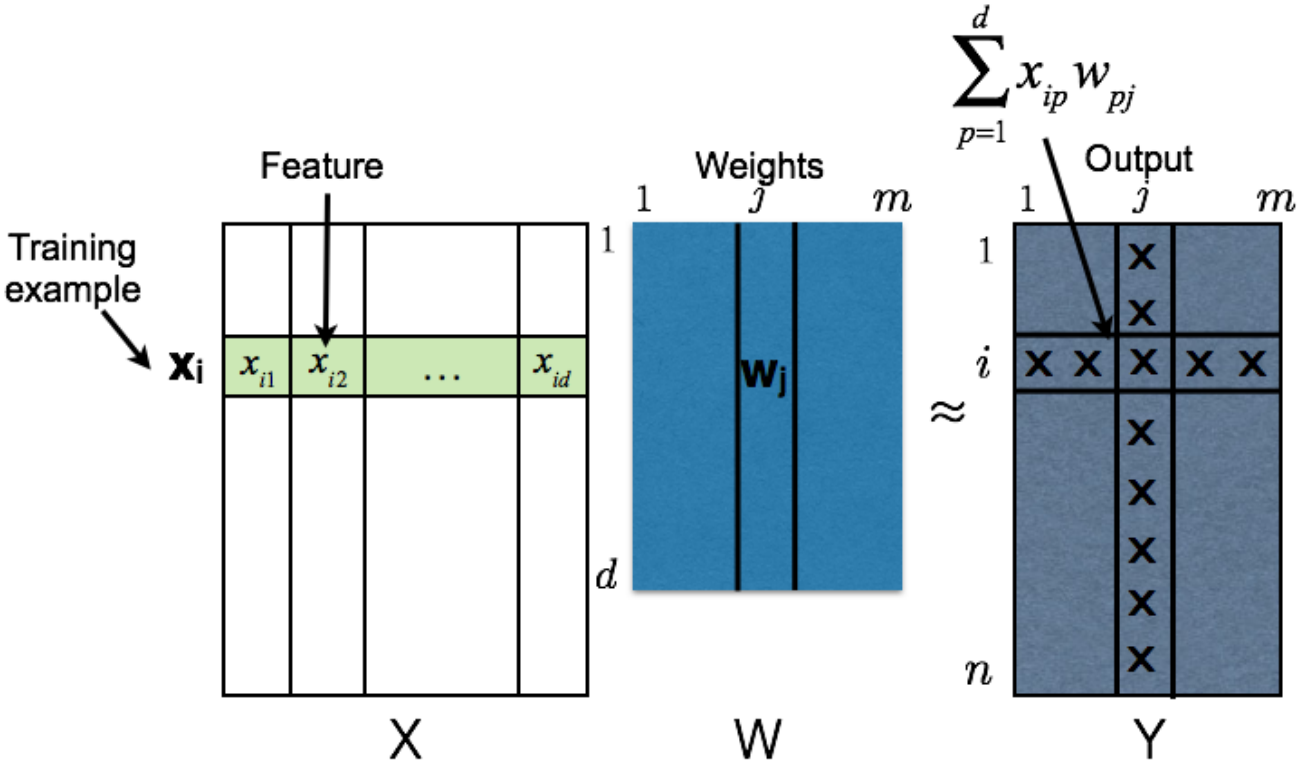
- Input data $\mathbf{x}_i$ is associated with m outputs, $\mathbf{y}_i = (y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(m)})$





Multi-Output Linear Prediction

- Basic model: Treat outputs independently
- Estimate regression coefficients $\mathbf{w}_j$ for each output j



Multi-Output Linear Prediction

- Assume outputs $\mathbf{y}^{(j)}$ are independent
- Linear predictive model: $\mathbf{y}_i \approx \mathbf{x}_i^\top W$
- Objective for multi-output regression:

$$\min_W \ \|Y - XW\|_F^2$$

- Closed-form solution:

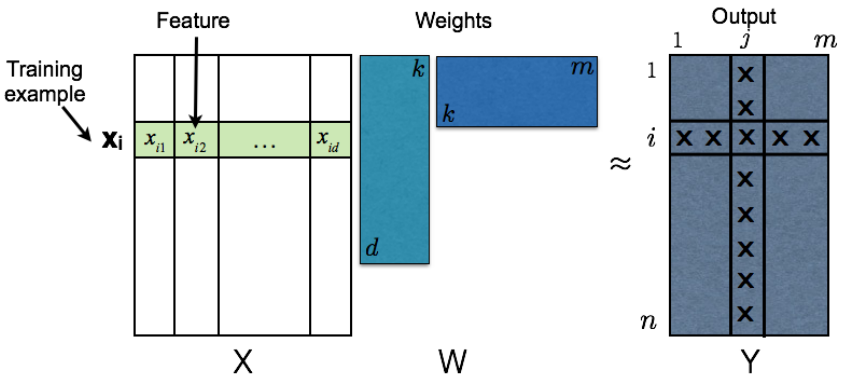
$$V\Sigma^{-1}U^\top Y = \arg\min_W \ \|Y - XW\|_F^2$$

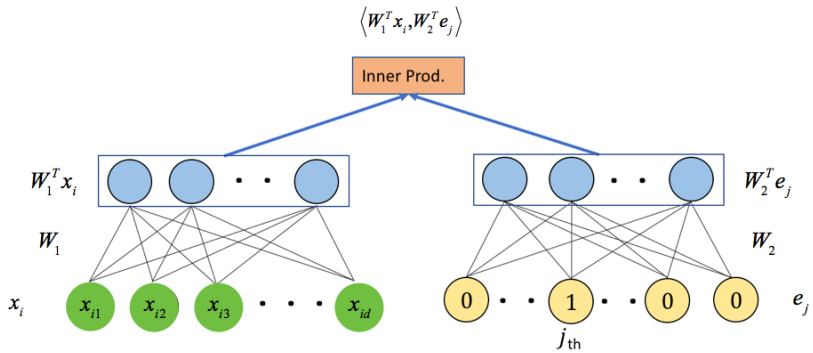
where $X = U\Sigma V^\top$ is the thin SVD of X

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

Multi-Output Linear Prediction: Low-rank Model

- Exploit correlations between outputs Y, where $Y \approx XW$
- Reduced-Rank Regression [A.J. Izenman, 1974] — model the coefficient matrix W as *low-rank*





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

Multi-Output Linear Prediction: Low-rank Model

- W is rank-k
- Linear predictive model: $\mathbf{y}_i \approx \mathbf{x}_i^\top W$
- Objective for low-rank multi-output regression:

$$\min_{W:rank(W)\leq k} \quad \|Y - XW\|_F^2$$

- Closed-form solution:

$$W^* = \arg \min_{W:rank(W)\leq k} \quad \|Y - XW\|_F^2$$

$$= \begin{cases} V\Sigma^{-1}U^\top Y_k & \text{if } X \text{ is full row rank,} \\ V\Sigma^{-1}M_k & \text{otherwise,} \end{cases}$$

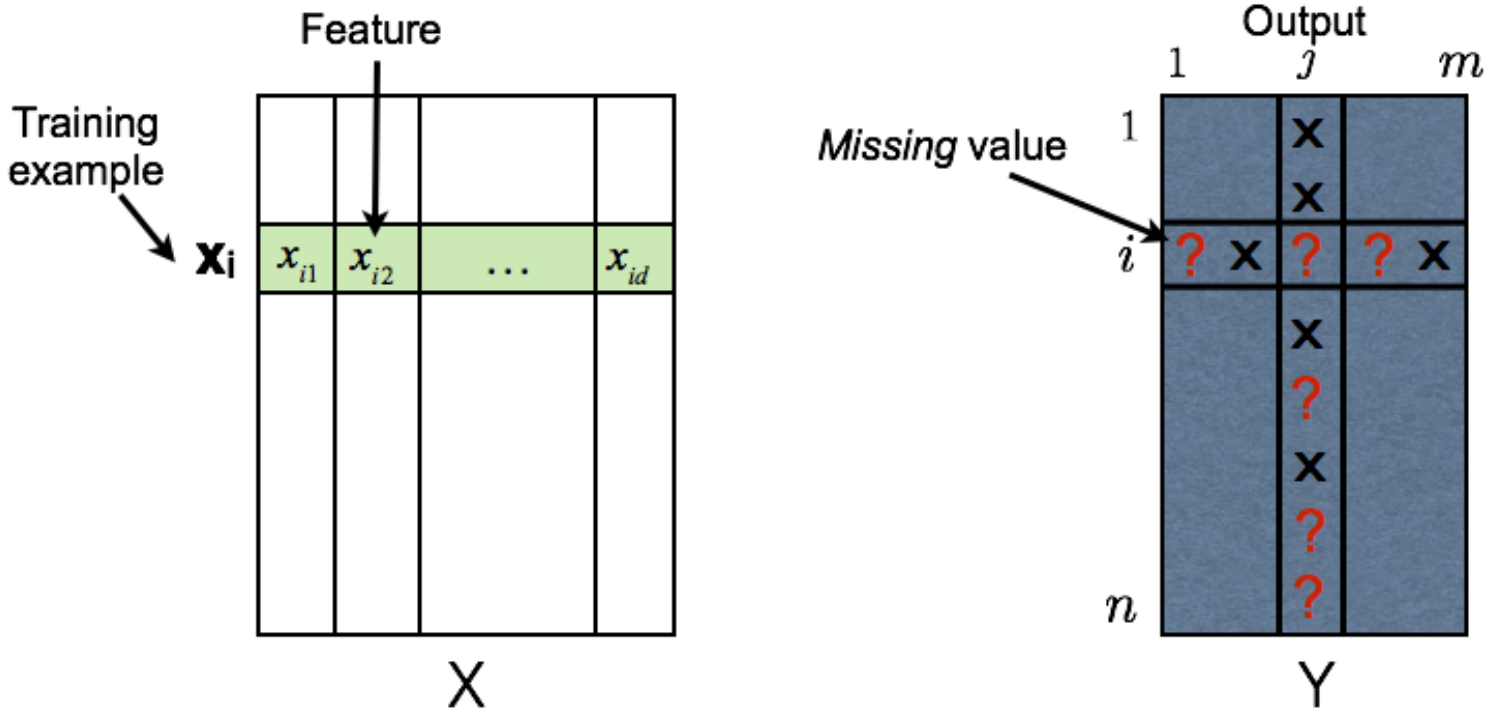
where $X = U\Sigma V^\top$ is the thin SVD of X, $M = U^\top Y$, and Y_k, M_k are the best rank-k approximations of Y & M respectively.

