Multi-Scale Link Prediction

Donghyuk Shin
Dept of Computer Science
UT Austin

CIKM'12
Oct 30, 2012

Joint work with S. Si and I. S. Dhillon

Outline

- Social Network Analysis
  - Link prediction
  - Matrix functions
  - Low-rank approximations
- Multi-Scale Link Prediction
  - Hierarchical clustering
  - Subspace approximation
  - Multi-scale prediction
- Experimental Results
- Conclusion and Future Work

Social Network Analysis

Networks

- Social networks (Facebook, MySpace, LiveJournal, ...)
- Biological networks (linking genes to diseases)
- Web graphs
- Collaboration and citation networks
- ...

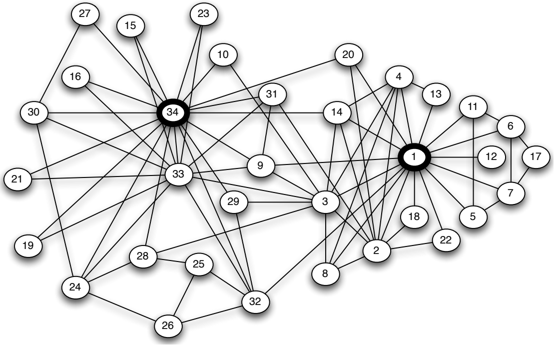
Challenges

- Massive scale — Facebook's social graph has 1 billion vertices
- Need scalable algorithms and "good" predictions

Social Networks

- Nodes: individual actors (people or groups of people)
- Edges: relationships (social interactions) between nodes

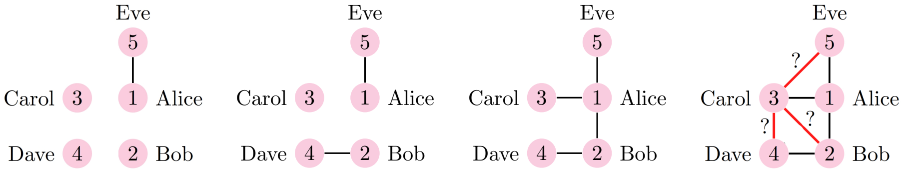
Example: Karate Club Network:



Link Prediction Problem

Consider a social network that evolves with time,

$$\cdots \longrightarrow A^{(t-2)} \longrightarrow A^{(t-1)} \longrightarrow A^{(t)} \xrightarrow{?} A^{t+1}$$



Q: Can we predict links that will form at time step $t+1$?

A: Many models exist, e.g. common neighbors, the Katz measure, etc.

Matrix Functions

- Powers of A:
  $(A^n)_{ij}$ counts the number of length-n walks between nodes i and j

- Matrix Functions, e.g. Katz measure:

$$f(A) = (I - \beta A)^{-1} = I + \beta A + \beta^2 A^2 + \beta^3 A^3 + \cdots$$

- Centrality: $f(A)_{ii}$, quantifies node i's "well-connectedness"
- Communicability: $f(A)_{ij}$, measures how easy it is for information to pass from node i to j
- Betweenness: quantifies influence of a node as information flows around a network, how communicability changes when a node is removed

$$\sum_{i \neq r} \sum_{j \neq i, j \neq r} \frac{f(A)_{ij} - f(A - E(r))_{ij}}{f(A)_{ij}}$$

Low-Rank Approximations

Truncated SVD:

$$A^i \approx V \Lambda^i V^\mathrm{T}$$

allows approximations of matrix functions:

$$\begin{aligned} f(A) &\approx Vf(\Lambda)V^T \\ (I - \beta A)^{-1} &\approx V(I - \beta\Lambda)^{-1}V^T \end{aligned}$$

BUT SVD has Major Drawback. For example, in LiveJournal graph:

- 3.8 million nodes and 65 million edges (avg. degree = 4.5)
- Requires about 1GB of memory to store adjacency matrix
- Rank-100 approximation requires about 5.7GB of memory

