# Lagrangian relaxation for tree-structured covariance estimation

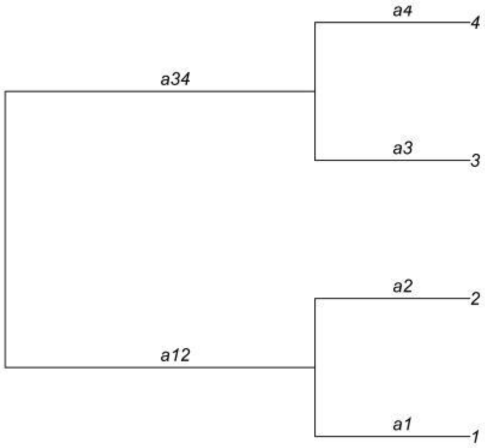
Xiyang Dai, Zebao Gao, Hao Zhou

# Why use a tree structure covariance?

- In many fields, large numbers of objects need to be categorized, a tree structure covariance is a nature way to model these categorizes, find interrelationships between categorizes and visualize them.

- Especially true for Bioinformatic problems:
  - Bioinformatic problems usually have
    - Large feature dimensions
    - Large number of samples
    - Relatively small differences
  - Traditional distance based measurements suffer from modeling large scale data
  - Tree structure covariance can provide an <u>intuitive</u> way to understand the similarity/differences from the data

# How to approach a tree structure covariance?




$$\Longleftrightarrow \begin{pmatrix} a_{12}+a_1 & a_{12} & 0 & 0 \\ a_{12} & a_{12}+a_2 & 0 & 0 \\ 0 & 0 & a_{34}+a_3 & a_{34} \\ 0 & 0 & a_{34} & a_{34}+a_4 \end{pmatrix}$$

- Let $d = [a_{12}\ a_{34}\ a_1\ a_2\ a_3\ a_4]^T$ be a column vector containing the branch lengths of the tree and $D = diag(d)$.
- Define a basis matrix $V$ such that each row is associated with a node in the tree and each column contains a set of binary variables corresponding node in the tree is on the path to the leaf associated with that row.
- The covariance matrix $B = VDV^T$

# Tree-structured Covariance Matrix

- Partition Property
  - The basis matrix *V* (size p x 2p-1, where p is the #leaves) models the Partition Property if it satisfies
    - V contains one column of all ones (root)
    - For every column *w* in *V* with more than one non-zero entry, it contains exactly two columns *u* and *v* such that *u* + *v* = *w*. (binary tree partition property)

(Every column *w* in *V* with one non-zero entry is a leave)

# Tree-structured Covariance Matrix

- Get t-s cov matrix B from basis matrix V
  - Matrix B is a t-s cov matrix ⇔ $B=VDV^T$

- Characteristics
  - $B=VDV^T$ generated by a single basis matrix V is convex
  - The set of all t-s cov matrices is not convex

$$V_1 = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix},\ V_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

If B is a convex combination of $B_1$ and $B_2$
we will have $B_{12}$ != 0 and $B_{23}$ != 0 but $B_{13}$ = 0

# Projection by Mixed-Integer Programming

- Problem
  - Estimating a t-s cov matrix when tree topology is unknown
  - Combinational optimization over set of B → Mixed-Integer Programs
  - MIP algorithms are well-studied and tools/libs available

- Constraints to minimize objective functions
  - $B_{ij} >= 0$ for all i and j
  - $B_{ii} >= B_{ij}$ for all i and j
  - $B_{ij} >= \min(B_{ik}, B_{jk})$ for all i, j and k
    - Rewrite the third constraint for each distinct triplet i > j > k
    - $B_{ij} >= B_{ik} = B_{jk}$
    - $B_{ik} >= B_{ij} = B_{jk}$
    - $B_{jk} >= B_{ij} = B_{ik}$

# Projection by Mixed-Integer Programming

- MIP formulation to the Problem:
  - Given a sample covariance matrix S ⇒ nearest t-s cov matrix B in norm ||•||

- Objective functions
  - Frobenius norm:
    - min **$||S - B||_F$**
    - or min **$f(b) = \frac{1}{2} b^T b - s^T b$** (where b and s are vectorization of B and S)

$$\|B\|_F = \sqrt{\sum_{ij} B_{ij}^2}$$

  - sum-absolute-value (sav) norm:
    - min $f(b) = ||s - b||_1$

$$\|B\|_{\text{sav}} = \sum_{ij} |B_{ij}|.$$

# Lagrangian Relaxation

Solve constrained optimization problem.

$$y^* = \arg\max_{y \in Y} h(y)$$
$$s.t. \quad Ay = b$$

# Lagrangian Relaxation

Solving its dual problem

$$L(u, y) = h(y) + u(Ay - b)$$

$$L(u) = \max_{y \in Y'} L(u, y)$$

$$\min_{u \in R^p} L(u)$$

Property

$$L(u) \geq \max_{y \in Y} h(y)$$

# Lagrangian Relaxation

Subgradient algorithm

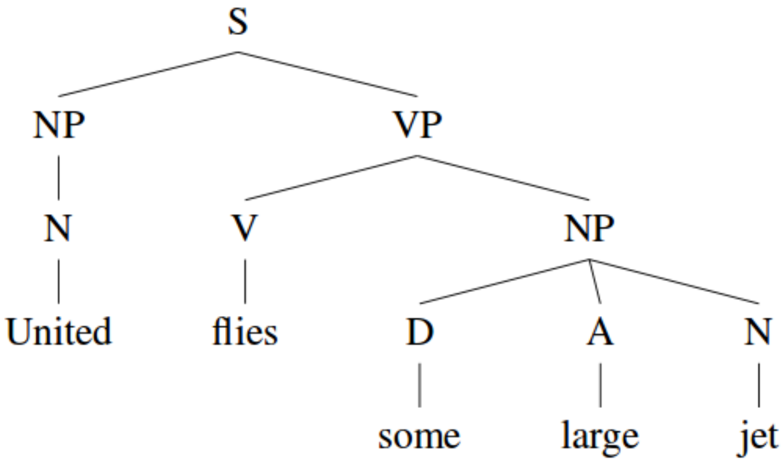
$$\min_{u \in R^p} \max_{y \in Y'} L(u, y)$$

$$\text{i.e. } \min_{u \in R^p} \max_{y \in Y'} h(y) + u(Ay - b)$$

Solving above problem w.r.t. u, y alternatively.