Multi-Output Prediction with Missing Values

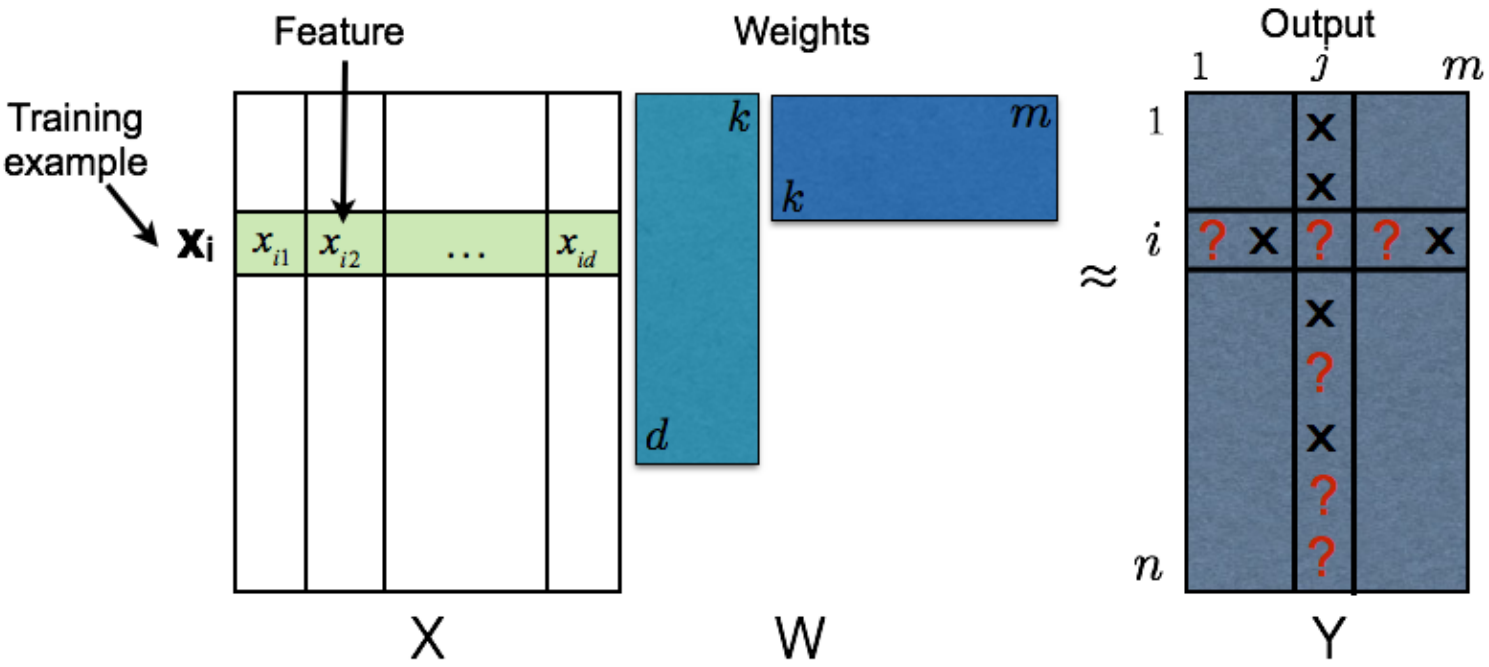
- In many applications, there may be lots of *missing* values
- E.g. Recommending tags for images and wikipedia articles

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

Multi-Output Prediction with Missing Values



Multi-Output Prediction with Missing Values



- Low-rank model: $\mathbf{y}_i = \mathbf{x}_i^\top W$ where W is low-rank

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-Output Prediction with Missing Values

- W is rank-k
- Linear predictive model: $\mathbf{y}_i \approx \mathbf{x}_i^\top W$
- Objective for low-rank multi-output regression with missing output values:

$$\min_{W:rank(W)\leq k} \quad \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W \boldsymbol{e}_j - Y_{ij})^2,$$

  where Ω is the set of observed outputs.
- No closed-form solution

Multilabel Ranking : Algorithms

- Algorithm 1 (LEML(Nuclear)): Nuclear-norm constraint objective

$$\min_{\|W\|_* \leq \mathcal{W}} \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W \boldsymbol{e}_j - Y_{ij})^2$$

  - Convex Relaxation

- Algorithm 2 (LEML(ALS)): Alternating Least Squares

$$\min_{W_1\in\mathbb{R}^{d\times k}, W_2\in\mathbb{R}^{m\times k}} \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W_1 W_2^\top \boldsymbol{e}_j - Y_{ij})^2 + \lambda(\|W_1\|_F^2 + \|W_2\|_F^2)$$

  - Alternately minimize w.r.t. W_1 and W_2
  - Non-convex optimization
  - Computationally cheaper than nuclear-norm method

Multi-Output Prediction: Modern Challenges

- Correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features

- Positive-unlabeled (PU) Learning

- Scaling Up - Millions of Correlated Outputs

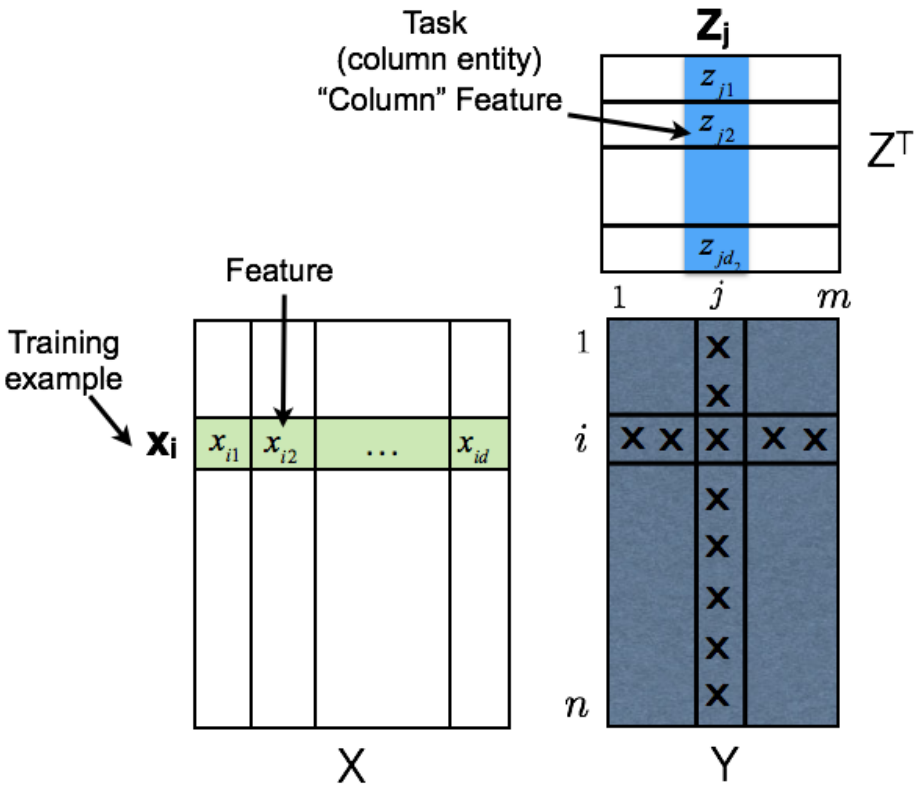
Multi-Output Prediction: Modern Challenges

- Correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features
  - Inductive Matrix Completion (IMC)
- Positive-unlabeled (PU) Learning

- Scaling Up - Millions of Correlated Outputs

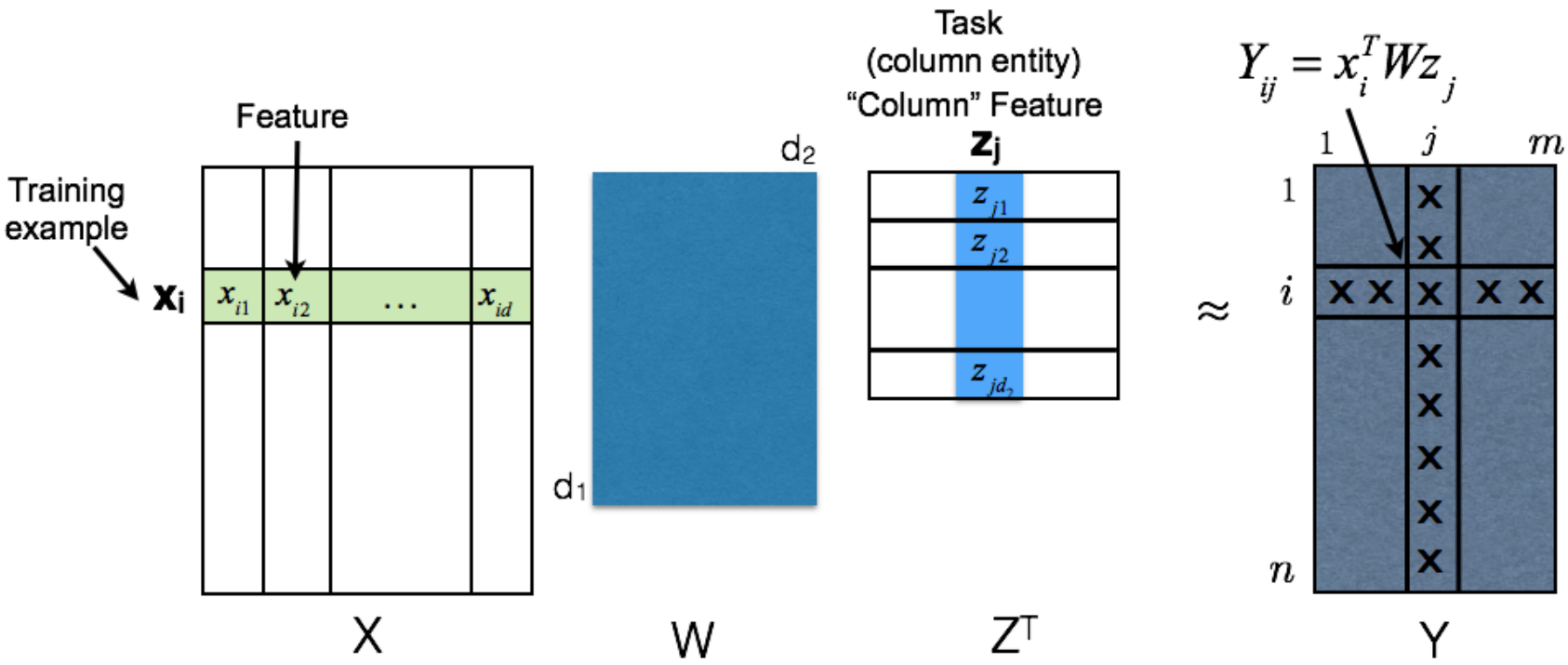
Inductive Matrix Completion (IMC)