PROBLEM: SVD Does Not Preserve SPARSITY

Clustered Low Rank Approximation (CLRA)

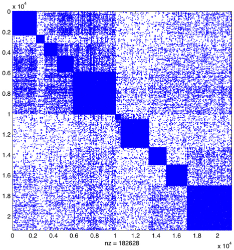
Input: An $m \times m$ adjacency matrix A of a graph, number of clusters c
Output: Clustered low rank approximation of A

1: Cluster the graph into c clusters
2: Compute a low rank approximation of each cluster

$$U_i S_i U_i^T \approx A_{ii}$$

3: Extend the cluster-wise approximations, into an approximation for the entire matrix A

$$S_{ij} = U_i^T A_{ij} U_j$$

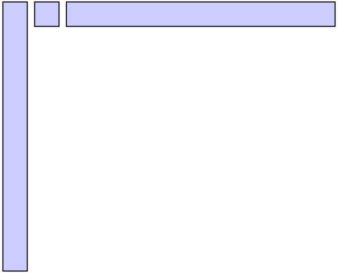


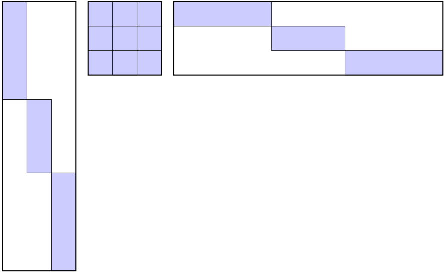
(Savas and Dhillon 2011)

Clustered Low Rank Approximation (CLRA)

Low rank: $A \approx U \Sigma U^{\mathsf{T}}$

Clustered low rank: $A \approx \begin{bmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{bmatrix}^{\mathsf{T}}$





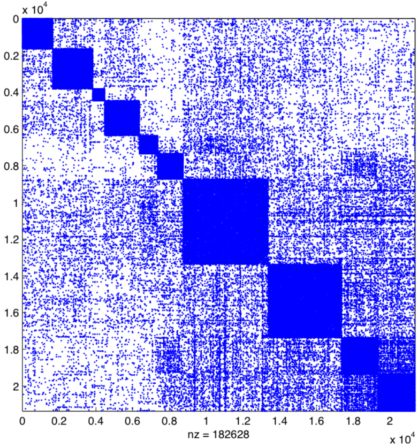
- Observe $\mathrm{diag}(U_1, U_2, \cdots, U_c)$ and has the same memory usage as U
- Matrix S captures the associations between clusters
- Experiments show that most of U is contained in $\mathrm{diag}(U_1, U_2, \cdots, U_c)$

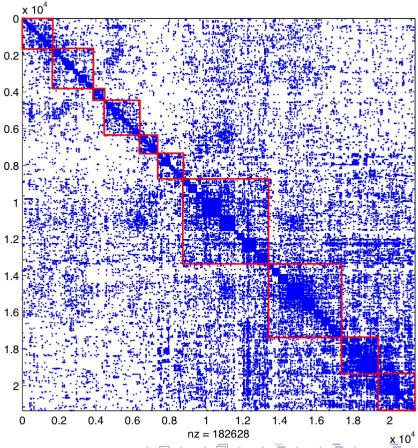
Proposed Method:
Multi-Scale Link Prediction

