School of Computer Science
Carnegie Mellon

Some Challenging Machine Learning Problems in Computational Biology:

Time-Varying Networks Inference and Sparse Structured Input-Out Learning

Eric P. Xing

School of Computer Science

Carnegie Mellon University

Acknowledgements: Amr Ahmed, Fan Guo, Steve Hanneke, Seyoung Kim. Song Le, Jun Zhu

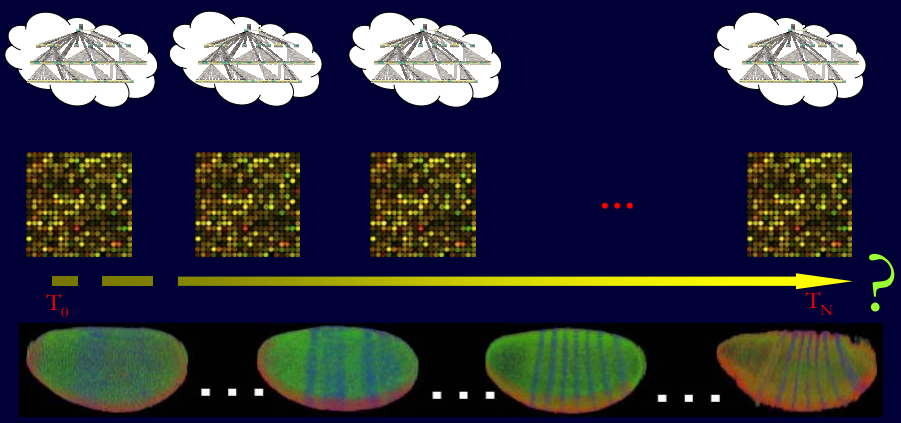
School of Computer Science
Carnegie Mellon

Biological Data Analysis

- Dynamic, noisy, heterogeneous, high-dimensional data

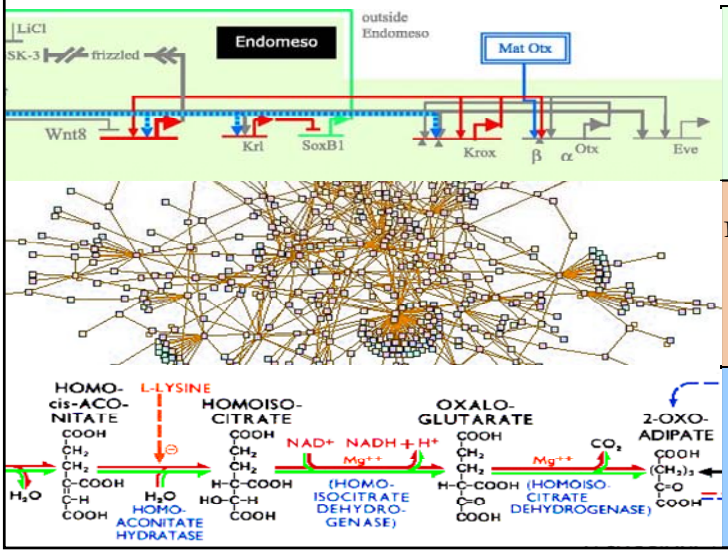
- "High-resolution" inference
- Parsimonious
- Scalability
- Stability
- Sample complexity
- Confidence bound

Inferring Time-Varying Networks



School of Computer Science
Carnegie Mellon

Regulation of cell response to stimuli is paramount, but we can usually only measure (or compute) steady-state interactions



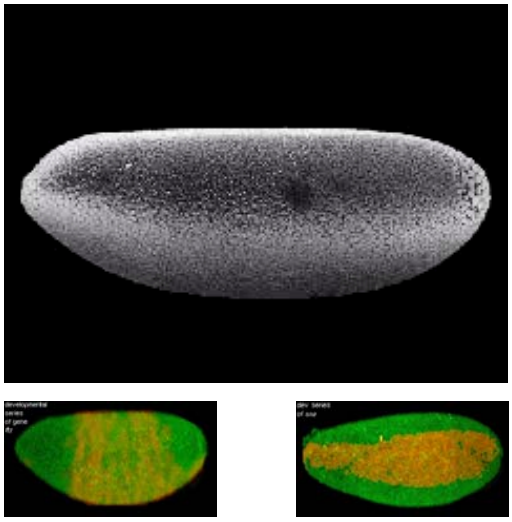
Transcriptional interactions
- Chromatin IP
- Microarrays

Protein—protein interactions
- Protein coIP
- Yeast two-hybrid

Biochemical reactions
- Metabolic flux measurements

School of Computer Science
Carnegie Mellon

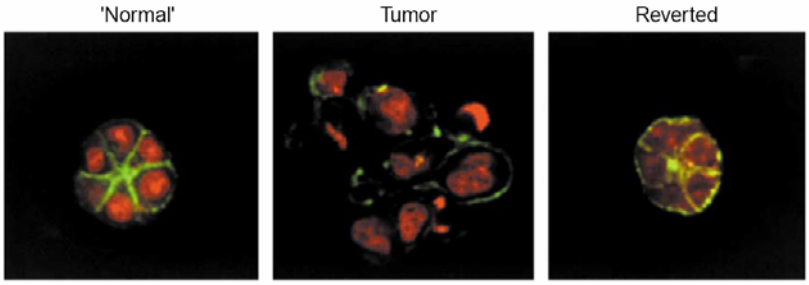
Biological regulations may be transient (in time and space) …



5

School of Computer Science
Carnegie Mellon

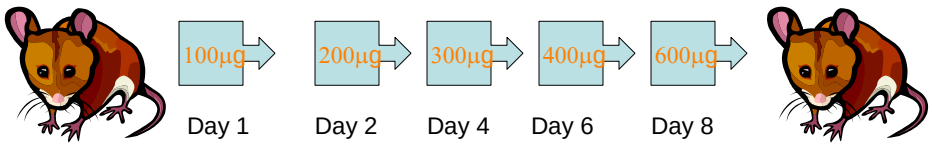
Example II: Breast Cancer Progression and Reversal in Organotypic Culture

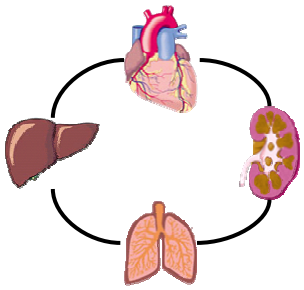


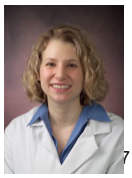
Dr. Mina Bissell, Berkeley

School of Computer Science
Carnegie Mellon

Example III: Inflammatory Response in Endotoxinated Mice





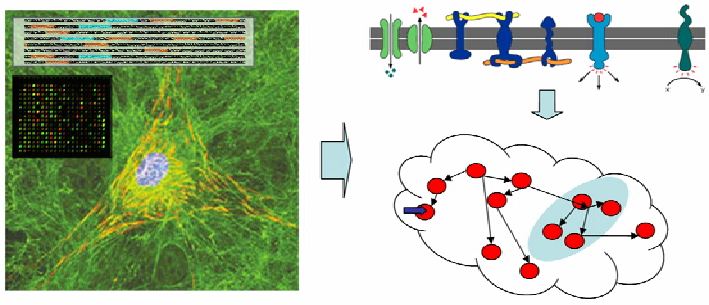


Judie Howrylak MD, UPMC

School of Computer Science
Carnegie Mellon

The Big-Picture Questions

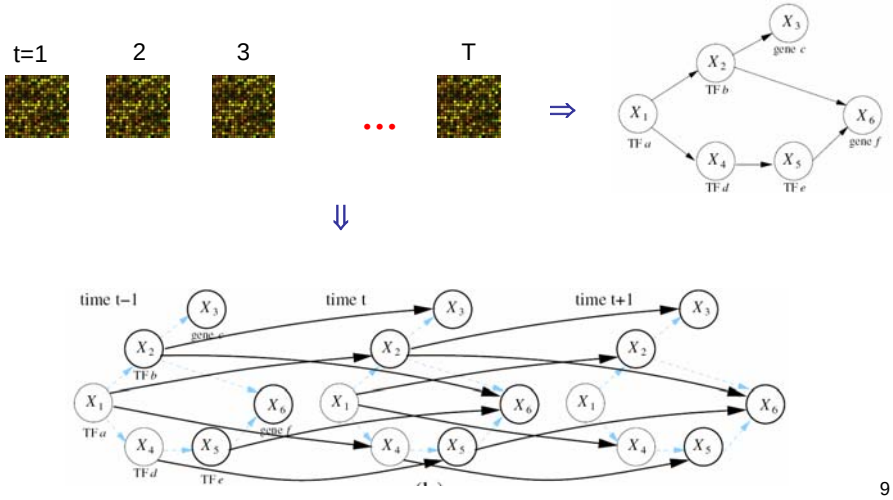
- What pathway is active under certain extra-cellular stimuli or a certain point of a dynamic process?



- How does the network response to environmental perturbation and biomolecular/genetic therapy?