Outputs have features



- Need to model *dyadic* or *pairwise* interactions
- Move from linear models to *bilinear* models — linear in input features *as well as* output features

Bilinear Prediction



Bilinear Prediction

- Bilinear predictive model: $Y_{ij} \approx \mathbf{x}_i^\top W \mathbf{z}_j$
- Objective for bilinear predictive model

$$\min_W \ \|Y - XWZ^\top\|_F^2$$

- Closed-form solution:

$$V\Sigma^{-1}U^\top Y\tilde{U}\tilde{\Sigma}^{-1}\tilde{V}^\top = \arg\min_W \ \|Y - XWZ^\top\|_F^2$$

where $X = U\Sigma V^\top$, $Z = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$ are the thin SVDs of X and Z

Bilinear Prediction: Low-rank Model

- W is rank-k
- Bilinear predictive model: $Y_{ij} \approx \mathbf{x}_i^\top W \mathbf{z}_j$
- Objective for low-rank bilinear model:

$$\min_{W:rank(W)\leq k} \quad \|Y - XWZ^\top\|_F^2$$

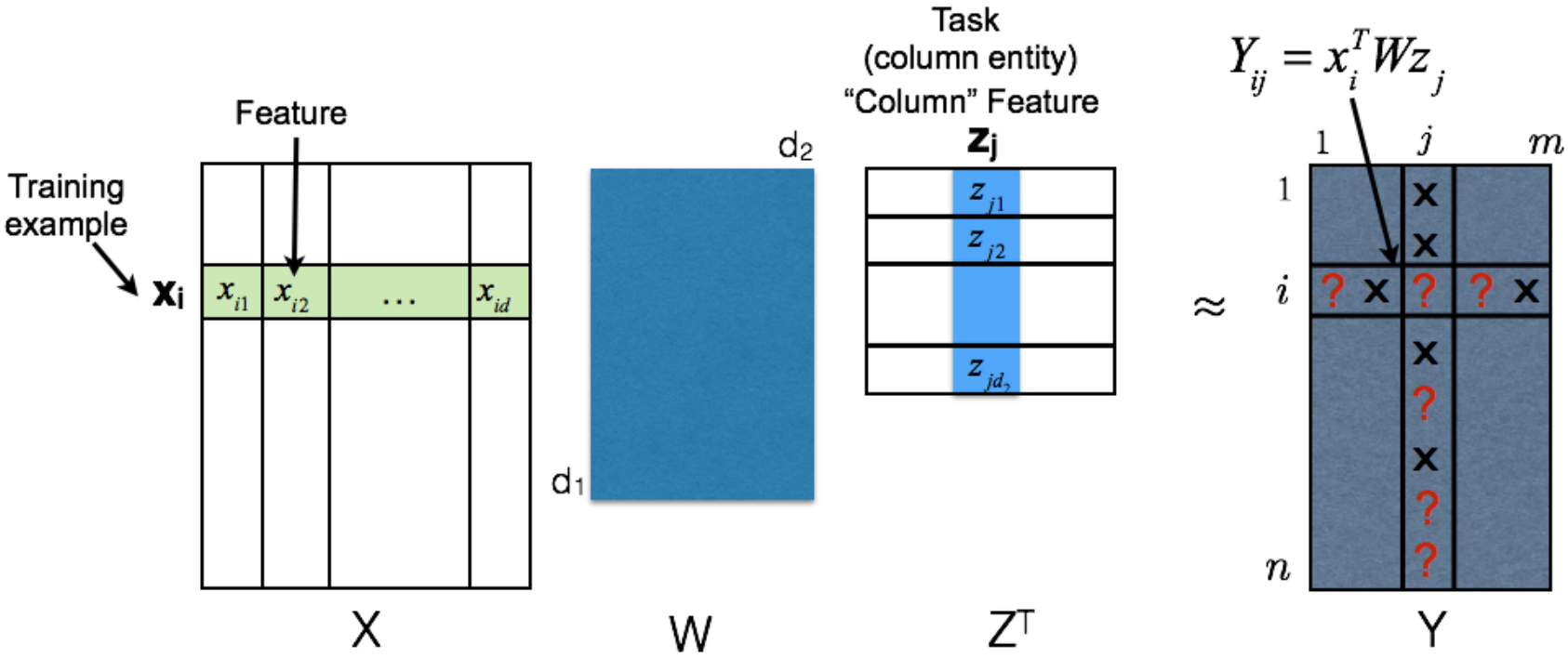
- Closed-form solution:

$$W^* = \min_{W:rank(W)\leq k} \quad \|Y - XWZ^\top\|_F^2$$

$$= \begin{cases} V\Sigma^{-1}U^\top Y_k \tilde{U}_Z \tilde{\Sigma}^{-1}\tilde{V}^\top & \text{if } X, Z \text{ are full row rank,} \\ V\Sigma^{-1} M_k \tilde{\Sigma}^{-1}\tilde{V}^\top & \text{otherwise,} \end{cases}$$

where $X = U\Sigma V^\top$, $Z = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$ are the thin SVDs of X & Z, $M = U^\top Y \tilde{U}$, and Y_k, M_k are best rank-k approximations of Y & M.

Bilinear Prediction with Missing Output Values



Bilinear Prediction with Missing Values: Algorithms

- Algorithm 1: Nuclear-norm constraint objective

$$\min_{\|W\|_* \leq \mathcal{W}} \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W \mathbf{z}_j - Y_{ij})^2$$

  - Convex Relaxation

- Algorithm 2: Alternating Least Squares (ALS)

$$\min_{W_1\in\mathbb{R}^{d\times k}, W_2\in\mathbb{R}^{d\times k}} \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W_1 W_2^\top \mathbf{z}_j - Y_{ij})^2 + \lambda(\|W_1\|_F^2 + \|W_2\|_F^2)$$

  - Non-convex optimization

Bilinear Prediction with Missing Values: Algorithms

- Algorithm 1: Nuclear-norm constraint objective

$$\min_{\|W\|_* \leq \mathcal{W}} \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W \mathbf{z}_j - Y_{ij})^2$$

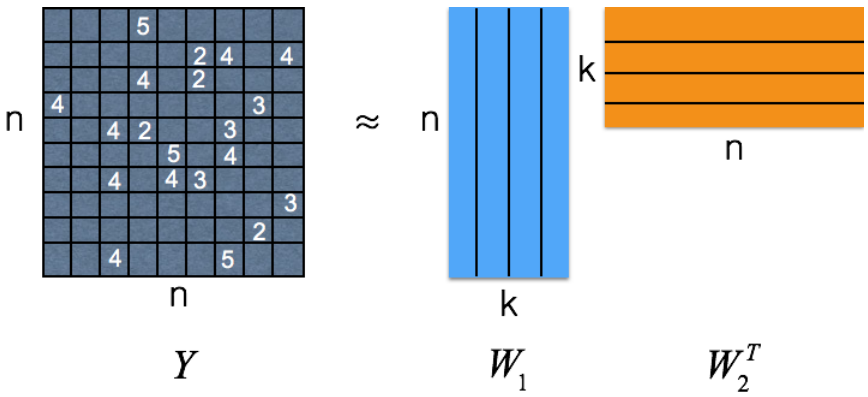
  - Convex Relaxation
- Algorithm 2: Alternating Least Squares (ALS)

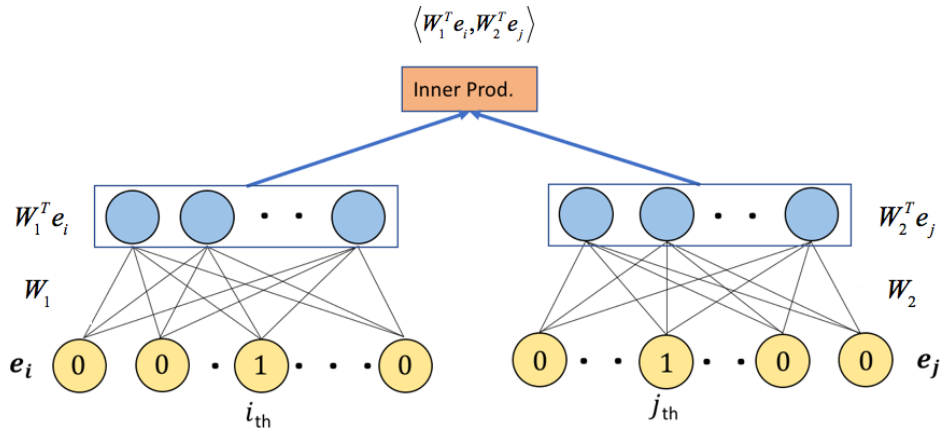
$$\min_{W_1\in\mathbb{R}^{d\times k}, W_2\in\mathbb{R}^{d\times k}} \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W_1 W_2^\top \mathbf{z}_j - Y_{ij})^2 + \lambda(\|W_1\|_F^2 + \|W_2\|_F^2)$$

  - Non-convex optimization
- Can we recover the model? How many observations are required?

