Learning with Matrix Parameters

Nati Srebro

Matrix Completion

movies

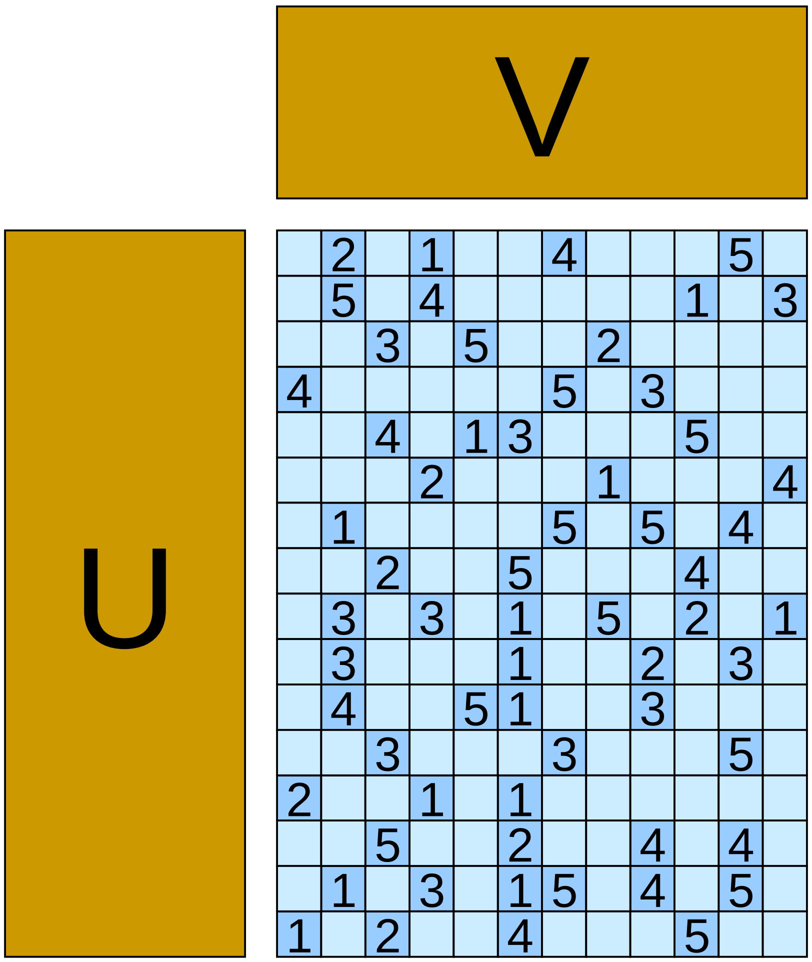
users

	2		1			4				5	
	5		4				?		1		3
		3		5			2				
4			?			5		3		?	
		4		1	3				5		
			2				1	?			4
	1					5		5		4	
		2		?	5		?		4		
	3		3		1		5		2		1
	3				1			2		3	
	4			5	1			3			
		3				3	?			5	
2	?		1		1						
		5			2	?		4		4	
	1		3		1	5		4		5	
1		2			4				5	?	