# Dual Decomposition an NLP example

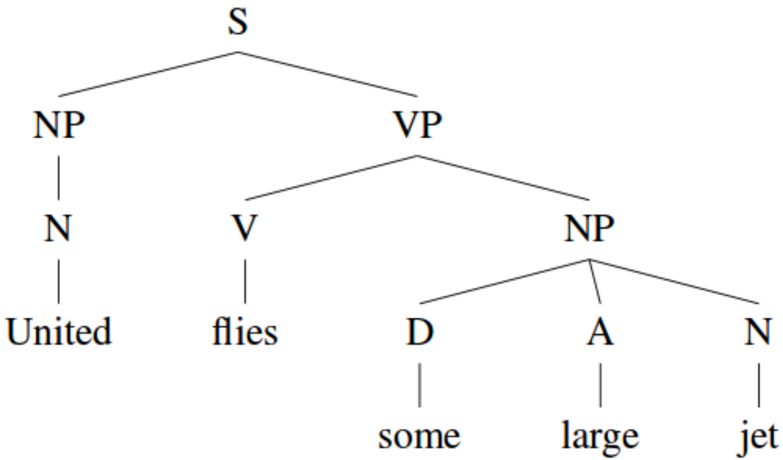
Parse tree




$$y^* = \arg\max_{y \in Y} h(y)$$
$$h(y) = f(y) + g(l(y))$$

# Dual Decomposition an NLP example

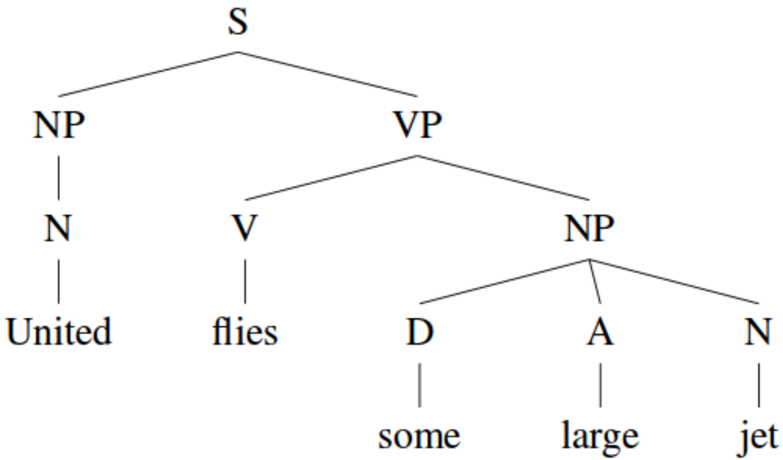
Parse tree




$$\arg \max_{y \in Y, z \in Z} f(y) + g(z)$$
$$s.t. \quad y(i,t) = z(i,t)$$

# Dual Decomposition an NLP example

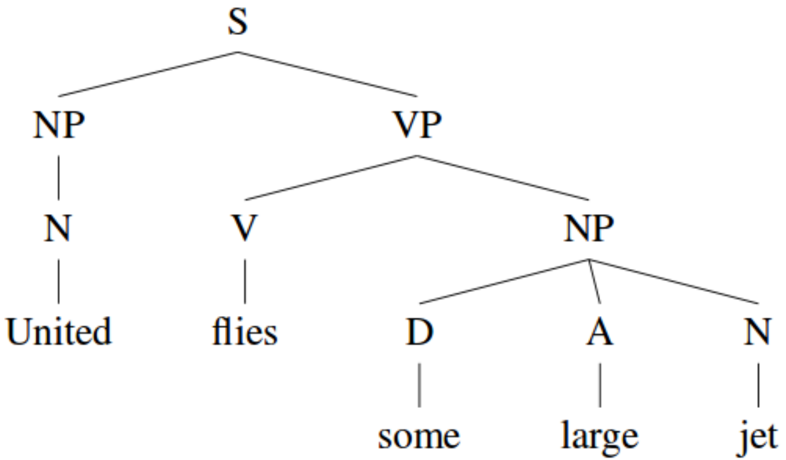
Parse tree




$$\arg \max_{y \in Y, z \in Z} f(y) + g(z) + \sum_{i,t} u(i,t)(y(i,t) - z(i,t))$$

# Dual Decomposition an NLP example

Parse tree




$$\arg \max_{y \in Y, z \in Z} \left(f(y) + \sum_{i,t} u(i,j) y(i,t)\right)$$
$$+ \left(g(z) - \sum_{i,t} u(i,t) z(i,t)\right)$$

# In this project...

- Try to reformulate the tree structured covariance problem

- Using Lagrangian relaxation to solve the problem

# References

[1] Héctor Corrada Bravo, Stephen J. Wright, Kevin H. Eng, Sunduz Keles, and Grace Wahba. Estimating tree-structured covariance matrices via mixed-integer programming. In AISTATS, JMLR Proceedings, pages 41–48, 2009.

[2] Alexander M. Rush and Michael Collins. A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing. Journal of Artificial Intelligence Research, 2012.