Recovery Guarantees: Matrix Completion

- Matrix Completion:
  - Recover a low-rank matrix from partially observed entries
- Exact recovery requires $\tilde{O}(kn)$ observed entries



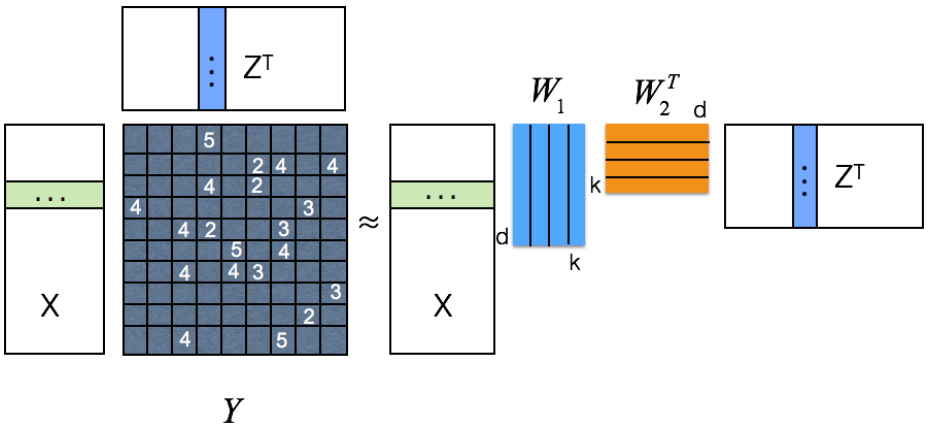


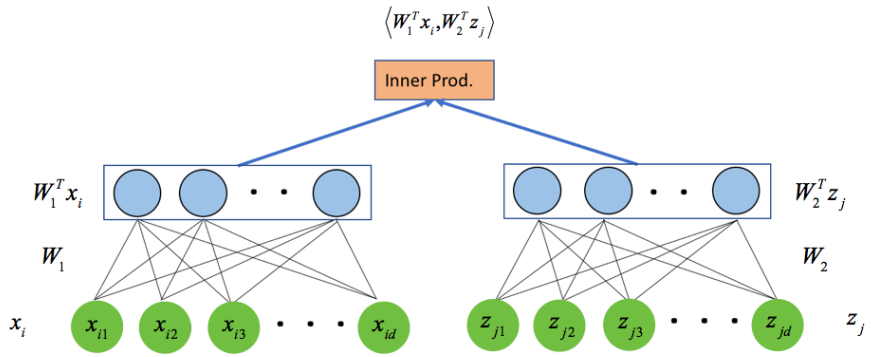
$\tilde{O}(n)$ hides polylog(n)

E. J. Candes and B. Recht. *Exact matrix completion via convex optimization.* Foundations of Computational mathematics (2009).

Inductive Matrix Completion

- Inductive Matrix Completion:
  - Recover a low-rank bilinear model from partially obtained outputs
- Degrees of freedom in W are $O(kd)$
- Can we get better sample complexity (than $\tilde{O}(kn)$)?





Recovery Guarantees

Theorem (Recovery Guarantees for Nuclear-norm Minimization)

Let $W_* = U_* \Sigma_* V_*^\top \in \mathbb{R}^{d \times d}$ be the SVD of W_* with rank k, and $Y = X W_* Z^\top$. Let $\mathcal{W} = \|W_*\|_*$. Assume X, Z 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 W_* with high probability.

The incoherence conditions are

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

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

K. Zhong, P. Jain, I. S. Dhillon. *Efficient Matrix Sensing Using Rank-1 Gaussian Measurements*. In ALT (2015).

Recovery Guarantees

Theorem (Convergence Guarantees for ALS)

Let W_* be a rank-k matrix with condition number β and $Y = XW_*Z^\top$. Assume X, Z 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, $\|W_1^H \left[W_2^{H+1}\right]^\top - W_*\|_2 \leq \epsilon$, where $H = O(\log(\|W_*\|_F/\epsilon))$.

The incoherence conditions are:

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

C2'. $\max_{i\in[n]} \| \left[W_1^h\right]^\top \mathbf{x}_i\|_2^2 \leq \frac{\mu_0 k}{n}$, $\max_{j\in[n]} \| \left[W_2^h\right]^\top \mathbf{z}_j\|_2^2 \leq \frac{\mu_0 k}{n}, \forall h = 1, 2, \cdots, H$

K. Zhong, P. Jain, I. S. Dhillon. *Efficient Matrix Sensing Using Rank-1 Gaussian Measurements*. In ALT (2015).

Sample Complexity for Recovery Guarantees

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

methods	IMC	MC
Nuclear-norm	$\tilde{O}(kd)$	$\tilde{O}(kn)$ (Recht &Cands, 2009)
ALS	$\tilde{O}(k^4\beta^2 d)$	$\tilde{O}(k^3\beta^2 n)$ (Hardt, 2014)

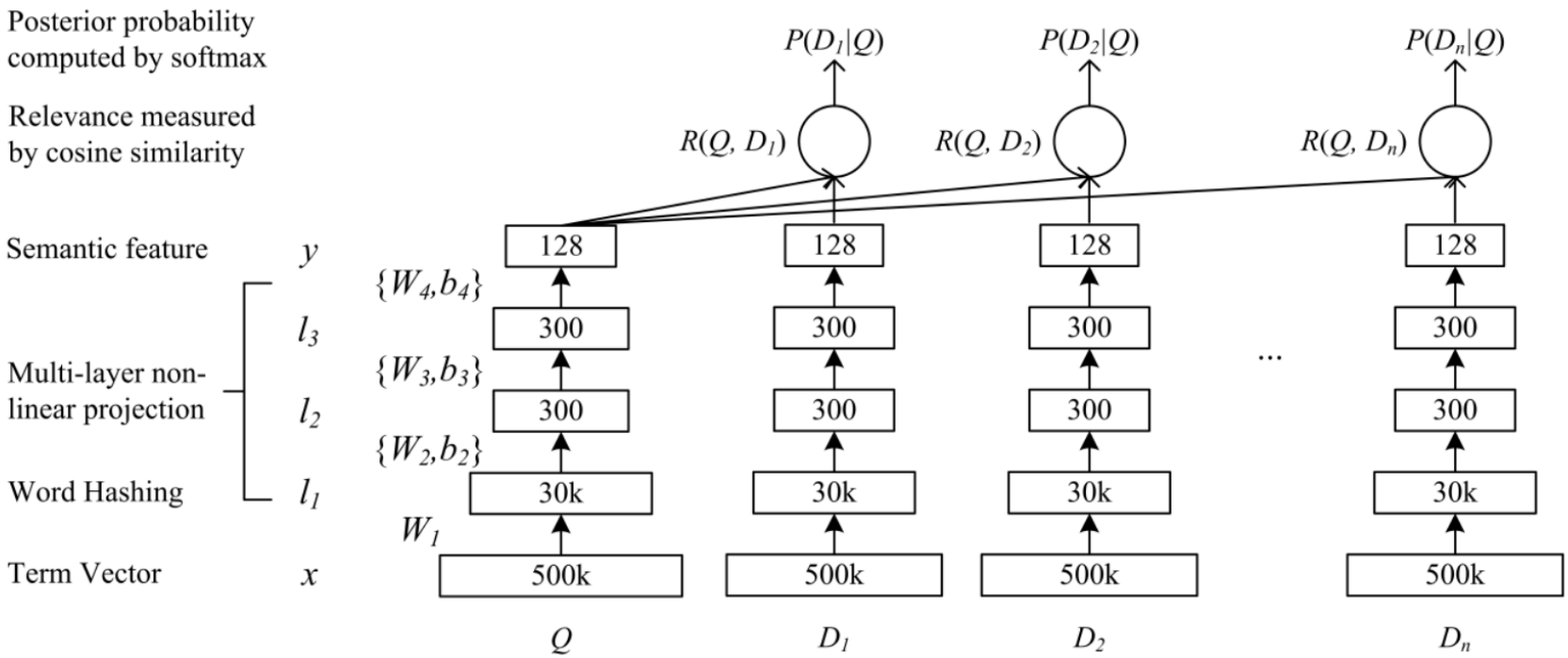
  where β is the condition number of W

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

K. Zhong, P. Jain, I. S. Dhillon. *Efficient Matrix Sensing Using Rank-1 Gaussian Measurements*. In ALT (2015).
E. J. Cands and B. Recht. *Exact matrix completion via convex optimization.* Foundations of Computational mathematics (2009).
M. Hardt. *Understanding alternating minimization for matrix completion.* Foundations of Computer Science (2014).

Multi-Output Prediction with Neural Networks

Web Search (DSSM Model)



P. S. Huang, X. He, J. Gao, L. Deng, A. Acero, & L. Heck, *Learning deep structured semantic models for web search using clickthrough data.* In Proceedings of the 22nd ACM Conference on Information & Knowledge Management (2013)

K. Zhong, Z. Song, P. Jain, and I. S. Dhillon. *Provable Non-linear Inductive Matrix Completion*. In Proceedings of The Neural Information Processing Systems Conference (NeurIPS) (2019).