- Predictor itself is a matrix

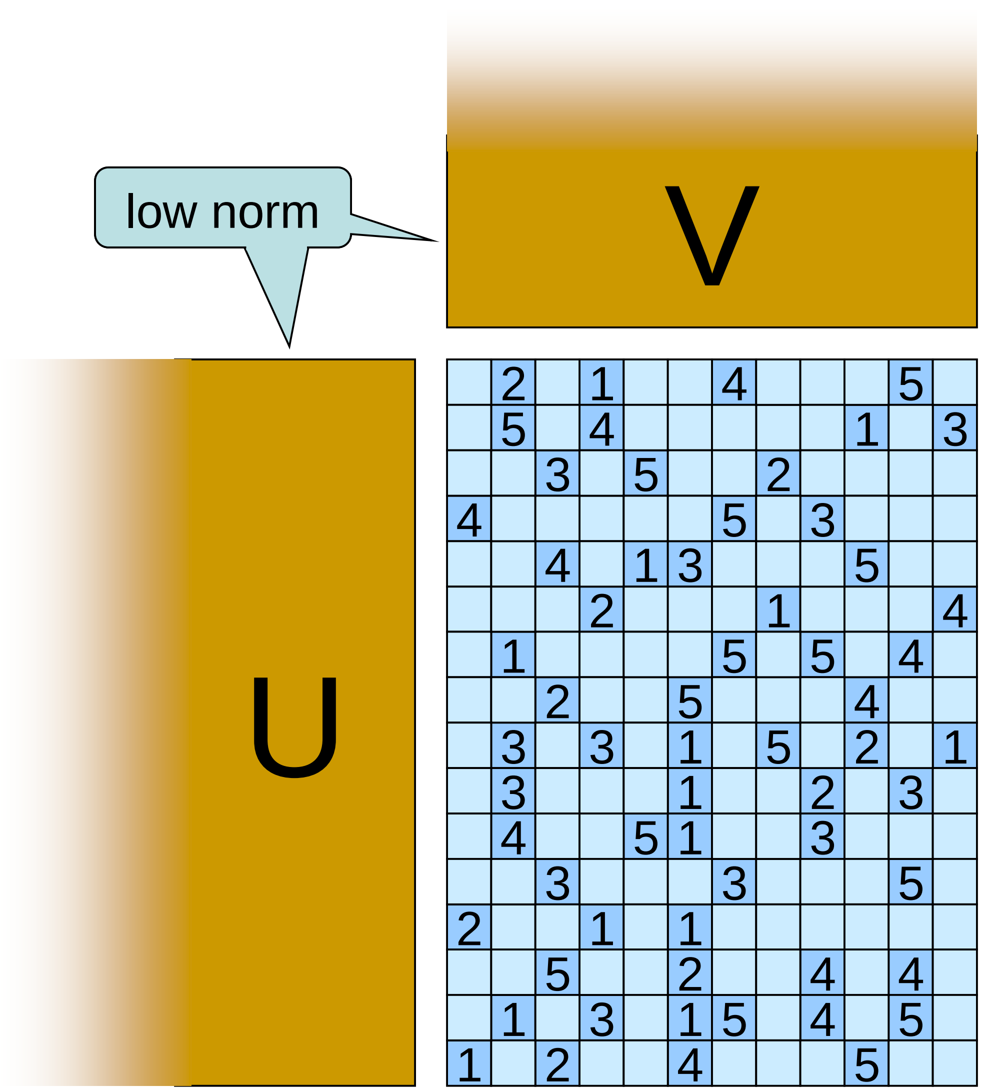
- To learn: **need bias**
  (prior / hypothesis class / regularizer)
- Elementwise (i.e. treat matrix as vector)
  → can't generalize
- Matrix constraints/regularizers:
  - Block/cluster structure (eg Plaid Model)
  - Rank
  - Factorization Norms: Trace-Norm, Weighted Tr-Norm, Max-Norm, Local Max-Norm, …
  - Spectral Regularizers
  - Group Norms

Matrix Factorization Models



$$\text{rank}(X) = \min_{X=UV'} \dim(U, V)$$

Matrix Factorization Models



$$\text{rank}(X) = \min_{X=UV'} \dim(U,V)$$

Bound avg norm of factorization:

$$\|U\|_F^2 = \sum_i |U_i|^2$$

$$\|X\|_{tr} = \min_{X=UV'} \|U\|_F \cdot \|V\|_F$$

Bound norm of fact. uniformly:

$$\|U\|_{2,\infty} = \max_i |U_i|$$

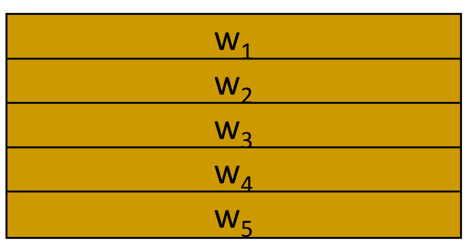
$$\|X\|_{max} = \min_{X=UV'} \|U\|_{2,\infty} \cdot \|V\|_{2,\infty}$$

aka $\gamma_2 : \ell_1 \rightarrow \ell_\infty$ norm

Transfer in Multi-Task Learning

- m related prediction tasks: **[Argyriou et al 2007]**
  Learn predictor ϕ_i for each task $i = 1..m$
- m classes, predict with $\arg\max_y \phi_y(x)$ **[Amit et al 2007]**
- Transfer from learned tasks to new task
- Semi-supervised learning:

  create auxiliary tasks from unlabeled data (e.g. predict held-out word from context), transfer from aux task to actual task of interest (e.g. parsing, tagging) **[Ando Zhang 2005]**

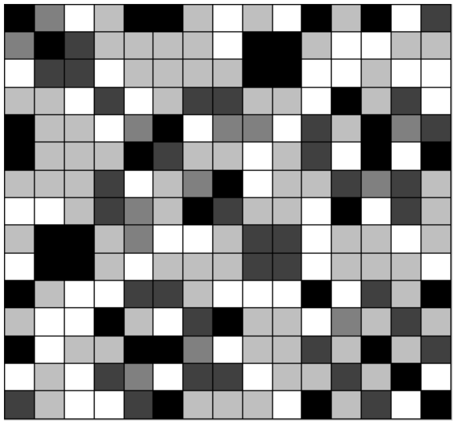


Factorization model ≡ two layer network, shared units (learned features) in hidden layer