School of Computer Science
Carnegie Mellon

Current Practice …

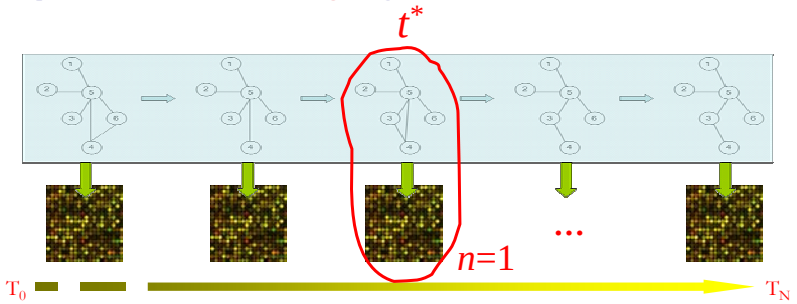


9

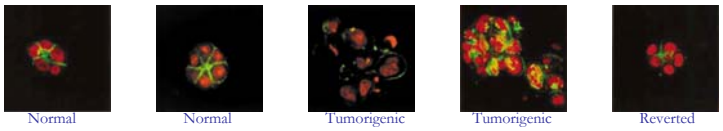
ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

Reverse engineer temporal/spatial-specific "rewiring" gene networks



EGFR-induced progression/reversion of breast epithelial cells



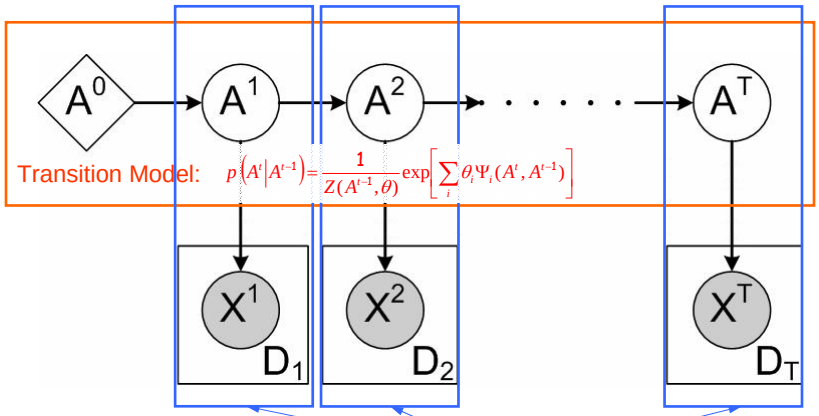
10

ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

Modeling Time-Varying Graphs

- The temporal exponential graph models *(Fan et al. ICML 2007)*



School of Computer Science
Carnegie Mellon

"Dynamic" Potentials

$$P\left(A^t \middle| A^{t-1}\right) = \exp\left\{\theta \cdot \Psi\left(A^t, A^{t-1}\right) - \ln Z\left(\theta, A^{t-1}\right)\right\}$$

- "Continuity": $\Psi_1\left(A^t, A^{t-1}\right) = \sum_{ij}\left(A_{ij}^t A_{ij}^{t-1} + \left(1 - A_{ij}^t\right)\left(1 - A_{ij}^{t-1}\right)\right)$
- "Reciprocity": $\Psi_2\left(A^t, A^{t-1}\right) = \sum_{ij} A_{ij}^t A_{ji}^{t-1}$
- "Transitivity": $\Psi_3\left(A^t, A^{t-1}\right) = \frac{\sum_{ijk} A_{ij}^t A_{ik}^{t-1} A_{kj}^{t-1}}{\sum_{ijk} A_{ik}^{t-1} A_{kj}^{t-1}}$
- "Density": $\Psi_4\left(A^t, A^{t-1}\right) = \sum_{ij} A_{ij}^t$

School of Computer Science
Carnegie Mellon

An idea for specifying a model

- A network might be decomposable into different types of "motifs" (e.g., "hub & spokes", "k-clique", "triangle",…).
- Write the potential functions to encode your understanding about how each motif evolves.
- It's nice because we can "plug in" our intuition about the data.

School of Computer Science
Carnegie Mellon

Maximum Likelihood Estimation

- Approximate MLE by MCMC (*Z* intractable)

$$\ln P\left(A^T, A^{T-1}, ..., A^2 \middle| A^1\right) = \theta \cdot \sum_{t=2}^{T} \Psi\left(A^t, A^{t-1}\right) - \sum_{t=2}^{T} \ln Z\left(\theta, A^{t-1}\right)$$

$$\nabla \ln P\left(A^T, A^{T-1}, ..., A^2 \middle| A^1\right) = \sum_{t=2}^{T} \Psi\left(A^t, A^{t-1}\right) - \sum_{t=2}^{T} E_\theta\left[\Psi\left(A, A^{t-1}\right) \middle| A^{t-1}\right]$$

- Use gradient ascent, using MCMC to estimate the expectation on each iteration (as in ERGM).

School of Computer Science
Carnegie Mellon

Estimation Toy Example

- Generate a series of 10 networks from the example.
- True model has θ_1=0, θ_2=5, θ_3=0, θ_4=-20 (i.e., reciprocity and density only)
- Estimated parameters:
  θ_1=-0.5, θ_2=4.2, θ_3=-0.08, θ_4=-20.2

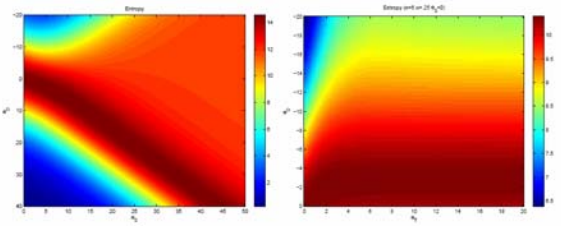
15

School of Computer Science
Carnegie Mellon

A tERGM is non-degenerate

- **Theorem 1**: when the transition distribution factors over the edges, a tERGM is non-degenerate:

$$H(A^t) \geq \sum_{i,j} H(A^t_{i,j}|A^{t-1}) \geq p \ln \frac{1}{p} + (1-p) \ln \frac{1}{(1-p)}$$



- → Maximum likelihood estimator exists!
  (actually should not be taken for granted for arbitrary models, be careful!)

16

School of Computer Science
Carnegie Mellon

What's it good for?

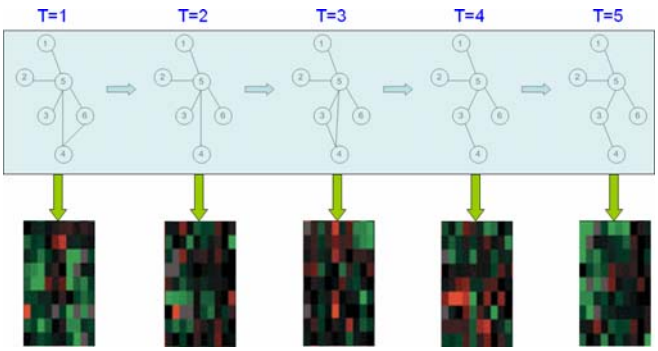
- Hypothesis Testing
- Data Exploration (e.g., node-classification)
- Foundation for Learning
  - Current approaches:
    - Time invariant structures
    - Global optimality?
    - Consistency guarantee?

17

School of Computer Science
Carnegie Mellon

Inferring Rewiring Biological Networks

- Networks rewire over discrete time-steps

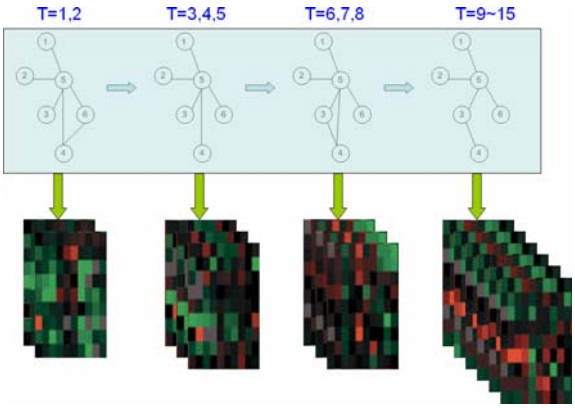


18

School of Computer Science
Carnegie Mellon

Inferring Rewiring Biological Networks

- Networks rewire over epochs

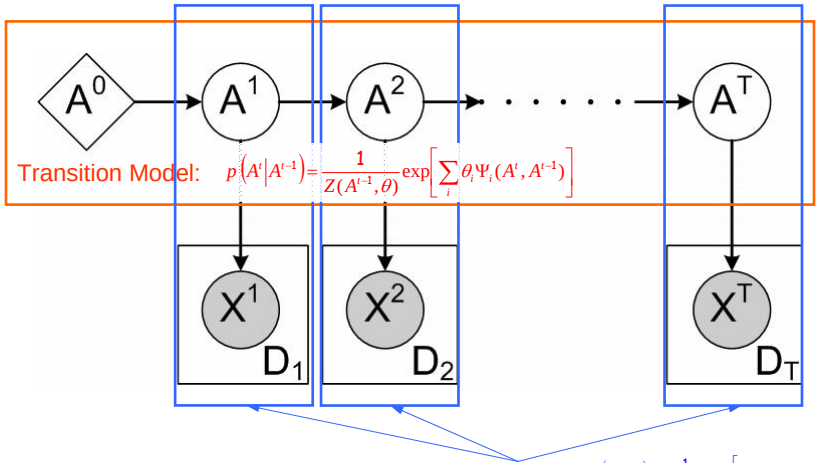


19

School of Computer Science
Carnegie Mellon

Modeling Time-Varying Graphs

- The temporal exponential graph models *(Fan et al. ICML 2007)*



20

School of Computer Science
Carnegie Mellon

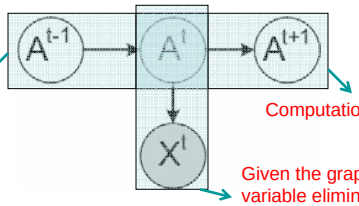
Inference (1)

P(Network|Data) ?

- Gibbs sampling:
  - Need to evaluate the log-odds

$$\mu_{ij}^t = \log \frac{P(A_{ij}^t = 1 | A_{t-1}, A_{-ij}^t, A^{t+1}, x^t)}{P(A_{ij}^t = 0 | A_{t-1}, A_{-ij}^t, A^{t+1}, x^t)}$$
$$= \log \frac{P(A_{-ij}^t, A_{ij}^t = 1 | A^{t-1})}{P(A_{-ij}^t, A_{ij}^t = 0 | A^{t-1})} + \log \frac{P(A^{t+1} | A_{-ij}^t, A_{ij}^t = 1)}{P(A^{t+1} | A_{-ij}^t, A_{ij}^t = 0)} + \log \frac{P(x^t | A_{-ij}^t, A_{ij}^t = 1)}{P(x^t | A_{-ij}^t, A_{ij}^t = 0)}$$

  - Difficulty: Evaluate the ratio of Partition function $Z(A')=\Sigma_A \exp(\theta\Phi(A,A'))$
  - So far scale to ~20 genes



Straightforward -- tractable transition model; the partition function is the product of per edge terms

Computation is non-trivial

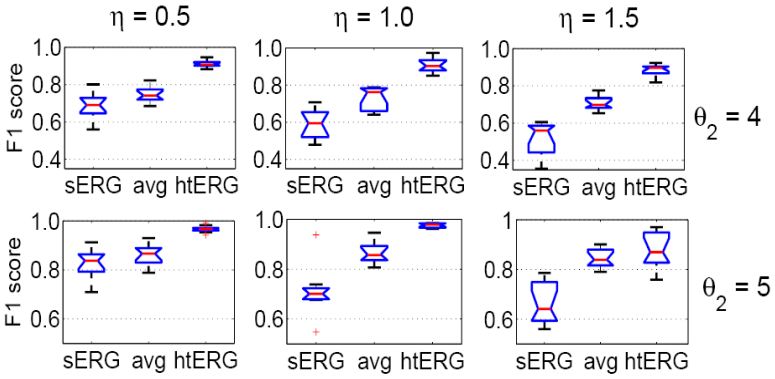
Given the graphical structure, run variable elimination algorithms, works well only for small graphs