Multi-Output Prediction: Modern Challenges

- Correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features
  - Inductive Matrix Completion (IMC)
- Positive-unlabeled (PU) Learning

- Scaling Up - Millions of Correlated Outputs

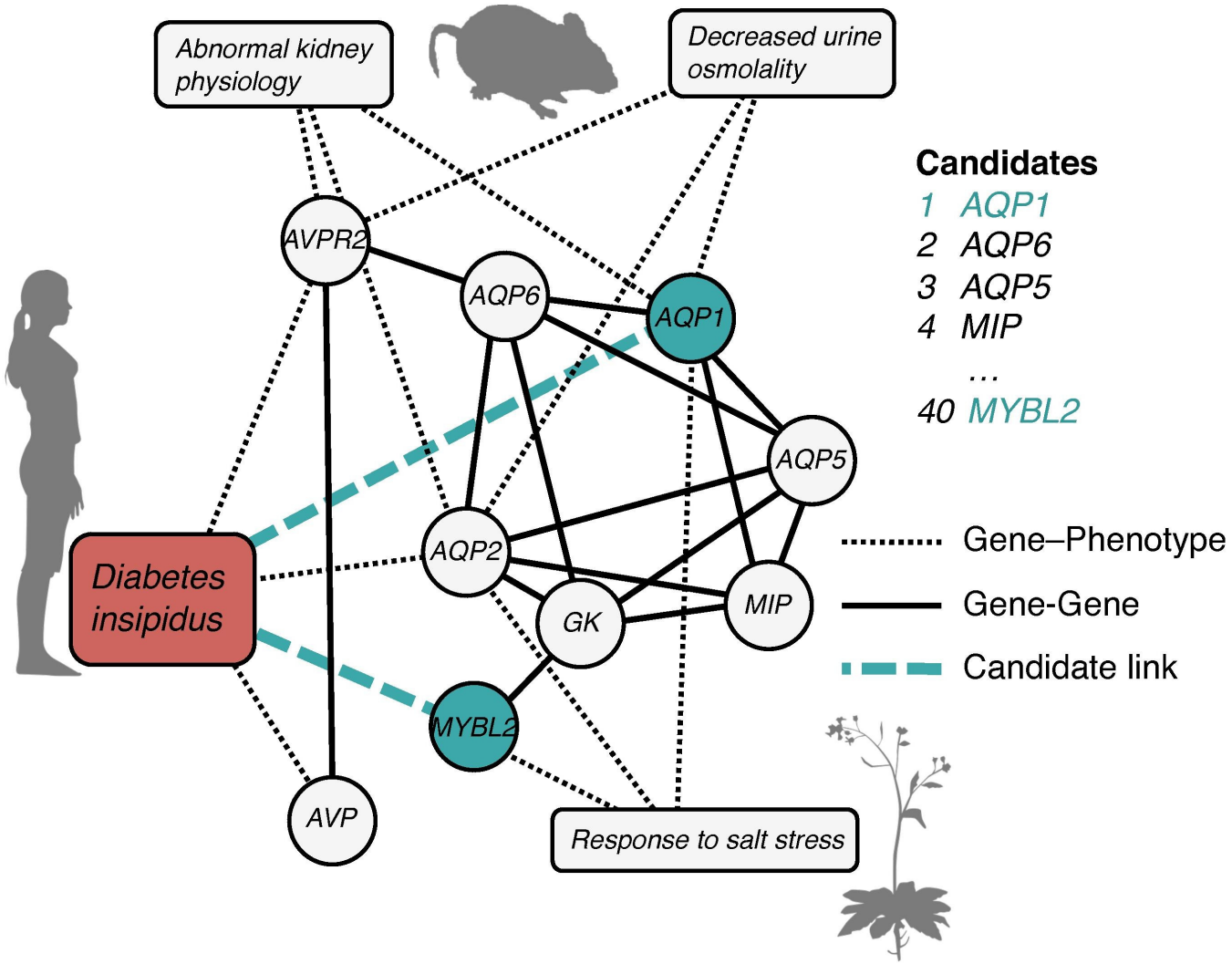
Multi-Output Prediction: Modern Challenges

- Correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features
  - Inductive Matrix Completion (IMC)
- Positive-unlabeled (PU) Learning
  - PU learning for IMC
- Scaling Up - Millions of Correlated Outputs

Positive-Unlabeled Learning

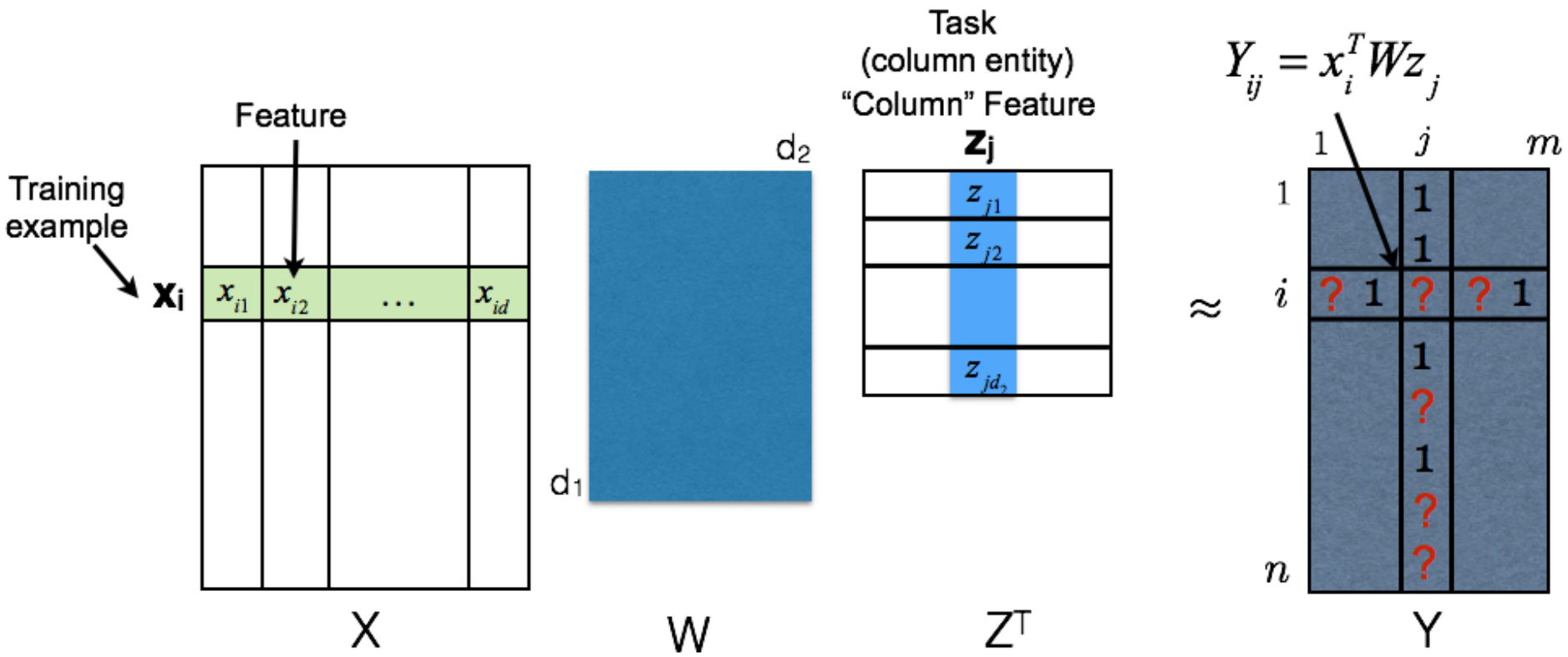
Modern Prediction Problems

Predicting related disease genes



Bilinear Prediction: PU Learning

In many applications, only "positive" labels are observed



PU Learning

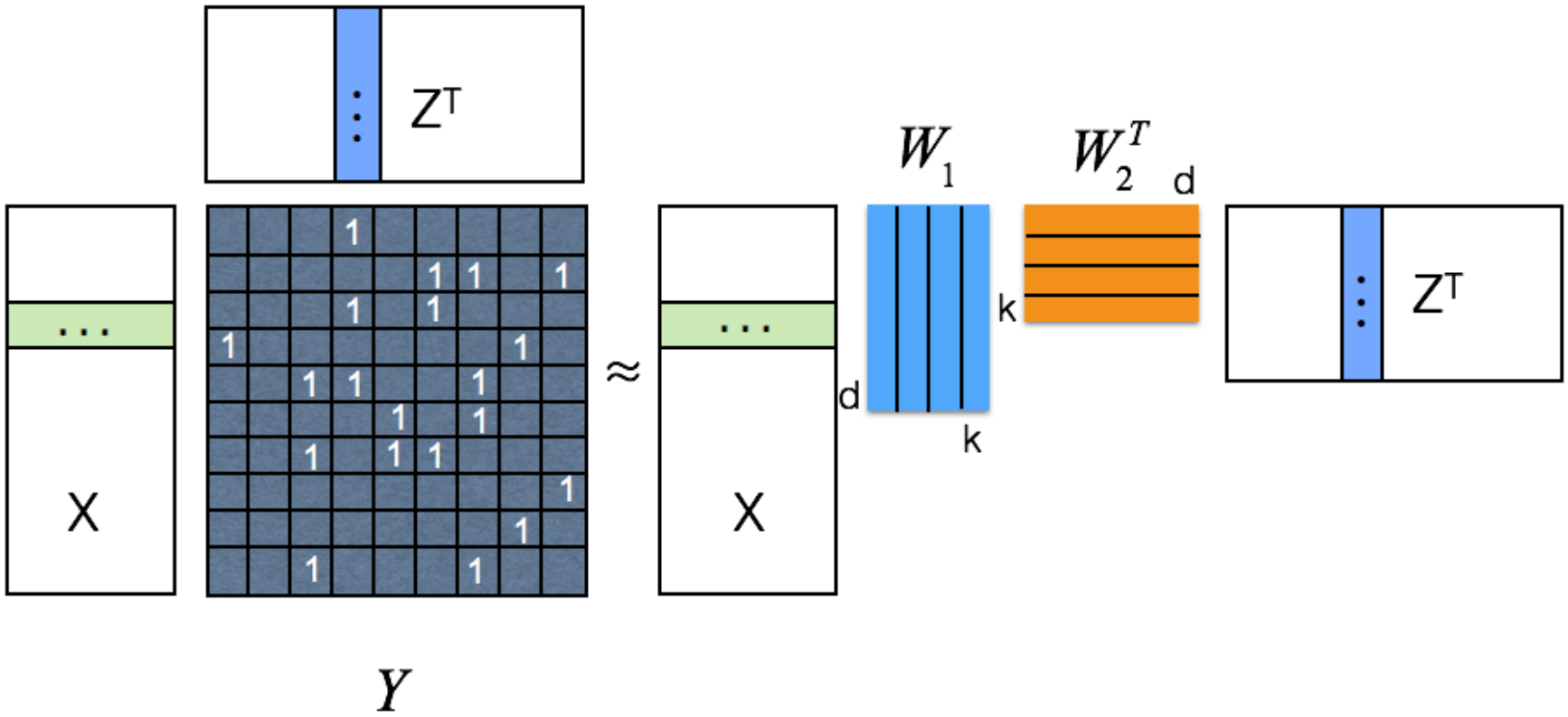
Learning Task	"Positives"	"Negatives"	"Unlabeled"
Supervised	✓	✓	
Semi-supervised	✓	✓	✓
Positive-Unlabeled (PU)	✓		✓
Unsupervised			✓