Hierarchical Clustering: arXiv data graph **condMat**

Networks exhibit a hierarchical structure

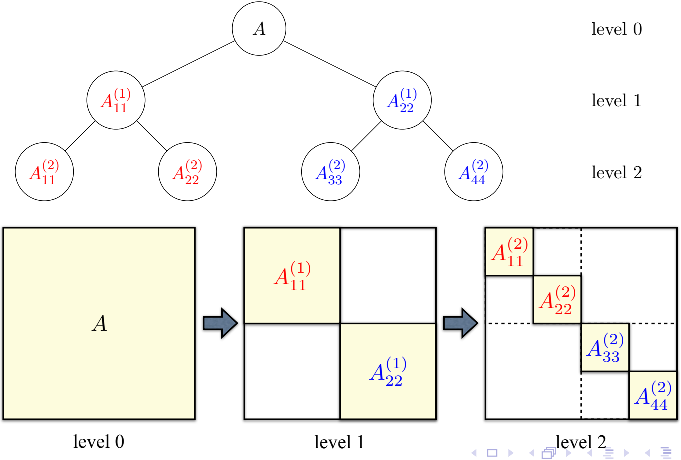
- Exploit hierarchical structure
- Predictions at multiple scales





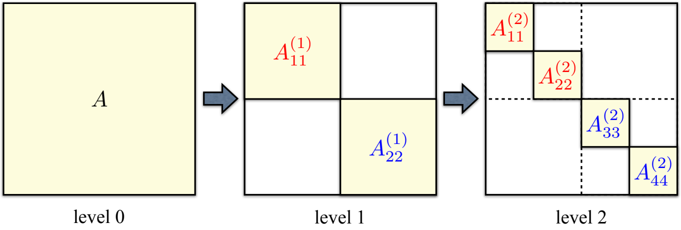
Hierarchical Clustering

- Top-down approach
- Fast multilevel graph clustering algorithms:
  GRACLUS (Dhillon et al. 2007) and METIS (Karypis and Kumar 1999)



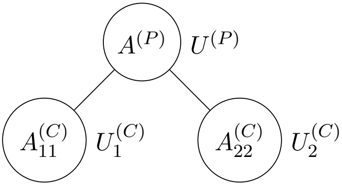
CLRA at Bottom Level

- Compute clustered low-rank approximation at bottom level



$$A \approx \begin{bmatrix} U_1^{(2)} & 0 & 0 & 0 \\ 0 & U_2^{(2)} & 0 & 0 \\ 0 & 0 & U_3^{(2)} & 0 \\ 0 & 0 & 0 & U_4^{(2)} \end{bmatrix} \begin{bmatrix} S_{11}^{(2)} & S_{12}^{(2)} & S_{13}^{(2)} & S_{14}^{(2)} \\ S_{21}^{(2)} & S_{22}^{(2)} & S_{23}^{(2)} & S_{24}^{(2)} \\ S_{31}^{(2)} & S_{32}^{(2)} & S_{33}^{(2)} & S_{34}^{(2)} \\ S_{41}^{(2)} & S_{42}^{(2)} & S_{43}^{(2)} & S_{44}^{(2)} \end{bmatrix} \begin{bmatrix} U_1^{(2)} & 0 & 0 & 0 \\ 0 & U_2^{(2)} & 0 & 0 \\ 0 & 0 & U_3^{(2)} & 0 \\ 0 & 0 & 0 & U_4^{(2)} \end{bmatrix}^T$$

Subspace Approximation



Q: How to get $U^{(P)}$?

A: SVD on $A^{(P)}$

$\longrightarrow$ Computationally expensive as cluster size increases

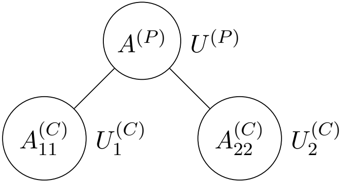
Subspaces between two adjacent levels should be close to each other:

$$U^{(P)} \text{ should be close to } \begin{bmatrix} U_1^{(C)} & 0 \\ 0 & U_2^{(C)} \end{bmatrix}$$

A: Use child cluster's subspace to compute parent cluster's subspace

$\longrightarrow$ Much faster than computing from scratch!