School of Computer Science
Carnegie Mellon

Results on Simulated Data

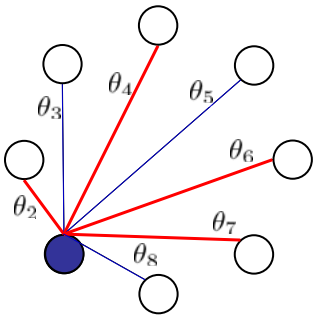
- F1 scores on different parameter settings (varying θ_2, η)

($\theta_1 = -0.5, \theta_3 = 4, D = 5$, 100k iterations of Gibbs sampling, 10 repetitions)



School of Computer Science
Carnegie Mellon

Graph Regression



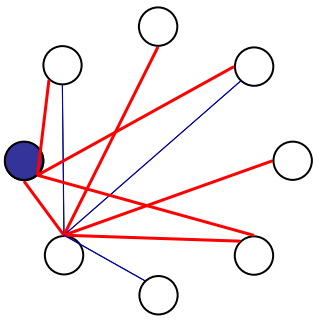
Lasso:

$$\hat{\theta} = \arg\min_{\theta} \sum_{t=1}^{T} l(\theta) + \lambda_1 \| \theta \|_1$$

23

School of Computer Science
Carnegie Mellon

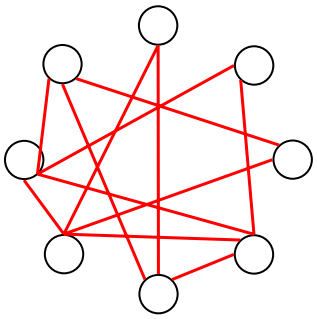
Graph Regression



24

School of Computer Science
Carnegie Mellon

Graph Regression



School of Computer Science
Carnegie Mellon

Inference II

- **TESLA**: Temporally Smoothed L_1-regularized logistic regression

$$\hat{\theta}_i^1, \ldots, \hat{\theta}_i^T = \arg\min_{\theta_i^1,\ldots,\theta_i^T} \sum_{t=1}^{T} l_{avg}(\theta_i^t) + \lambda_1 \sum_{t=1}^{T} \| \theta_{-i}^t \|_1 + \lambda_2 \sum_{t=2}^{T} \| \theta_i^t - \theta_i^{t-1} \|_q^q,$$

$$\text{where } l_{avg}(\theta_{\mathbf{i}}^{\mathbf{t}}) = \frac{1}{N^t} \sum_{d=1}^{N^t} \log P\left(x_{d,i}^t | \mathbf{x}_{\mathbf{d},-\mathbf{i}}^{\mathbf{t}}, \theta_{\mathbf{i}}^{\mathbf{t}}\right).$$

- Constrained convex optimization
  - Now scale to ~5000 nodes, how about 20K+ ?

School of Computer Science
Carnegie Mellon

Inference III

- Kernel Weighting (Kolar and Xing, 2008):

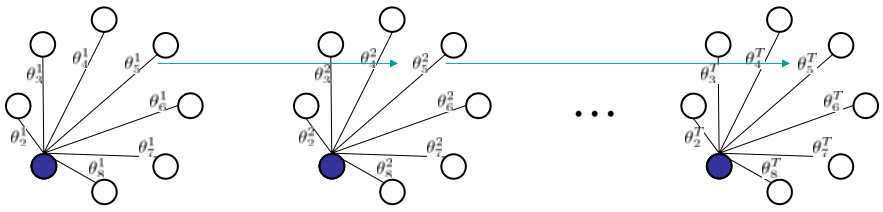
$$\hat{\theta}_i^1, \ldots, \hat{\theta}_i^T = \arg\min_{\theta_i^1,\ldots,\theta_i^T} \sum_{t=1}^{T} l_w(\theta_i^t) + \lambda_1 \sum_{t=1}^{T} \| \theta_{-i}^t \|_1$$

$$\text{where } l_w(\theta_i^t) = \sum_{t'=1}^{T} w(\mathbf{x}^{t'}; \mathbf{x}^t) \log P\left(x_i^{t'} | \mathbf{x}_{-i}^{t'}, \theta_i^t\right).$$

- Constrained convex optimization
  - Could scale to ~10^4 genes, but under stronger smoothness assumptions

School of Computer Science
Carnegie Mellon

Temporally Smoothed Graph Regression



TESLA:

$$\min_{\substack{\theta_i^1,\ldots,\theta_i^T \\ \mathbf{u}_i^1,\ldots,\mathbf{u}_i^T;\mathbf{v}_i^2,\ldots,\mathbf{v}_i^T}} \sum_{t=1}^{T} \ell(\mathbf{x}^t; \theta_i^t) + \lambda_1 \sum_{t=1}^{T} \mathbf{1}'\mathbf{u}_i^t + \lambda_2 \sum_{t=2}^{T} \mathbf{1}'\mathbf{v}_i^t$$

$$\text{s.t.} \quad -u_{i,j}^t \le \theta_{i,j}^t \le u_{i,j}^t, \ t = 1, \ldots, T, \forall j \in V \setminus i,$$

$$\text{s.t.} \quad -v_{i,j}^t \le \theta_{i,j}^t - \theta_{i,j}^{t-1} \le v_{i,j}^t, \ t = 2, \ldots, T, \forall j \in V \setminus i;$$

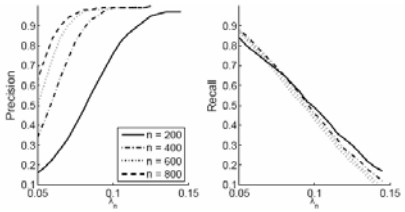
ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

Consistency

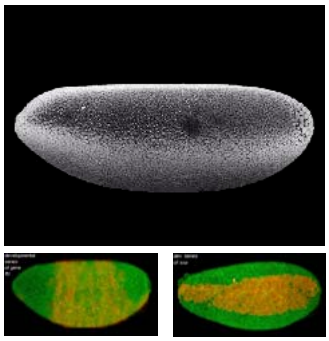
- **Theorem 2**: for the kernel weighting method, under certain verifiable conditions (omitted here for simplicity):

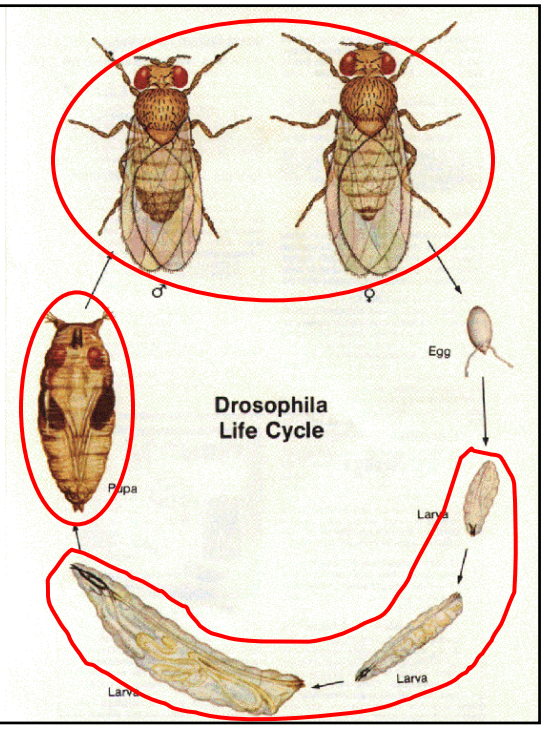
$$\mathbb{P}\left[\hat{G}(\lambda_n, t_0) \neq G^{t_0}\right] = \mathcal{O}\left(\exp\left(-C\frac{nh_n}{s_n^3} + C'\log p\right)\right) \to 0$$



- Consistency for TESLA is yet to be proven! (very hard!)

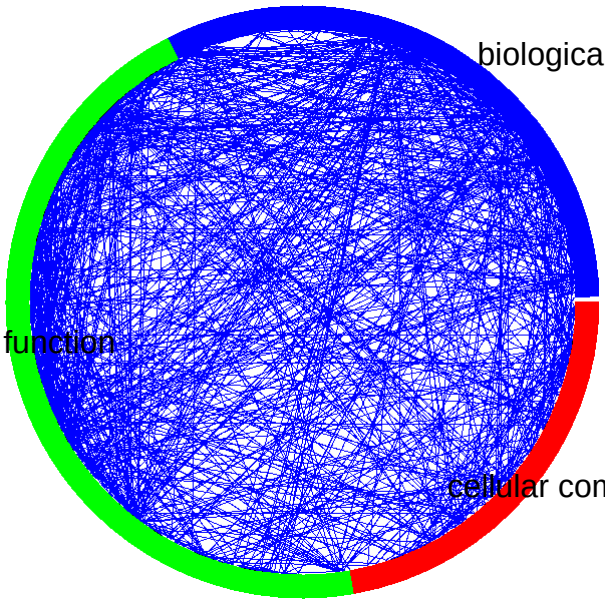
School of Computer Science
Carnegie Mellon





ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

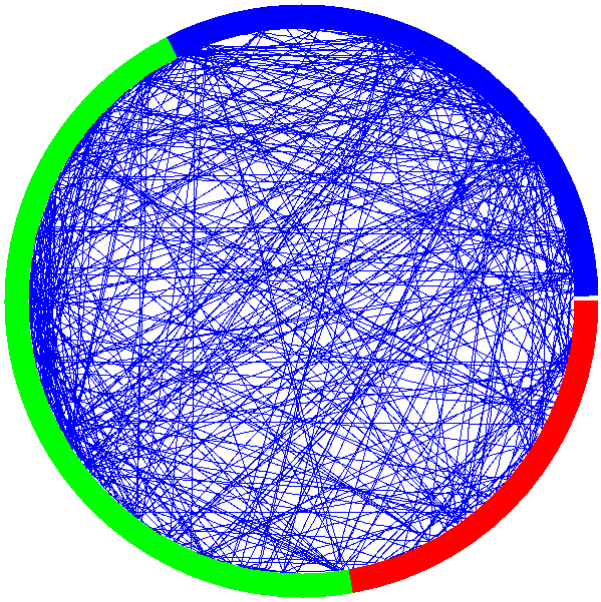


T=1

31

ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

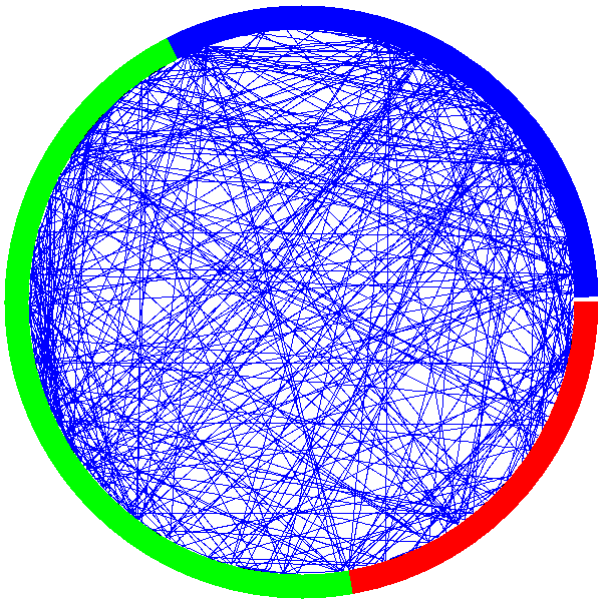


T=2

32

ML Lunch @ CMU, 11/ 24, 2008

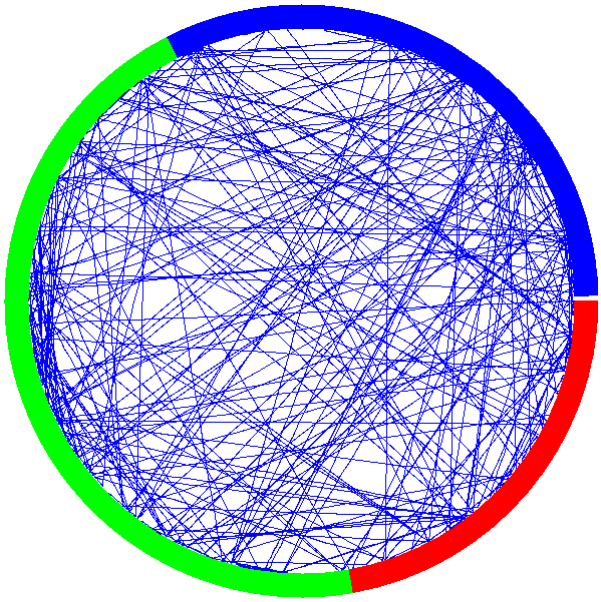
School of Computer Science
Carnegie Mellon



T=3

33

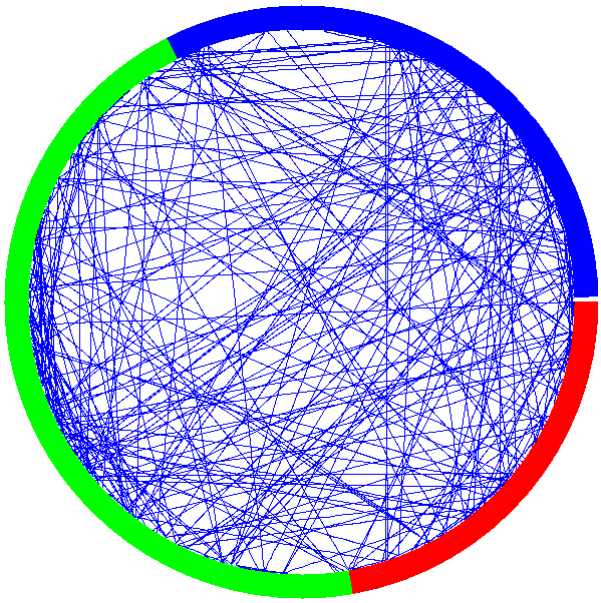
School of Computer Science
Carnegie Mellon



T=4

34

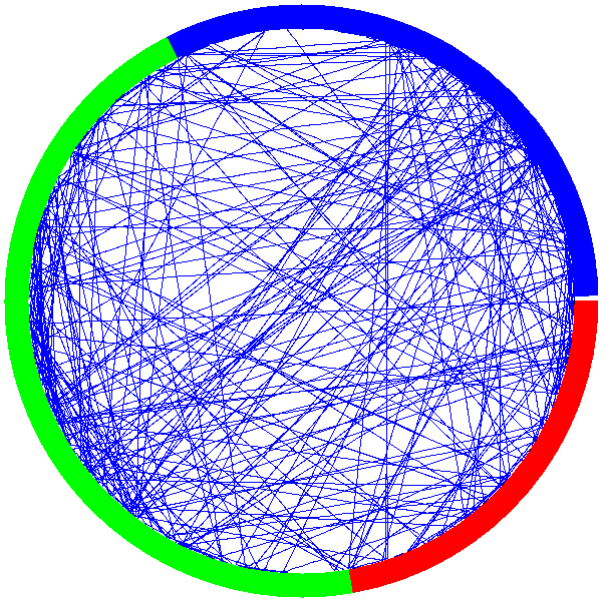
School of Computer Science
Carnegie Mellon



T=5

35

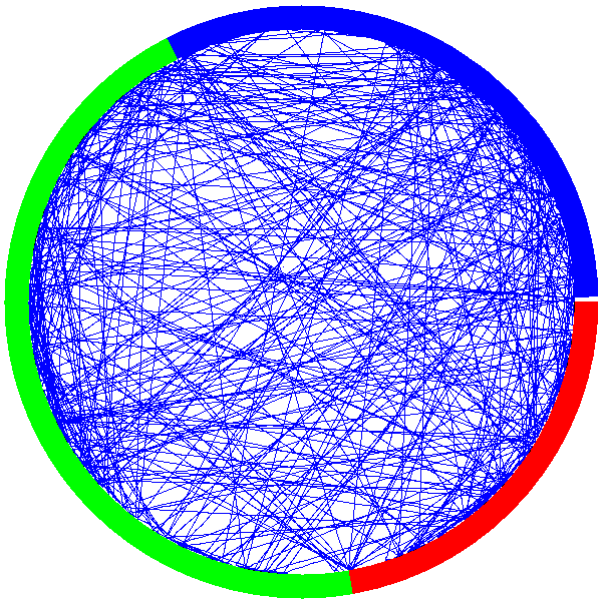
School of Computer Science
Carnegie Mellon



T=6

36

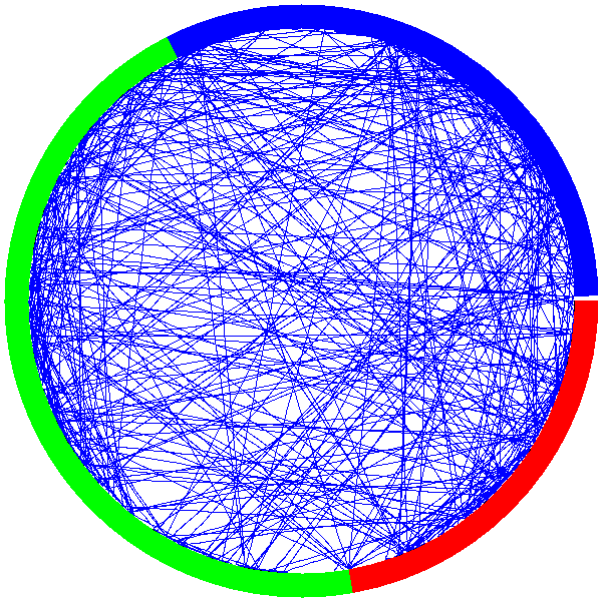
School of Computer Science
Carnegie Mellon



T=7

37

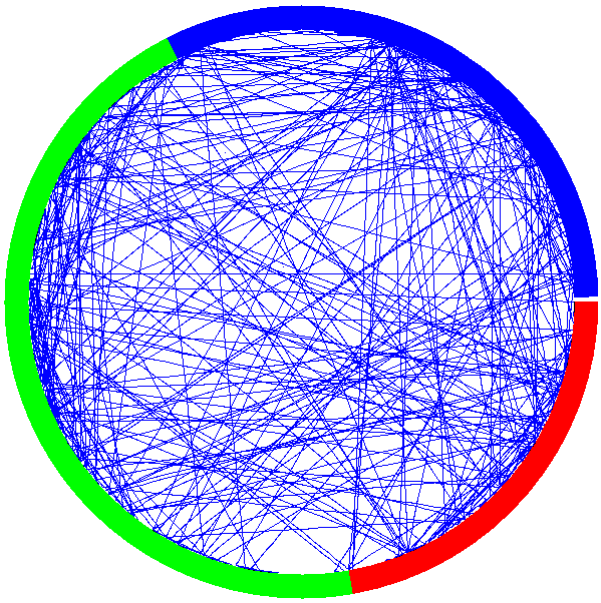
School of Computer Science
Carnegie Mellon



T=8

38

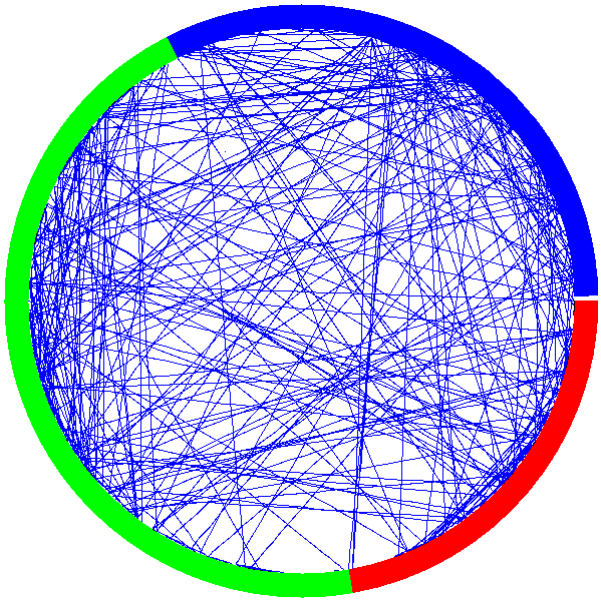
School of Computer Science
Carnegie Mellon