- 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:

$$\min_{\|W\|_* \leq \mathcal{X}} \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W \mathbf{z}_j - Y_{ij})^2$$

- Commonly used PU strategy: Biased Matrix Completion

$$\min_{\|W\|_* \leq \mathcal{X}} \alpha \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W \mathbf{z}_j - Y_{ij})^2 + (1-\alpha) \sum_{(i,j)\notin\Omega} (\mathbf{x}_i^\top W \mathbf{z}_j - 0)^2$$

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:

$$\min_{\|W\|_* \leq \mathcal{X}} \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W \mathbf{z}_j - Y_{ij})^2$$

- Commonly used PU strategy: Biased Matrix Completion

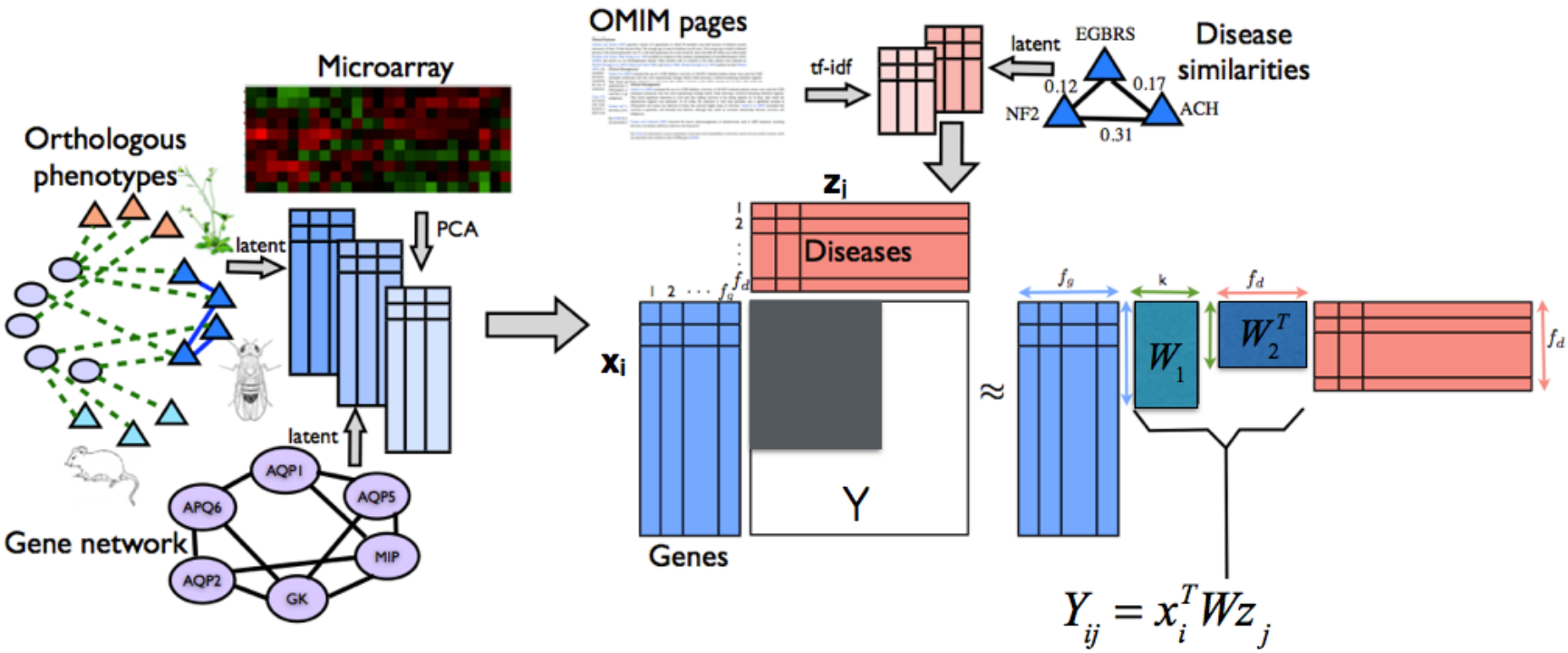
$$\min_{\|W\|_* \leq \mathcal{X}} \alpha \sum_{(i,j)\in\Omega} (\mathbf{x}_i^\top W \mathbf{z}_j - Y_{ij})^2 + (1-\alpha) \sum_{(i,j)\notin\Omega} (\mathbf{x}_i^\top W \mathbf{z}_j - 0)^2$$

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

- We can show theoretical guarantees for the biased formulation

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).

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).

Multi-Output Prediction: Modern Challenges

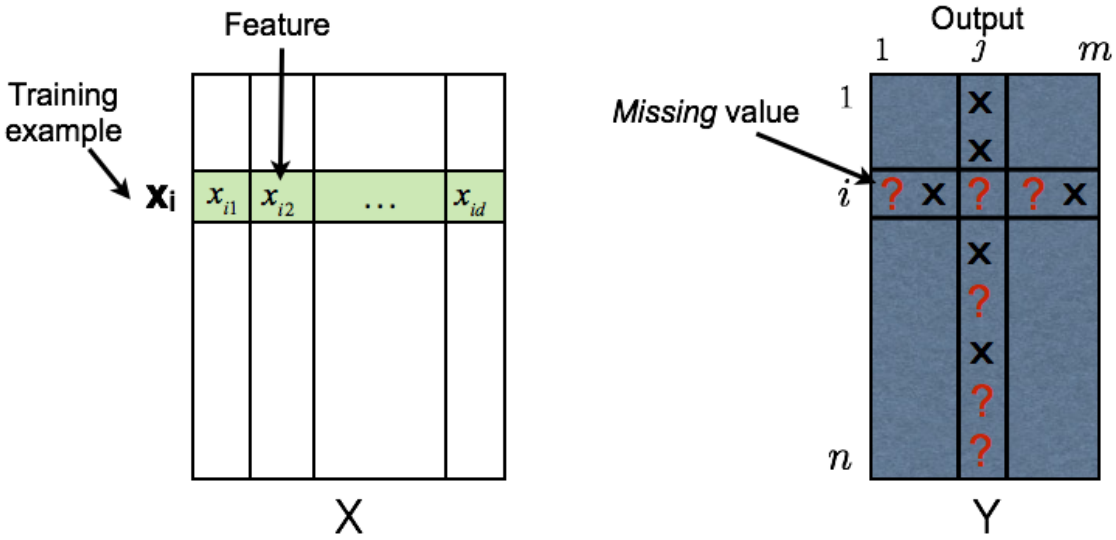
- Correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features
  - Bilinear Prediction: Inductive Matrix Completion (IMC)
- Positive-unlabeled (PU) Learning
  - PU learning for IMC
- Scaling Up - Millions of Correlated Outputs

Multi-Output Prediction: Modern Challenges

- Correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features
  - Bilinear Prediction: Inductive Matrix Completion (IMC)
- Positive-unlabeled (PU) Learning
  - PU learning for IMC
- Scaling Up - Millions of Correlated Outputs
  - Prediction for Enormous and Correlated Output Space (PECOS)

Prediction for Enormous and Correlated Output Spaces (PECOS)

eXtreme Multilabel Ranking (XMR)



- Millions of outputs/labels,
- Most values are "missing"; PU learning problem,
- Correlations between labels.

Prior Research on XMR

My Group's Research:

- LEML (ICML 14)
- GIMC (KDD 16)
- PD-Sparse (ICML 16)
- PPD-Sparse (KDD 17)
- GBDT-SPARSE (ICML 17)
- SeCSeq (NeurIPS Workshop 18)
- X-BERT (NeurIPS Workshop 19)

Other Research:

- FastXML (KDD 14)
- SLEEC (NeurIPS 15)
- PfastreXML (KDD 16)
- DiSMEC (WSDM 17)
- Parabel (WWW 18)
- Slice (WSDM 19)
- AttentionXML (NeurIPS 19)
- MACH (NeurIPS 19)

eXtreme Multilabel Ranking (XMR): Challenges

- Consider Wikipedia data set with 500K labels (output size):
  - $N = 1.5M, L = 0.5M$, Average document length around $1k$
  - TFIDF vectorizer with vocabulary $2.5M$
- Suppose we have L binary one-versus-rest classifiers:
  - Assume each classifier can be trained in 50 seconds.
  - Overall training time would be **28.9 months** on 1 CPU, and **1.8 months** with 16-way parallelization.
  - Full model (single precision) requires 5TB disk usage
- Softmax over L labels:
  - Overall training time is even worse as all the parameters need to be trained together
  - Memory footprint still 5TB.
- Naive approaches are prohibitively expensive.

eXtreme Multilabel Ranking (XMR): Challenges

- Enormous output space ($>$ 10MM outputs)
  - Computationally expensive for training and inference
  - Might require $<$ 100ms inference latency
- Long-tail output distribution
  - Paucity of training data for tail outputs.
- Only positive training examples available
  - Need to identify negatives due to enormous output space

Prediction for Enormous and Correlated Output Spaces (PECOS)

- Enormous output space ($>$ 10MM outputs)
- Long-tail output distribution
- Only positive training examples available

Prediction for Enormous and Correlated Output Spaces (PECOS)

- Enormous output space ($>$ 10MM outputs)
  - Computationally efficient schemes for training and inference
- Long-tail output distribution
  - Exploits correlation to transfer training data from head to tail outputs.
- Only positive training examples available
  - Efficiently infers "strong negatives"

PECOS: Three-Stage ML Framework

Information Retrieval (IR): a very specific XMR problem

- Each document $\leftrightarrow$ each label
- Each query $\leftrightarrow$ each instance
- #documents very large
- Both queries & documents are in same domain (text in same language)

Modern IR Systems: a very specific solution to this type of XMR problem.