Tree-Structured Approximation of Subspace



1: Capture range space of $A^{(P)}$ by Y:

$$Y = A^{(P)} \begin{bmatrix} U_1^{(C)} & 0 \\ 0 & U_2^{(C)} \end{bmatrix}$$

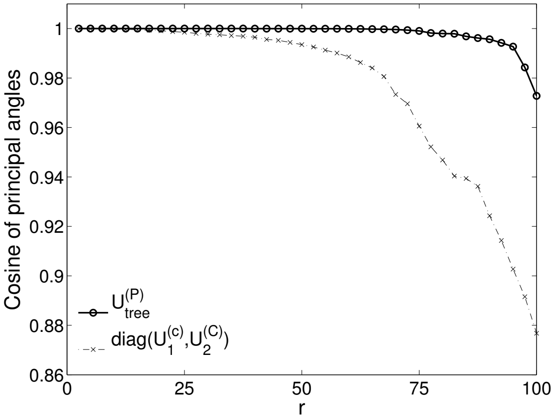
2: Compute Q as an orthonormal basis for Y

3: $A \approx Q(Q^T A Q)Q^T = QBQ^T = Q(V \Lambda V^T)Q^T$

4: $U^{(P)} = QV$

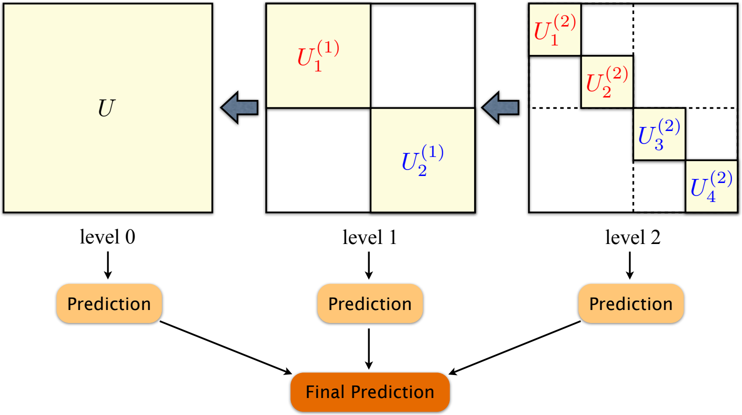
Closeness to SVD: Principal Angles

- Cosine of principal angles between $U_{eig}^{(P)}$ on Flickr dataset



Multi-Scale Link Prediction (MSLP)

- Combine predictions at each level to make final predictions



Multi-Scale Link Prediction (MSLP)

Input: A, # of levels ℓ, # of clusters c, target rank r

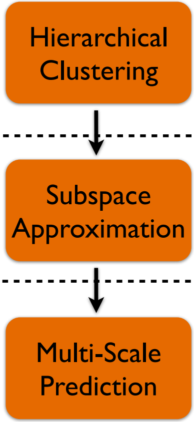
Output: top-k predictions

for $i = 0$ **to** ℓ **do**
    for $j = 1$ **to** c^i **do**
        Cluster $A_{jj}^{(i)}$ into c clusters (e.g. Graclus)
    end for
end for

Compute $U^{(\ell)}$, $S^{(\ell)}$ using CLRA
for $i = \ell - 1$ **to** 0 **do**
    Compute $U^{(i)}$ using child clusters and $S^{(i)} = {U^{(i)}}^T A U^{(i)}$
end for

---

for $i = \ell$ **to** 0 **do**
    $K_i = f(A^{(i)}) = U^{(i)} f(S^{(i)}) {U^{(i)}}^T$
end for
$P = w_0 K_0 + w_1 K_1 + \ldots + w_\ell K_\ell$
return top-k predictions according to P