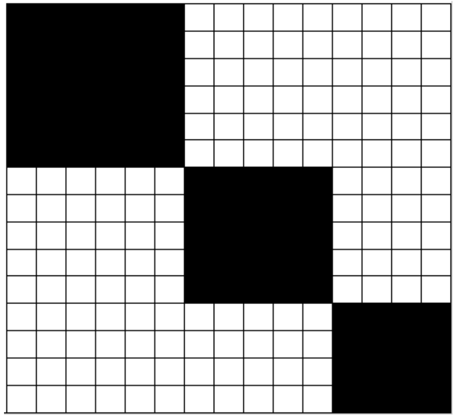
- Predictors naturally parameterized by a matrix (but there is no requirement that we output a matrix)

Correlation Clustering as Matrix Learning

[Jalaia et al 2011,2012]



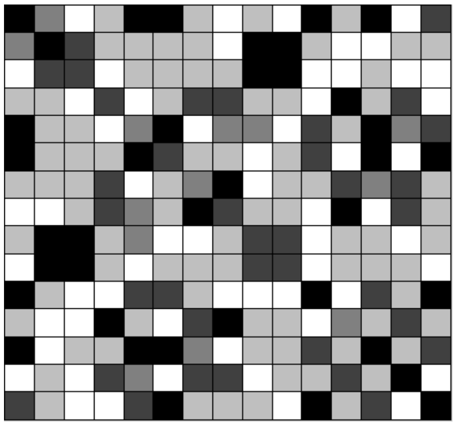
input similarity



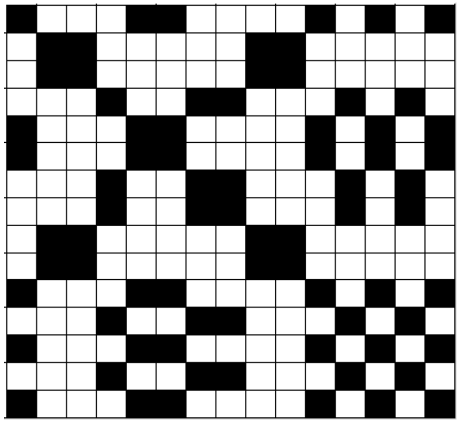
clustering matrix

Correlation Clustering as Matrix Learning

[Jalaia et al 2011,2012]

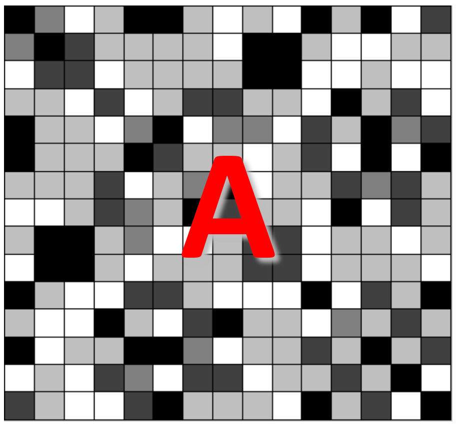


input similarity



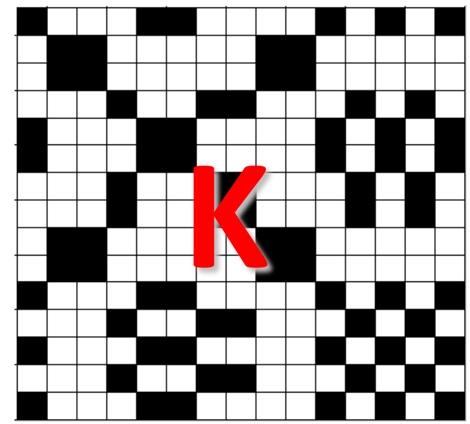
clustering matrix

Correlation Clustering as Matrix Learning [Jalaia et al 2011,2012]



input similarity

$\approx$



clustering matrix

$\min_K ||K-A||_1$ s.t. K is permuted block-1 diag.