- 1) Lexical indexing
- 2) Lexical matching
- 3) ML Ranking

General XMR:

- Each output might not have associated text
- Inputs and outputs might not be in same domain/space
- Input instance is presented as feature vector

PECOS: a framework for general XMR problems.

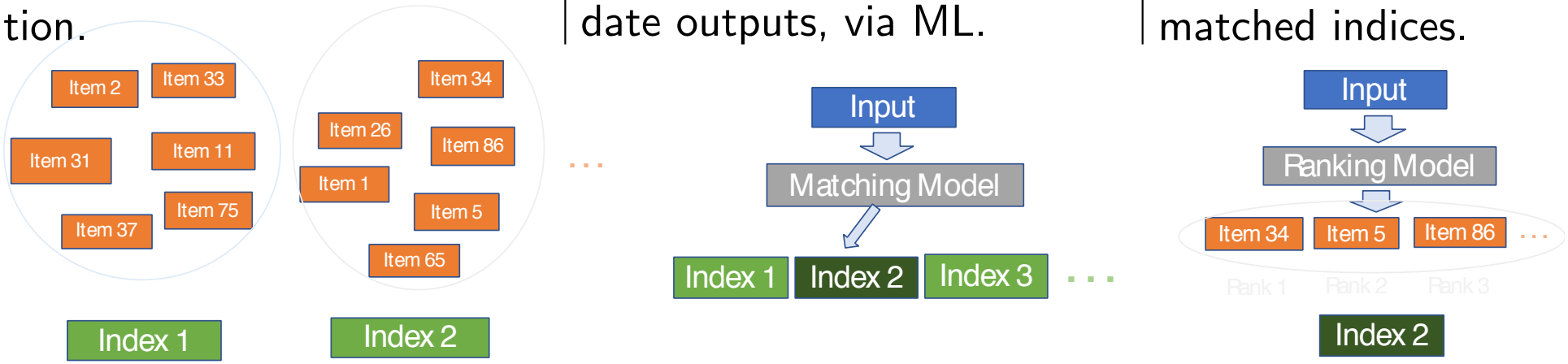
- Motivated by modern IR systems:
- 1) **Semantic** Label Indexing
- 2) ML Matching
- 3) Ensemble ML Ranking

PECOS: Three-stage Framework

1. **Indexing**: Index all outputs using semantic information.

2. **ML Matching**: Find relevant indices, i.e., the candidate outputs, via ML.

3. **Ranking**: Rank candidates from matched indices.



- Inference time complexity reduction

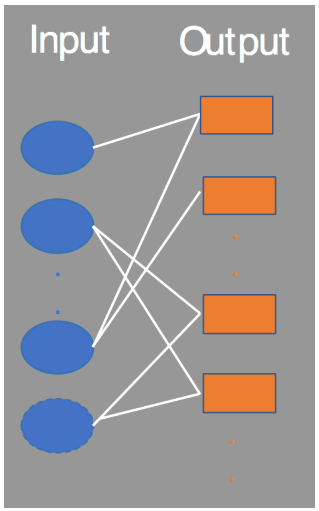
$$O(\#outputs) \rightarrow \underbrace{O(\#indices)}_{\text{Matching}} + \underbrace{O(\#candidates)}_{\text{Ranking}}$$

- Correlated outputs are in the same index
- Transfers information to tail outputs

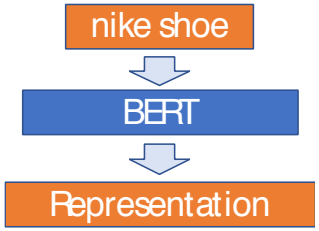
Stage 1: Semantic Indexing

1. Build semantic representations for outputs

- Represent outputs through input output relationships

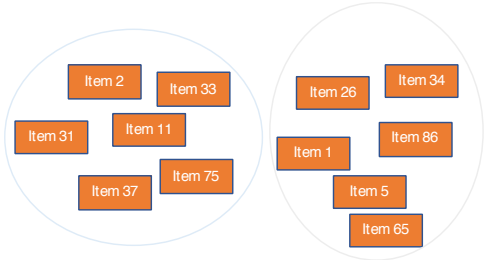


- Extract semantic features from text, such as ELMO, BERT, etc.

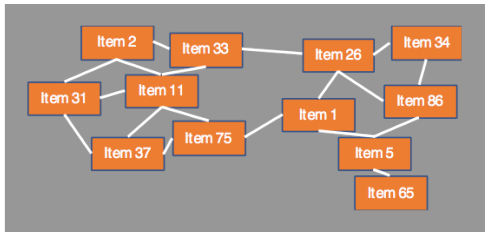


2. Index outputs using representations

- **hierarchical k-means clustering**,



- Approximate nearest neighbor graph,



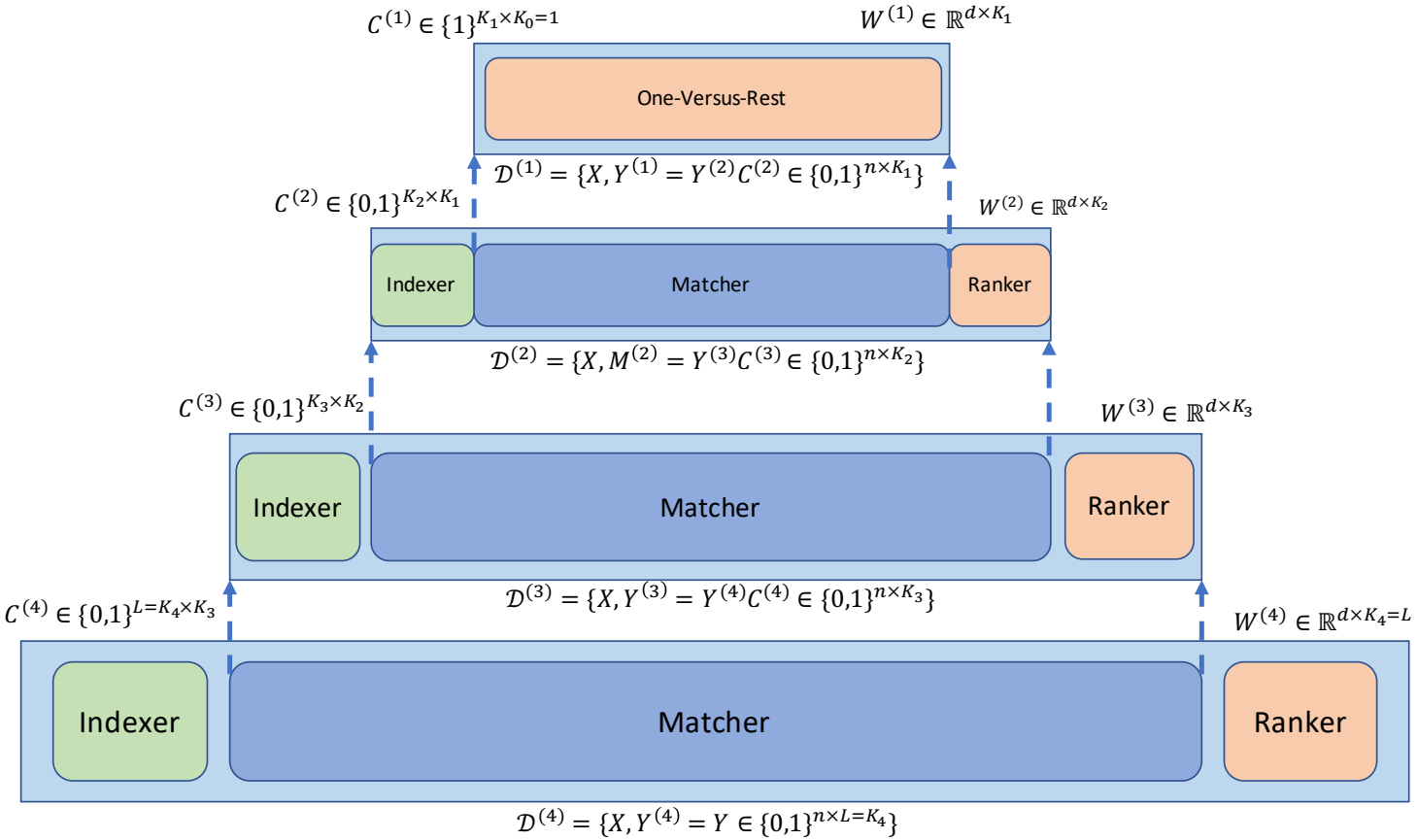
- Other indexing structures.

Stage 3: Efficient Ranking

- Ranking using TF-IDF features v.s. deep learning features
- TF-IDF features:
  - Average number of non-zeros of TF-IDF features <20.
    - Time complexity $= 20 \times O(\#candidates)$
  - Weight vectors with L1-regularization can also be extremely sparse.
  - Use dictionary-based data structure for sparse weights to further reduce time complexity.
- DL features:
  - Dimension, such as BERT ≥ 768.
    - Time complexity $= 768 \times O(\#candidates)$
  - In batch mode, it can be accelerated by GPU.
  - In real-time mode, GPU acceleration is limited.