T=9

39

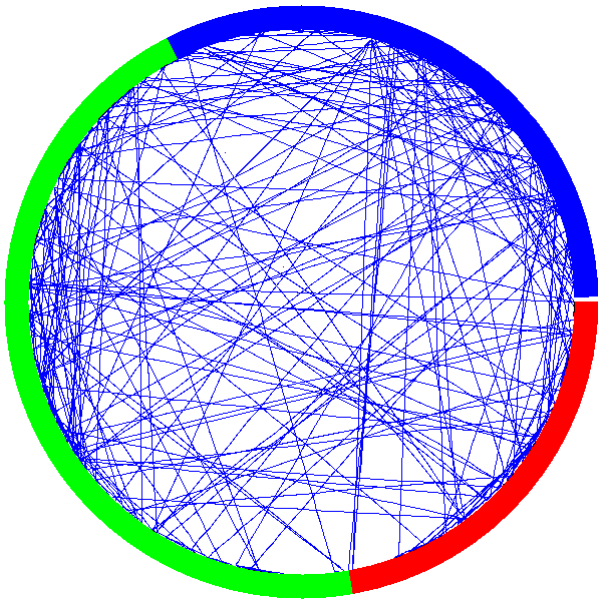
School of Computer Science
Carnegie Mellon



T=10

40

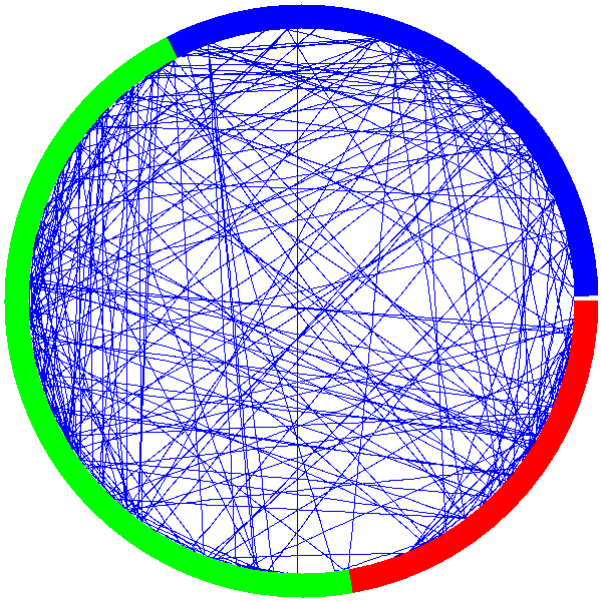
School of Computer Science
Carnegie Mellon



T=11

41

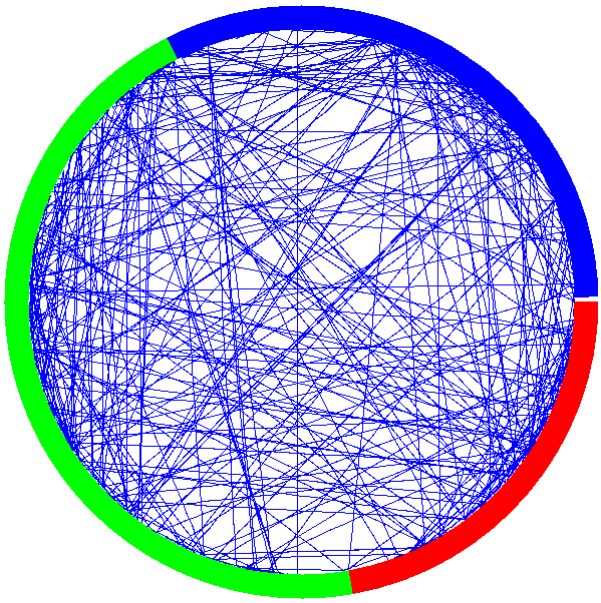
School of Computer Science
Carnegie Mellon



T=12

42

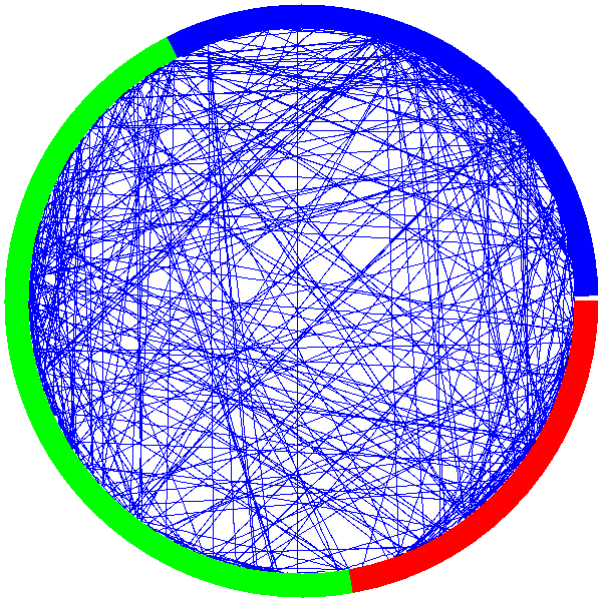
School of Computer Science
Carnegie Mellon



T=13

43

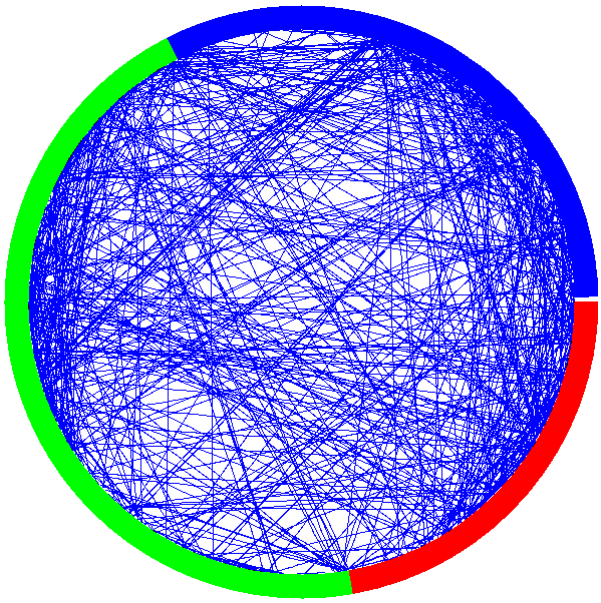
School of Computer Science
Carnegie Mellon



T=14

44

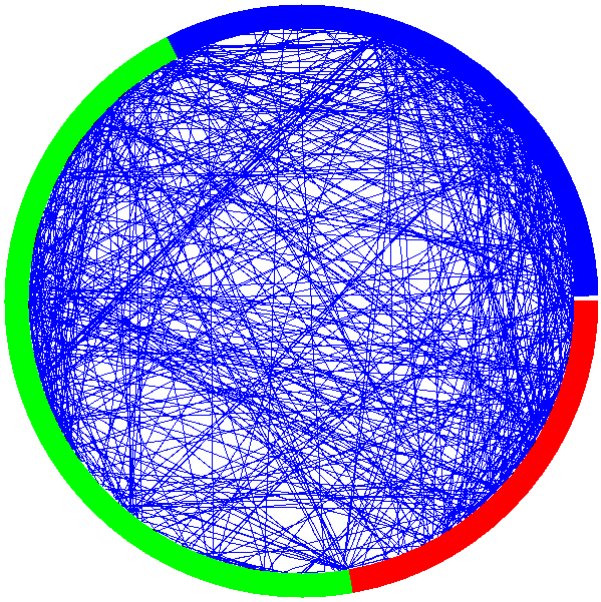
School of Computer Science
Carnegie Mellon



T=15

45

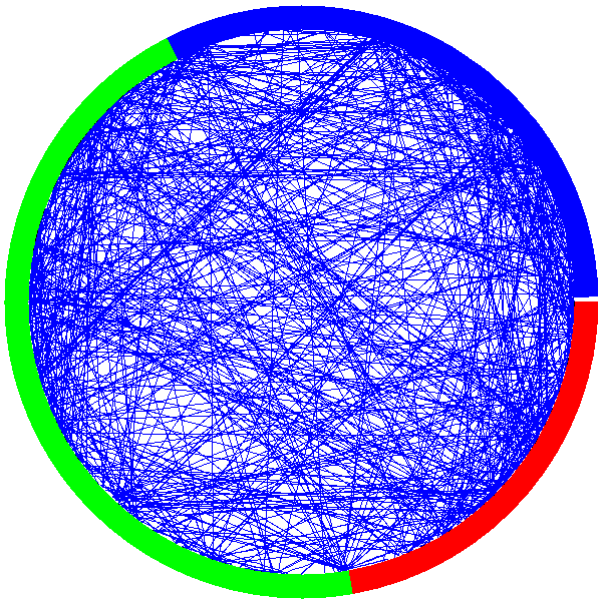
School of Computer Science
Carnegie Mellon