- Can represent desired object as a matrix

Also:

- Corwdsourced similarity learning [Tamuz et al 2011]
- Binary hashing [Tavakoli et al 2013]
- Collaborative permutation learning

Covariance/Precision Matrix Estimation

- Learning Mahalanobis Metric

$$d(x, y) \propto \exp(-x' \mathrm{M} x)$$

- Inferring Dependency Structure
  - Sparse Markov Net ➔ Sparse Precision Matrix
  - *k* Latent Variables ➔ + Rank *k*
  - Many latent variables with regularized affect ➔
    + Trace-Norm/Max-Norm

Principal Component Analysis

- View I: low rank matrix approximation
  - $\min_{rank(A)\leq k} ||A - K||$
  - Approximating matrix itself is a matrix parameter
  - Does not give compact representation
  - Does not generalize

- View II: find subspace capturing as much of data distribution as possible
  - Maximizing variance inside subspace: $\max E[\|Px\|^2]$
  - Minimizing reconstruction error: $\min E[\|x - Px\|^2]$
  - Parameter is low-dim subspace ➔ represent as matrix

Principal Component Analysis: Matrix Representation

- $\min_{rank(A)\leq k} ||A - X||$

- Represent subspace using basis matrix $U \in R^{d\times k}$

$$\min E\left[\min_{v} \|x - Uv\|^2\right]$$
$$= \min_{0 \preccurlyeq U \preccurlyeq I} E[x'(I - UU')x]$$

- Represent subspace using projector $P = UU' \in R^{d\times d}$

min E[x'(I-P)x]

s.t. $0 \preccurlyeq P \preccurlyeq I$

rank(P)≤k

Principal Component Analysis: Matrix Representation

- $\min_{rank(A)\leq k} ||A - X||$

- Represent subspace using basis matrix $U \in R^{d\times k}$

$$\min E\left[\min_{v} \|x - Uv\|^2\right]$$

$$= \min_{0 \preccurlyeq U \preccurlyeq I} E[x'(I - UU')x]$$

- Represent subspace using projector $P = UU' \in R^{d\times d}$

min E[x'(I-P)x]

s.t. $0 \preccurlyeq P \preccurlyeq I$

~~rank(P)≤k~~ tr(P)≤k

- Optimum preserved
- Efficiently extract rank-k $\tilde{P}$ using rand rounding, without loss in objective

Matrix Learning

- Matrix Completion, Direct Matrix Learning
  - Predictor itself is a matrix
- Multi-Task/Class Learning
  - Predictors can be parameterized by a matrix
- Similarity Learning, Link Prediction, Collaborative Permutation Learning, Clustering
  - Can represent desired object as a matrix
- Subspace Learning (PCA), Topic Models
  - Basis Matrix or Projector

What is a good inductive bias?

Desired output ***must*** have specific structure

Possible Inductive Bias: Matrix Constraints / Regularizers

Elementwise

Frobenious: $|X|_2$

$|X|_1$

$|X|_\infty$

Factorization

Rank

Trace-Norm

Weighted Tr-Norm

Max-Norm

Local Max-Norm

NMF

Sparse MF

Operator Norms

Spectral Norm $\|X\|_2$

Group Norms

Group Lasso

$\|X\|_{2,\infty}$

Structural

Plaid Models

Block Structure

Spectral Functions

- Spectral function: F(X) = f(singular values of X)
- F is spectral iff it is rotation invariant: F(X)=F(UXV')
- Examples:
  - rank(X) = $|\text{spectrum}|_0$
  - Frobenious $|X|_2$ = $|\text{spectrum}|_2$
  - Trace-Norm = $|\text{spectrum}|_1$
  - Spectral Norm $||X||_2$ = $|\text{spectrum}|_\infty$
  - Positive semi-definite $\equiv$ spectrum ≥ 0
  - Trace of p.s.d. matrix = $\sum$spectrum
  - Relative entropy of spectrum
- Can lift many vector properties:
  - Convexity, (strong convexity)
  - $\nabla F(X) = U \nabla f(S) V'$
  - Projection operations
  - Duality: $F^*(X)=Uf^*(S)V'$
  - Mirror-Descent Updates (e.g. "multiplicative matrix updates")
  - $\approx$ Concentration Bounds

Possible Inductive Bias: Matrix Constraints / Regularizers

Elementwise

Frobenious: $|X|_2$

$|X|_1$

$|X|_\infty$

Factorization

Rank

Trace-Norm

Weighted Tr-Norm

Max-Norm

Local Max-Norm

NMF

Sparse MF

Operator Norms

Spectral Norm $||X||_2$

Group Norms

Group Lasso

$||X||_{2,\infty}$

Structural

Plaid Models

Block Structure

Learning with Matrices

- Matrices occur explicitly or implicitly in many learning problems
- Advantages of matrix view (even when not explicit): can use existing tools, relaxations, opt methods, concentration and generalization bounds, etc

- What is a good inductive bias?
  - Is some structure required or is it just an inductive bias?

- Spectral functions convenient, but don’t capture everything!