Stage 2: XR-Linear (Recursive Linear Matching)

- Observation: With hierarchical clustering, sub-problem for the matcher is also an XMR problem.
- Matching can be done recursively:



Vectorizers for Text Inputs

- Given $\mathbf{t}_i$ the text sequence associated with the ith input.
- Function $\mathbf{x} = \varphi(\mathbf{t}, \Theta)$ converts $\mathbf{t}$ into a d-dimensional feature vector.
  - Example 1: Term Frequency-Inverse Document Frequency

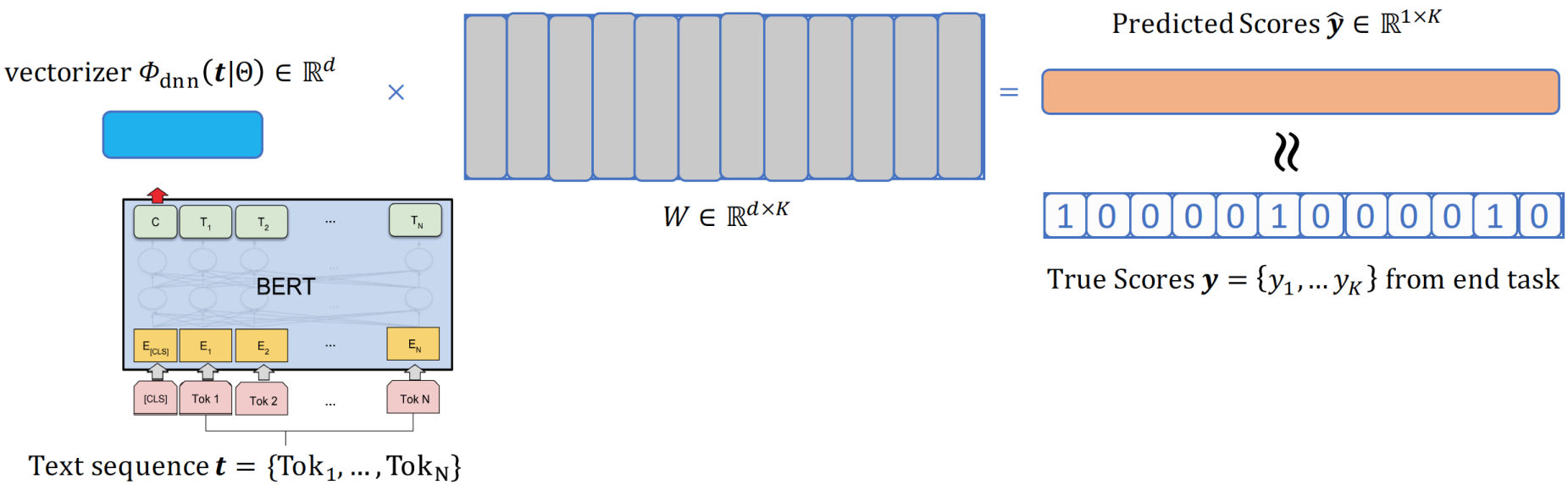
$$\mathbf{x} = \varphi_{tfidf}(\mathbf{t}, \Theta)$$

  - Example 2: Deep Pre-trained Transformers, e.g., BERT.

$$\mathbf{x} = \varphi_{dnn}(\mathbf{t}, \Theta)$$

- By adding a task-specific linear layer on top, the DNN model provides a trainable vectorizer.

Deep Learned Matcher for Text Inputs



- Incorporate a trainable deep text vectorizer for a deep learned matcher:

$$g_{dnn}(\mathbf{t}, k) = \mathbf{w}_k^\top \varphi_{dnn}(\mathbf{t}, \Theta)$$

Stage 2: X-Transformer

Three choices for deep text vectorizers:

- BERT: Deep Bidirectional Transformers
- XLNet: Generalized autoregressive pre-training with Transformer-XL
- RoBERTa: robustly optimized version of BERT

X-Transformer for trainable deep text vectorizer:

- Training Objective:

$$\min_{W,\Theta} \sum_i \mathcal{L}\left(\boldsymbol{C}^\top \mathbf{y}_i, \mathbf{W}^\top \varphi_{dnn}(\mathbf{t}_i, \Theta)\right)$$

- $\mathcal{L}(a, b) = \max(0, 1 - ab)^2$ is the squared hinge loss.
- $\boldsymbol{C} \in \{0, 1\}^{L \times K}$ is the label-to-cluster assignment matrix.
- $\mathbf{y}_i \in \{0, 1\}^L$ is the label assignment vector for instance i.

PECOS Results

- **AmazonCat-13K**: $(|\mathcal{Y}| = 13,330,\ n_{\text{trn}} = 1,186,239,\ n_{\text{tst}} = 306,782)$

	Prec@1	Prec@3	Prec@5	Recall@1	Recall@3	Recall@5
X-TRANSFORMER	96.65	83.74	68.42	27.60	63.35	79.51
XR-LINEAR	94.22	79.95	64.77	26.77	60.74	75.97
fasttext						
hierarchical softmax	86.77	73.26	58.99	24.67	56.50	70.58
softmax	90.55	77.36	62.92	25.75	59.35	74.61

PECOS Results

- **Wiki-500K** ($|\mathcal{Y}| = 501,070,\ n_{\text{trn}} = 1,779881,\ n_{\text{tst}} = 769,421$)

	Prec@1	Prec@3	Prec@5	Recall@1	Recall@3	Recall@5
X-TRANSFORMER	78.19	58.26	45.84	25.98	48.61	58.69
XR-LINEAR	65.17	46.39	36.28	20.92	37.93	45.80
fasttext						
hierarchical softmax	31.59	18.47	13.47	10.70	16.44	18.92

Hardware Cost Estimates

- **AmazonCat-13K**: $(|\mathcal{Y}| = 13,330,\ n_{\text{trn}} = 1,186,239,\ n_{\text{tst}} = 306,782)$
- Yearly Training Cost in Dollars based on AWS machines:
  - i2.4xlarge: \$1.248 per hour, 16CPUs
  - p3.16xlarge: \$ 24.48 per hour, 8GPUs

	Prec@1	Time(s)	Daily Training	Weekly Training	Monthly Training
XR-LINEAR	94.22	3,176.2	\$401	\$57	\$13
X-TRANSFORMER	96.65	464,130.0	\$1,151,971	\$164,116	\$37,873

Hardware Cost Estimates

- **Wiki-500K** ($|\mathcal{Y}| = 501,070,\ n_{\text{trn}} = 1,779881,\ n_{\text{tst}} = 769,421$)
- Yearly Training Cost in Dollars based on AWS machines:
  - i2.4xlarge: \$1.248 per hour, 16CPUs
  - p3.16xlarge: \$ 24.48 per hour, 8GPUs

	Prec@1	Time(s)	Daily Training	Weekly Training	Monthly Training
XR-LINEAR	65.17	19,040.3	\$4,818	\$686	\$158
X-TRANSFORMER	78.19	1,242,580.0	\$3,084,084	\$439,376	\$101,395

- Practitioner should analyze cost-benefit tradeoff, which might change over time

Conclusions and Future Work

- Millions of correlated outputs, and missing output values
- Outputs have features
- Positive-unlabeled (PU) Learning
- Scaling Up - Millions of Correlated Outputs

Conclusions and Future Work

- Millions of correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features
  - Inductive Matrix Completion (IMC)
- Positive-unlabeled (PU) Learning
  - PU learning for IMC
- Scaling Up - Millions of Correlated Outputs
  - Prediction for Enormous and Correlated Output Space (PECOS)

Conclusions and Future Work

- Millions of correlated outputs, and missing output values
  - Low-rank + Alternating Least Squares
- Outputs have features
  - Inductive Matrix Completion (IMC)
- Positive-unlabeled (PU) Learning
  - PU learning for IMC
- Scaling Up - Millions of Correlated Outputs
  - Prediction for Enormous and Correlated Output Space (PECOS)

- Future Work:
  - More efficient neural network training and inference
  - End-to-end training
  - Extensions to Contextual Bandits and Reinforcement Learning

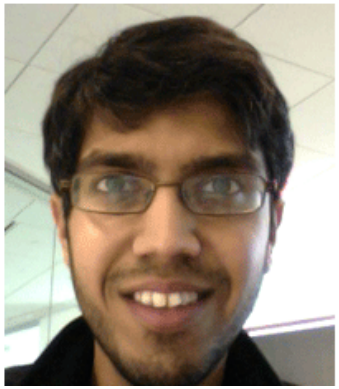
Collaborators

Wei-Cheng Chang

Cho-Jui Hsieh

Prateek Jain

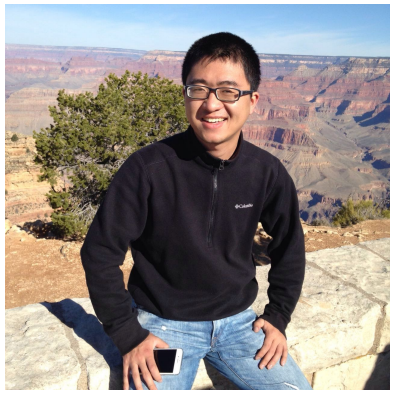
Nagarajan Natarajan



Nikhil Rao

Si Si

Hsiang-fu Yu



Kai Zhong

References

[1] H. F. Yu, K. Zhong, and I. S. Dhillon. *Prediction for Enormous and Correlated Output Spaces*. Working Manuscript (2020).

[1] P. Jain, and I. S. Dhillon. *Provable inductive matrix completion*. In arXiv preprint arXiv:1306.0626 (2013).

[2] K. Zhong, P. Jain, I. S. Dhillon. *Efficient Matrix Sensing Using Rank-1 Gaussian Measurements*. In ALT (2015).

[3] N. Natarajan, and I. S. Dhillon. *Inductive matrix completion for predicting gene disease associations*. In 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 ICML (2014).

[5] C-J. Hsieh, N. Natarajan, and I. S. Dhillon. *PU Learning for Matrix Completion*. In ICML (2015).

[6] S. Si, K.-Y. Chiang, C.-J. Hsieh, N. Rao, and I.S.Dhillon *Goal-Directed Inductive Matrix Completion* In KDD, 2016.

[7]. K. Zhong, Z. Song, P. Jain, and I. S. Dhillon. *Provable Non-linear Inductive Matrix Completion*. In NeurIPS (2019).

[8] K. Zhong, Z. Song, P. Jain, P. L. Bartlett & I. S. Dhillon. *Recovery Guarantees for One-hidden-layer Neural Networks.* In ICML (2017)