T=16

46

School of Computer Science
Carnegie Mellon

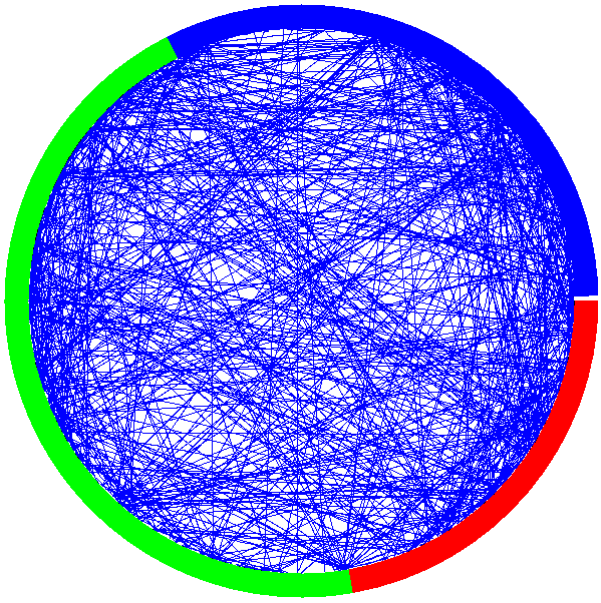


T=17

47

ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

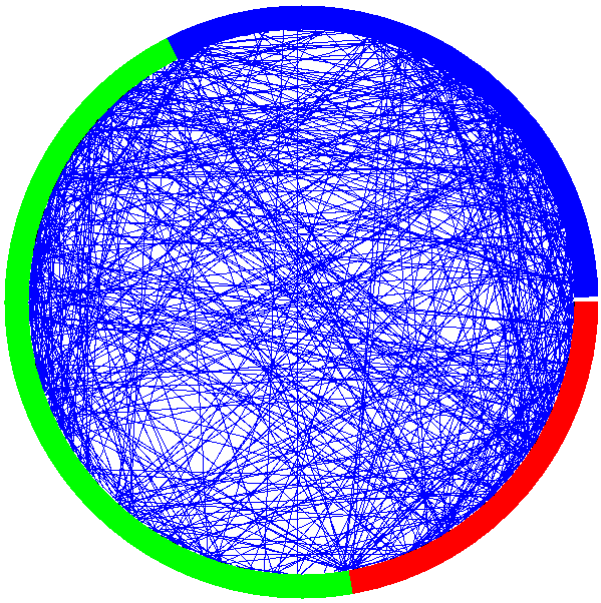


T=18

48

ML Lunch @ CMU, 11/ 24, 2008

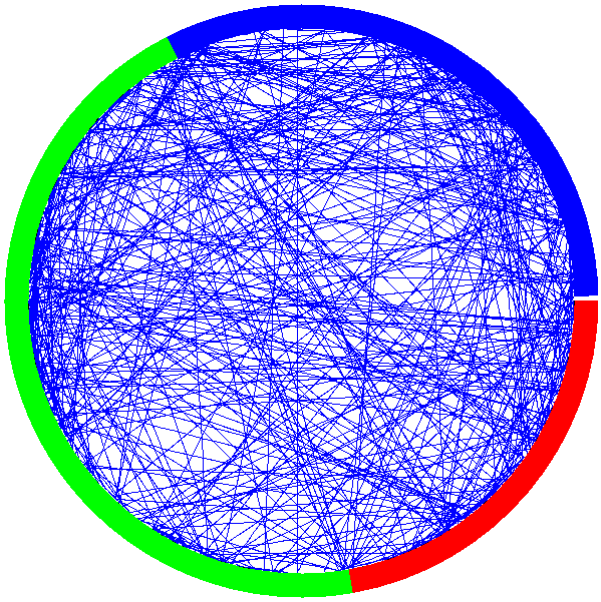
School of Computer Science
Carnegie Mellon



T=19

49

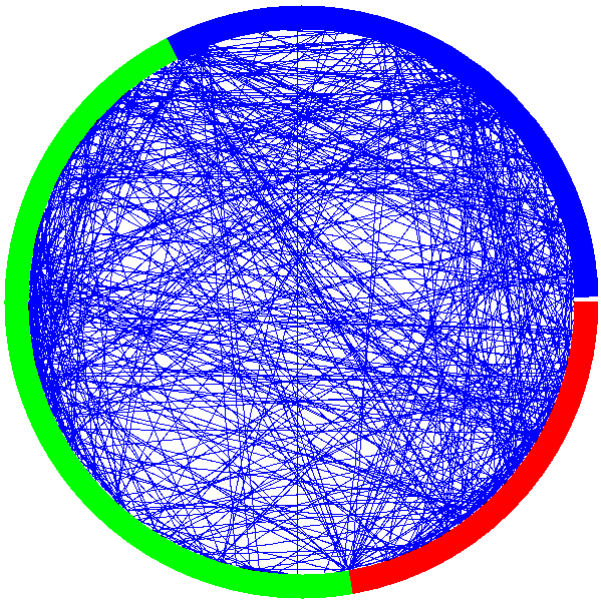
School of Computer Science
Carnegie Mellon



T=20

50

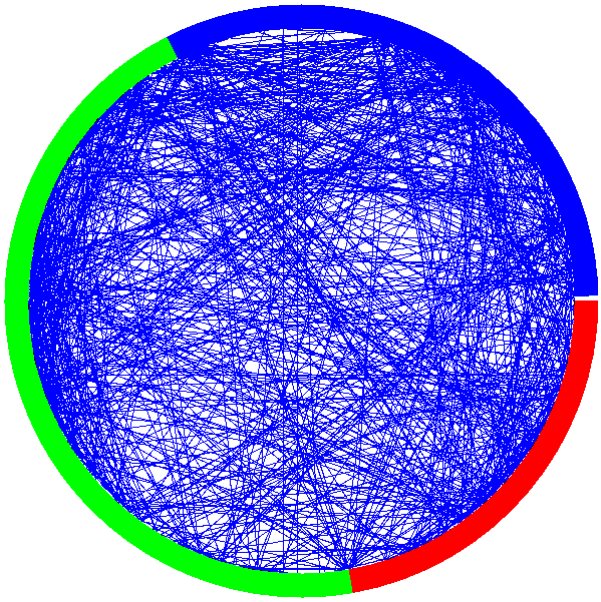
School of Computer Science
Carnegie Mellon



T=21

51

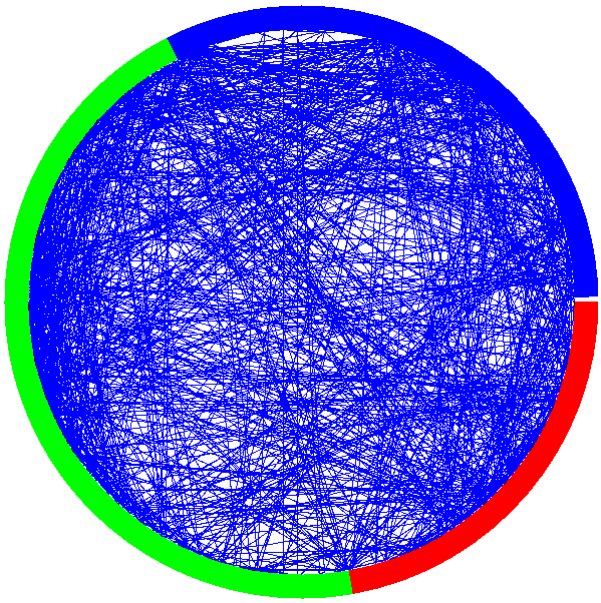
School of Computer Science
Carnegie Mellon



T=22

52

School of Computer Science
Carnegie Mellon

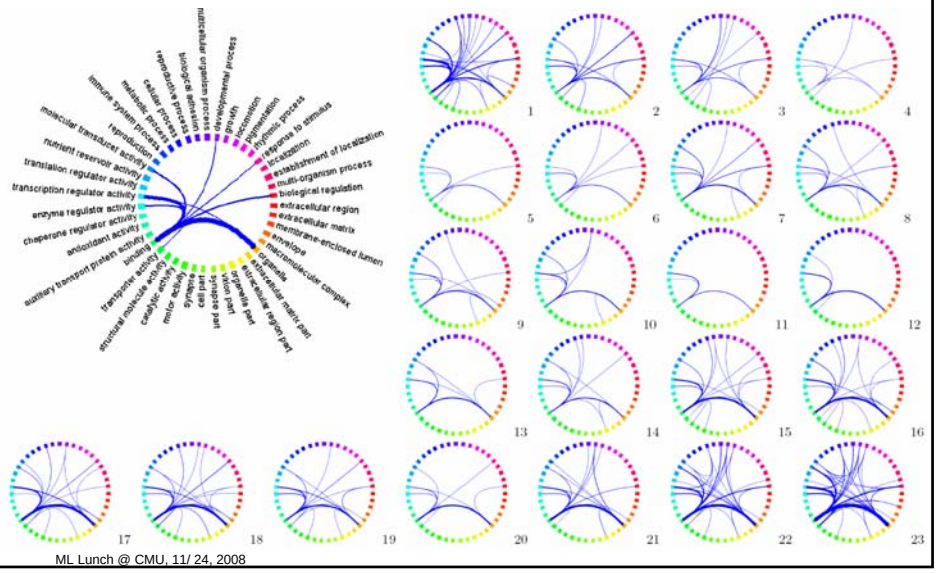


T=23

53

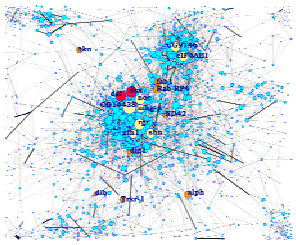
School of Computer Science
Carnegie Mellon

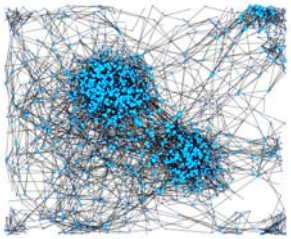
Transient Interaction

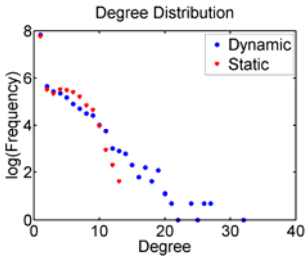


School of Computer Science
Carnegie Mellon

Static Versus Dynamic

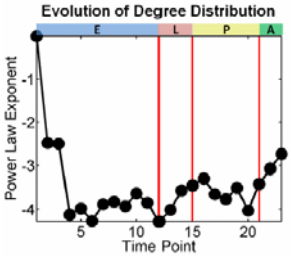


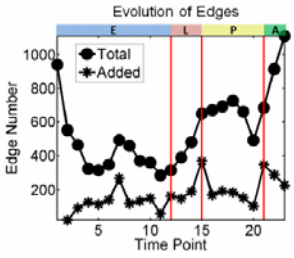


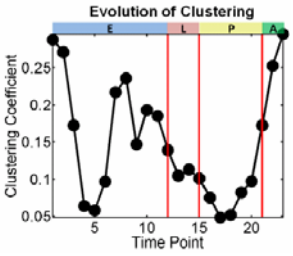


School of Computer Science
Carnegie Mellon

Evolution of Network Signatures

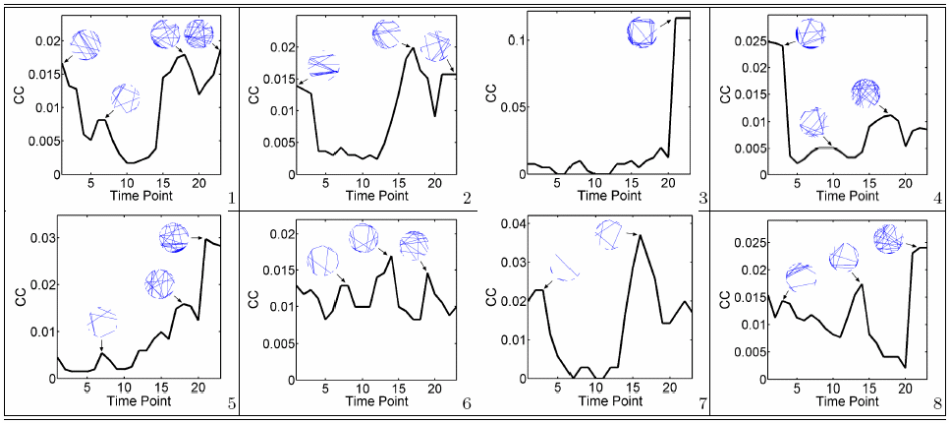






School of Computer Science
Carnegie Mellon

Transient Subgraph



School of Computer Science
Carnegie Mellon

Future Work

- Analyzing time-space data in biological processes
  - Drosophila life cycle
  - Breast cancer progression and reversal
  - Inflammatory response in endotoxinated mice
- Other dynamic behaviors of networks
  - Differentiation: tree of networks
  - Detection of sudden changes
  - Active learning – when to get more samples
- Open theoretical issues
  - Consistence (pattern, value, …)
  - Confidence
  - Stability
  - Sample complexity