Experimental Results

- Karate club: 34 nodes, 78 edges
- Leave-one-out experiment

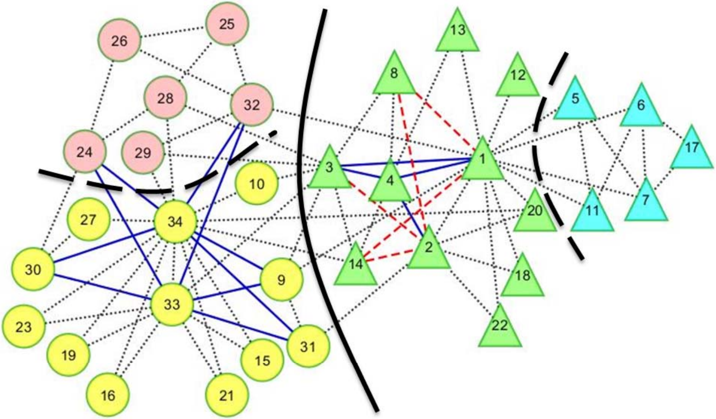
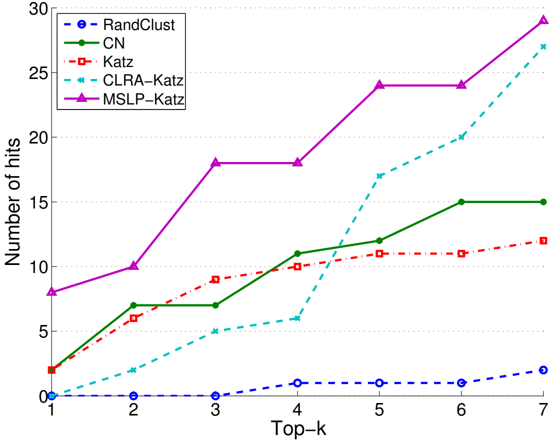


Figure: Top-3 hits

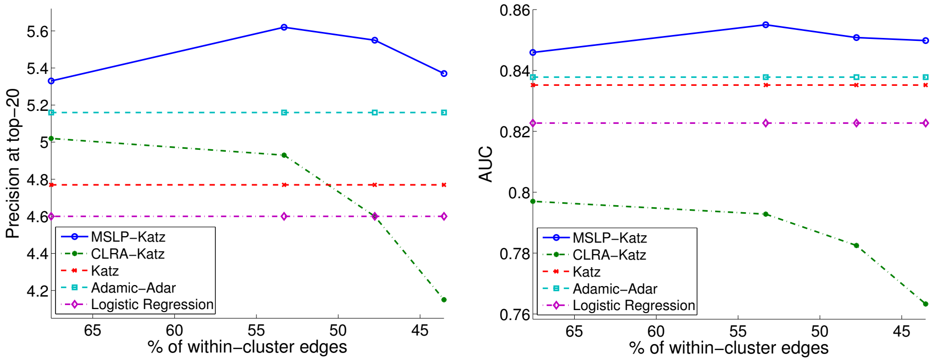
Case Study: **Karate Club**

- Karate club: 34 nodes, 78 edges
- Leave-one-out experiment



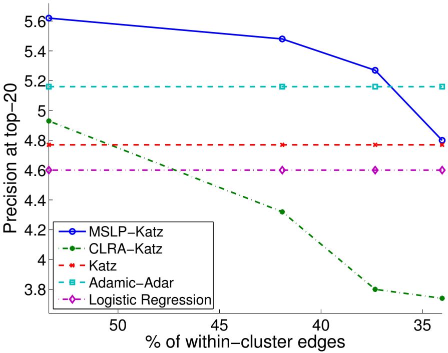
Clustering Structure

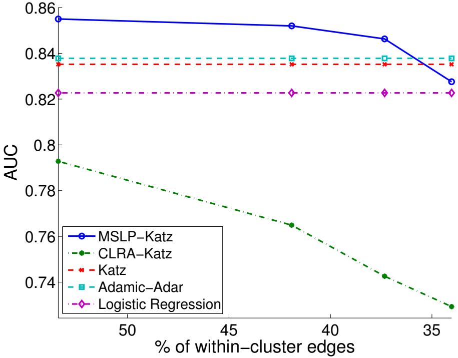
- Epinions dataset: 32,223 nodes, 684,026 edges
- $c = 2$, $\ell = 2, 3, 4, 5$, $r = 20$



Clustering Structure

- Epinions dataset: 32,223 nodes, 684,026 edges
- $c = 2, 3, 4, 5,\ \ell = 3,\ r = 20$