Genome-Phenome Association and Structured Input-Out Learning

Traditional view

causal SNP

ACGTTTTACTGTACAATT

a univariate phenotype:

i.e., disease/control

Modern view

causal SNP networks

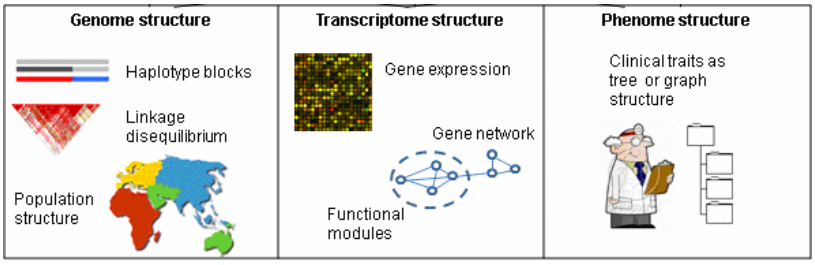
ACGTTTTACTGTACAATT

Multivariate complex syndrome (e.g., asthma):

age at onset,
presence/absence of eosinophilic inflammation,
history of eczema
genome-wide expression profile
…

School of Computer Science
Carnegie Mellon

Genome and Phenome Structures

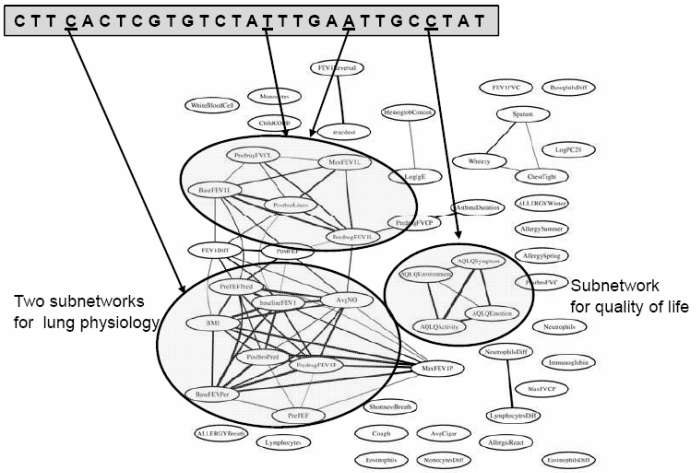


60

ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

The Asthma Phenotype Network



61

School of Computer Science
Carnegie Mellon

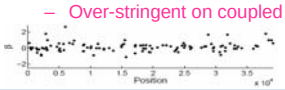
Inferring Genome-Phenome Association

Traditional view

ACGTTTTACTGTACAATT

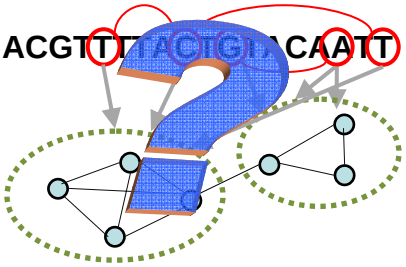
- **Pair-wise association tests?**
  - Ignore SNP dependencies
  - Many many FPs

- **Regression?**
  - Over-stringent on coupled SNPs



Modern view

ACGTTTTACTGTACAATT



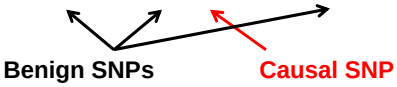
- **Structured regularized regression**
  - ✓ explicitly capture structures
  - ✓ efficient
  - ✓ sparse (parsimonious)
  - ✓ provable guarantees

62

School of Computer Science
Carnegie Mellon

Association Mapping

	Phenotype (BMI)	Genotype
Individual 1	2.5	. . C . . . . . T . . C . . . . T . . . . . C . . . . . A . . C . . . . T . . .
Individual 2	4.8	. . G . . . . . A . . G . . . . A . . . . . C . . . . . T . . C . . . . T . . .
⋮		
Individual N	4.7	. . G . . . . . T . . C . . . . T . . . . . G . . . . . T . . G . . . . T . . .



School of Computer Science
Carnegie Mellon

Association Mapping as Regression

	Phenotype (BMI)	Genotype
Individual 1	2.5	. . 0 . . . . . 1 . . 0 . . . . 0 . . .
Individual 2	4.8	. . 1 . . . . . 1 . . 1 . . . . 1 . . .
⋮		
Individual N	4.7	. . 2 . . . . . 2 . . 1 . . . . 0 . . .

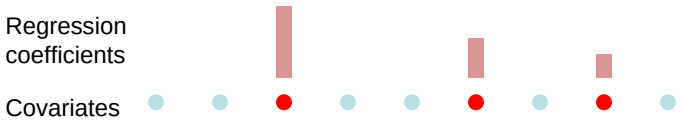
$$\mathbf{y}_i = \sum_{j=1}^{J} x_{ij}\beta_j$$

SNPs with large $|\beta_j|$ are relevant

School of Computer Science
Carnegie Mellon

Sparse Regression

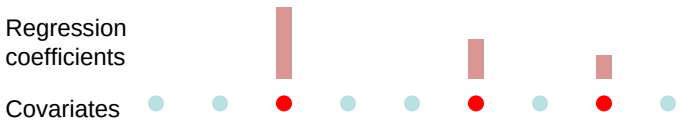
- Lasso (Tibshirani, 1996) : Learn a sparse regression model



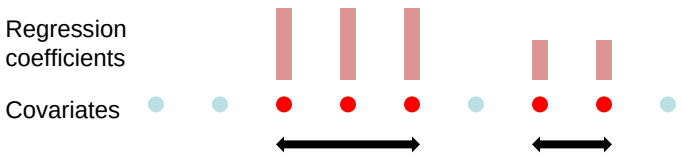
School of Computer Science
Carnegie Mellon

Sparse Regression

- Lasso (Tibshirani, 1996) : Learn a sparse regression model



- Fused lasso (Tibshirani et al., 2005) : Fuse adjacent coefficient values, assuming covariates are ordered



School of Computer Science
Carnegie Mellon

Sparse Regression

- Block-regularized regression

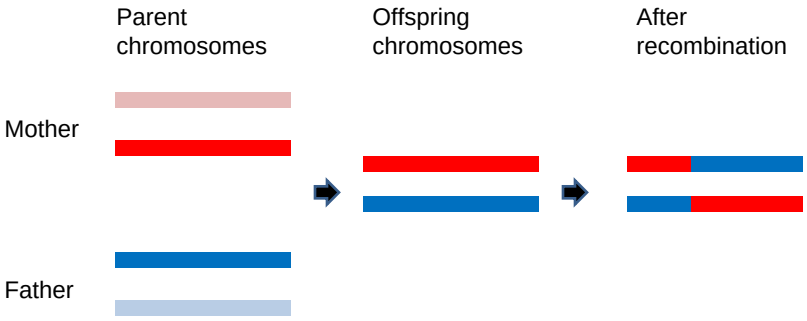
  - The block boundaries are determined probabilistically.

- Motivated by **association mapping** problem in computational biology
  - The block structure in genome arising from a non-random recombination process

67

School of Computer Science
Carnegie Mellon

Recombination

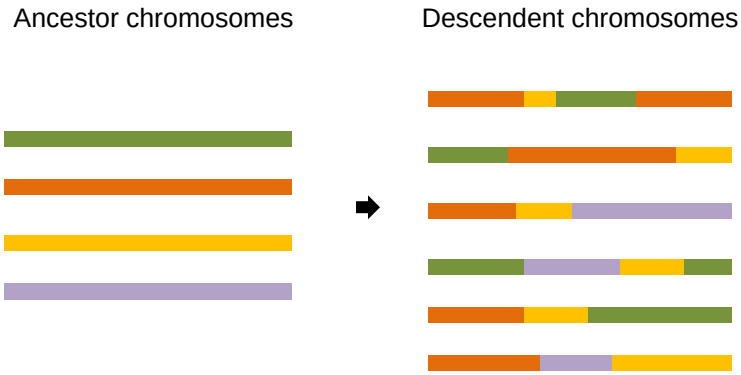


Recombination rate ρ : frequency of recombination per unit distance on chromosome (often, per kb)

68

School of Computer Science
Carnegie Mellon

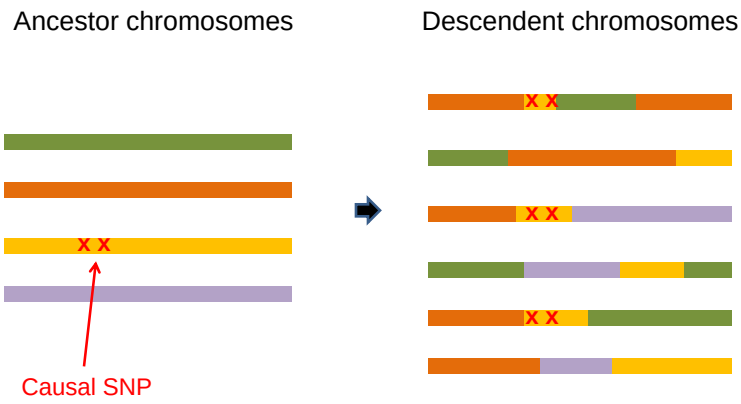
After Many Generations with Recombination ...



69

School of Computer Science
Carnegie Mellon

After Many Generations with Recombination ...