Large Datasets

- Three large datasets: Flickr(`FL`), LiveJournal(`LJ`), MySpace(`MS`)
- About 2M nodes, 40M to 90M edges
- $c = 2$, $\ell = 5$, $r = 100$

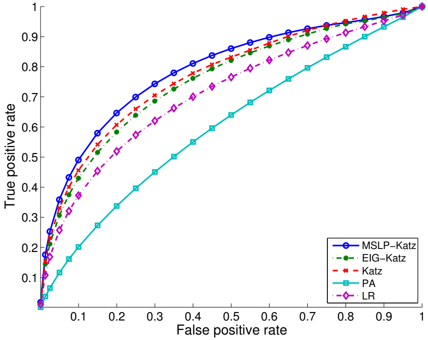
Table: Precision at top-100

Method	FL	LJ	MS
PA(Preferential Attachment)	1.02	1.32	4.57
AA(Adamic-Adar)	7.29	5.93	7.44
RWR(Random Walk w/ Restart)	5.49	3.46	1.30
LR(Logistic Regression)	2.54	2.23	4.95
CN(Common Neighbors)	7.08	5.94	7.18
CLRA-CN	6.91	5.21	6.88
MSLP-CN	7.03	5.59	7.05
Katz	7.17	5.86	6.18
CLRA-Katz	12.13	6.11	7.64
MSLP-Katz	**13.34**	**6.72**	**8.38**

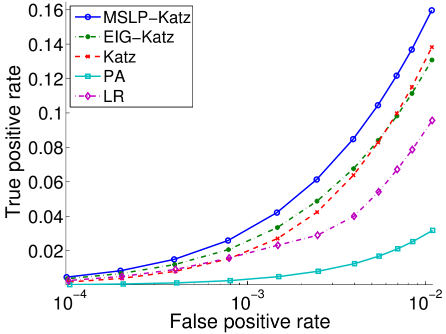
Large Datasets

- Three large datasets: Flickr(`FL`), LiveJournal(`LJ`), MySpace(`MS`)
- About 2M nodes, 40M to 90M edges
- $c = 2$, $\ell = 5$, $r = 100$

- Full ROC curve for `LJ`



- ROC curve on low FPR region



Conclusions and Future Work

- Multi-Scale Link Prediction
  - Combine predictions from multiple scales via hierarchical clustering
  - Fast and scalable tree-structured subspace approximation method
  - Superior performance on real-world social networks
  - Robust to different clustering structures
- Future Work
  - Learn the weights — different levels may have different predictive performance
  - Coping with unbalanced hierarchical clustering structures
  - Potential for parallelization

References

[1] D. Liben-Nowell and J. Kleinberg. *The Link Prediction Problem for Social Networks*. CIKM, 2003.

[2] I. S. Dhillon, Y. Guan and B. Kulis. *Weighted Graph Cuts without Eigenvectors: A Multilevel Approach*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007.

[3] E. Estrada and D. J. Higham. *Network Properties Revealed through Matrix Functions*. SIAM Review, 2010.

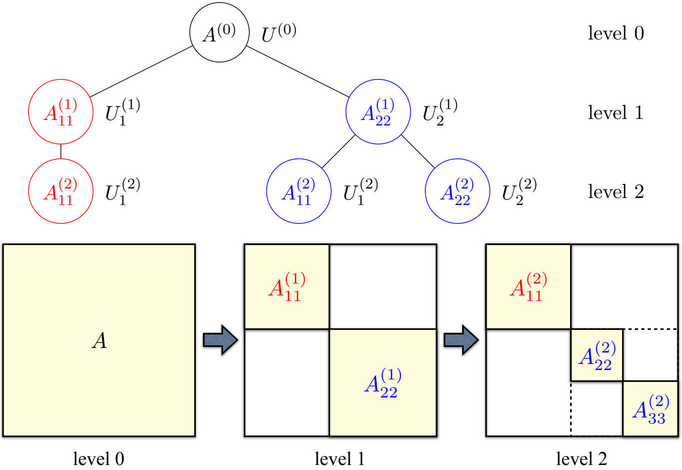
[4] B. Savas and I. S. Dhillon. *Clustered Low Rank Approximation of Graphs in Information Science Applications*. SDM, 2011.

[5] H. Song, B. Savas, T. Cho, V. Dave, Z. Lu, I. S. Dhillon, Y. Zhang and L. Qiu. *Clustered Embedding of Massive Social Networks*. SIGMETRICS, 2012.

[6] X. Sui, T. Lee, J. Whang, B. Savas, S. Jain, K. Pingali and I. S. Dhillon. *Parallel Clustered Low-rank Approximation of Graphs and Its Application to Link Prediction*. LCPC, 2012.

Unbalanced Tree

- Divide the graph into smaller subgraphs via hierarchical clustering
- $A_{11}^{(1)} = A_{11}^{(2)}$



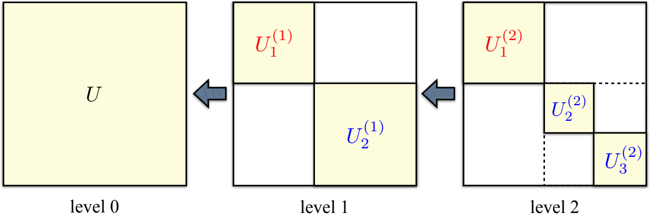
Clustered Low Rank Approximation

- Compute clustered low-rank approximation at bottom level

$$A \approx \begin{bmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} U_1 & 0 & 0 \\ 0 & U_2 & 0 \\ 0 & 0 & U_3 \end{bmatrix}^{\mathsf{T}}$$

where $S_{ij} = U_i^{\mathsf{T}} A_{ij} U_j$ for $i \neq j$

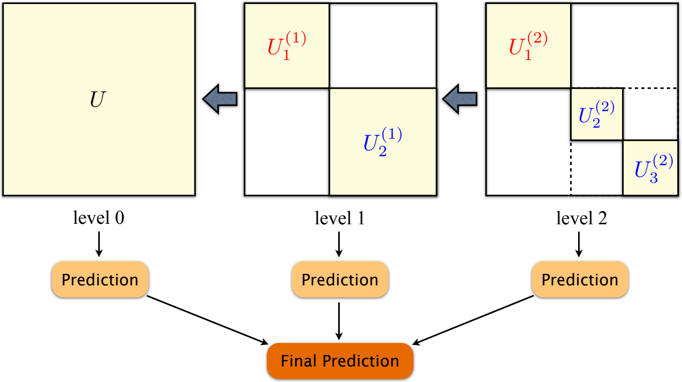
- Obtain low-rank approximation for higher levels using lower level's approximation



Much faster than computing from scratch!

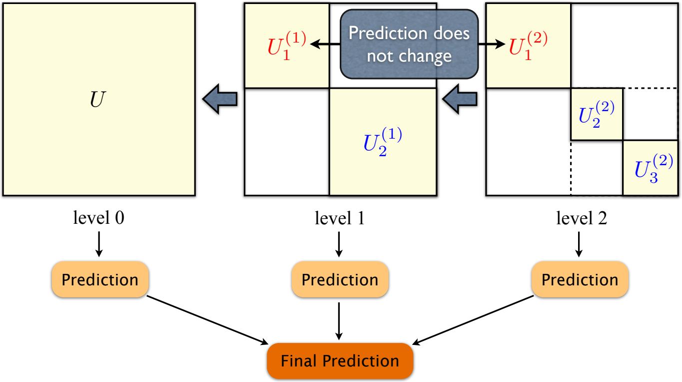
Unbalanced Tree

- Combine predictions at each level to make final predictions



Unbalanced Tree

- Combine predictions at each level to make final predictions



Unbalanced Tree

- Combine predictions at each level to make final predictions