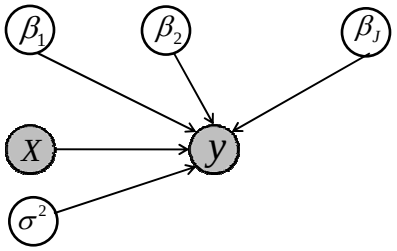
70

School of Computer Science
Carnegie Mellon

Bayesian Variable Selection

(George and McCulloch, 1993, Ishwaran and Rao, 2005)

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$



School of Computer Science
Carnegie Mellon

Bayesian Variable Selection

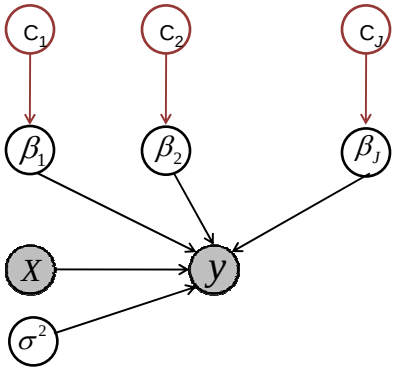
(George and McCulloch, 1993, Ishwaran and Rao, 2005)

If C_j = 0 (irrelevant), $\beta_j = 0$

If C_j = 1 (relevant), use Laplacian prior

$$\beta_j | c_j \sim \frac{1}{2(2\lambda\sigma^2)} \exp\left(-\frac{|\beta_j|}{2\lambda\sigma 2}\right)$$

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$



ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

Bayesian Variable Selection

(George and McCulloch, 1993, Ishwaran and Rao, 2005)

Bernoulli prior on C_i 's

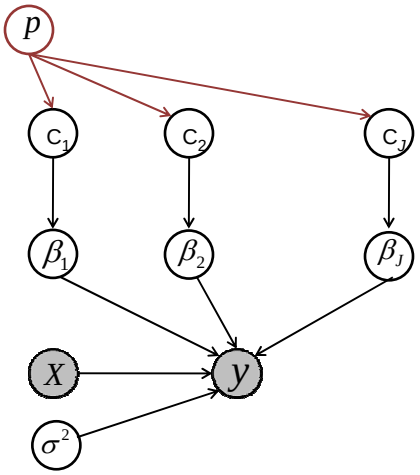
$$c_j \sim \text{Bernoulli}(p)$$

If C_j = 0 (irrelevant), $\beta_j = 0$

If C_j = 1 (relevant), use Laplacian prior

$$\beta_j | c_j \sim \frac{1}{2(2\lambda\sigma^2)} \exp\left(-\frac{|\beta_j|}{2\lambda\sigma 2}\right)$$

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$



73

School of Computer Science
Carnegie Mellon

Markov Chain Prior

$$P(\mathbf{c}) = P(c_1) \prod_{j=2}^{J} P(c_j | c_{j-1})$$

74

School of Computer Science
Carnegie Mellon

Markov Chain Prior

$$P(\mathbf{c}) = P(c_1) \prod_{j=2}^{J} P(c_j|c_{j-1})$$

- $c_j = c_{j-1}$ if
  1) the distance between the two SNPs is small, or
  2) the recombination rate between the two SNPs is small

75

School of Computer Science
Carnegie Mellon

Markov Chain Prior

$$P(\mathbf{c}) = P(c_1) \prod_{j=2}^{J} P(c_j|c_{j-1})$$

Poisson process

$$P(c_j|c_{j-1}) = \exp(-d_j\rho_j)\ \delta(c_j, c_{j-1}) + (1 - \exp(-d_j\rho_j))\ \Pi_{c_{j-1},c_j}$$

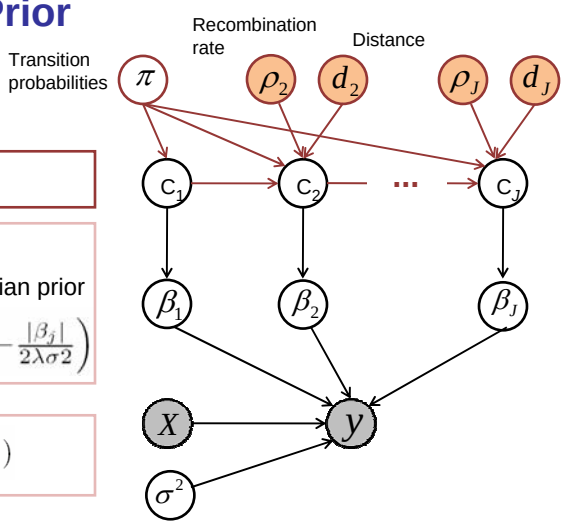
- ρ_j : Recombination rate at *j*th SNP
- d_j : Distance between *j*th and (*j*-1)th SNP
- Π : Transition probability matrix

$$\begin{pmatrix} \pi_0 & 1-\pi_0 \\ 1-\pi_1 & \pi_1 \end{pmatrix}$$

76

School of Computer Science
Carnegie Mellon

Block-regularized Regression with Markov Chain Prior



Markov chain prior on C_j 's

If C_j = 0 (irrelevant), $\beta_j = 0$

If C_j = 1 (relevant), use Laplacian prior

$$\beta_j | c_j \sim \frac{1}{2(2\lambda\sigma^2)} \exp\left(-\frac{|\beta_j|}{2\lambda\sigma 2}\right)$$

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

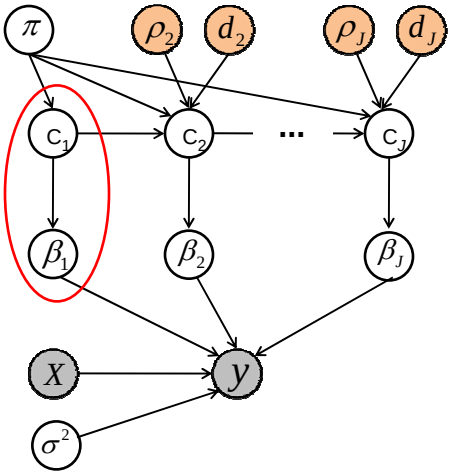
School of Computer Science
Carnegie Mellon

Learning with MCMC

- In each iteration
  - Sample (C_j, β_j)'s

$$p(\beta_j, c_j | \boldsymbol{\beta}_{-j}, \mathbf{c}_{-j}, \mathbf{y}, \mathbf{X}, \sigma^2)$$
$$= p(\beta_j | \boldsymbol{\beta}_{-j}, \mathbf{c}, \mathbf{y}, \mathbf{X}, \sigma^2)$$
$$\cdot P(c_j | \boldsymbol{\beta}_{-j}, \mathbf{c}_{-j}, \mathbf{y}, \mathbf{X}, \sigma^2)$$

  - Sample π
  - Sample σ^2



ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

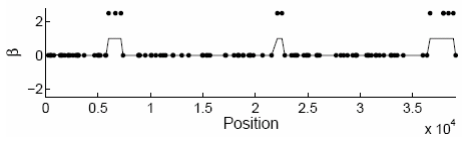
Experiments

- Simulation study
  - Comparison with
    - Bayesian variable selection with independent Bernoulli prior
    - Lasso
    - Ridge regression
  - Simulate covariates from *ms* (Hudson, 2002)
  - Estimate recombination rates using *PHASE* (Li and Stephens, 2004)
  - 10 relevant SNPs out of 100-250 SNPs
  - 180 individuals
  - MCMC sampling for 5000 iterations after 2000 burn-in
- Mouse dataset

School of Computer Science
Carnegie Mellon

Simulations

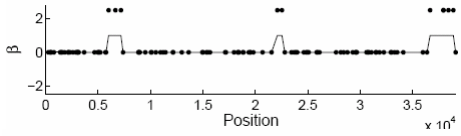
True Model



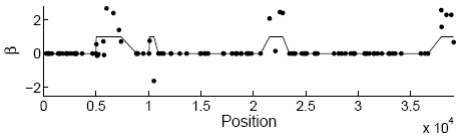
School of Computer Science
Carnegie Mellon

Simulations

True Model



Block-regularized regression

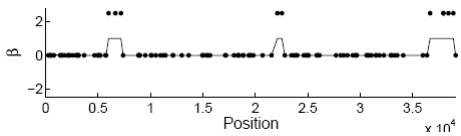


81

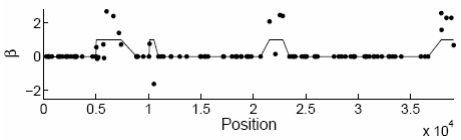
School of Computer Science
Carnegie Mellon

Simulations

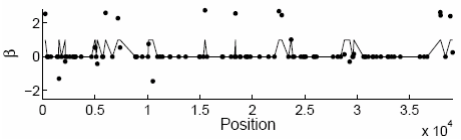
True Model



Block-regularized regression



Independent Bernoulli prior

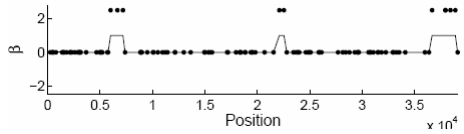


82

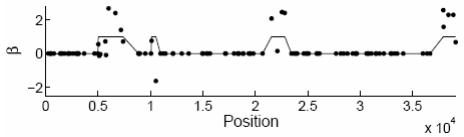
School of Computer Science
Carnegie Mellon

Simulations

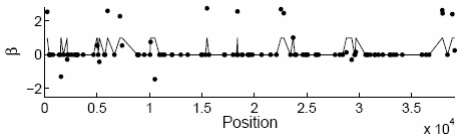
True Model



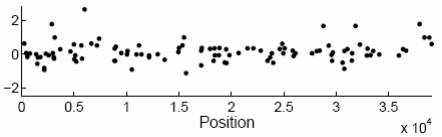
Block-regularized regression



Independent Bernoulli prior



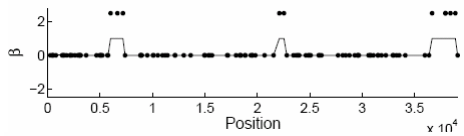
Ridge regression



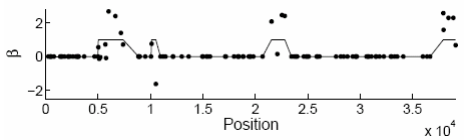
School of Computer Science
Carnegie Mellon

Simulations

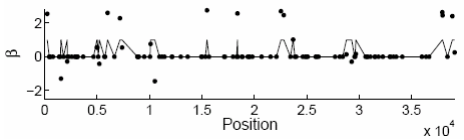
True Model



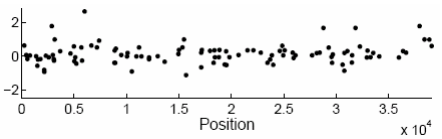
Block-regularized regression



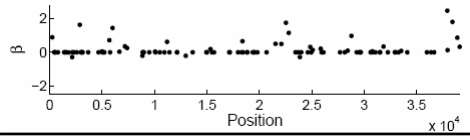
Independent Bernoulli prior



Ridge regression



Lasso



ML Lunch @ CMU, 11/ 24, 2008

School of Computer Science
Carnegie Mellon

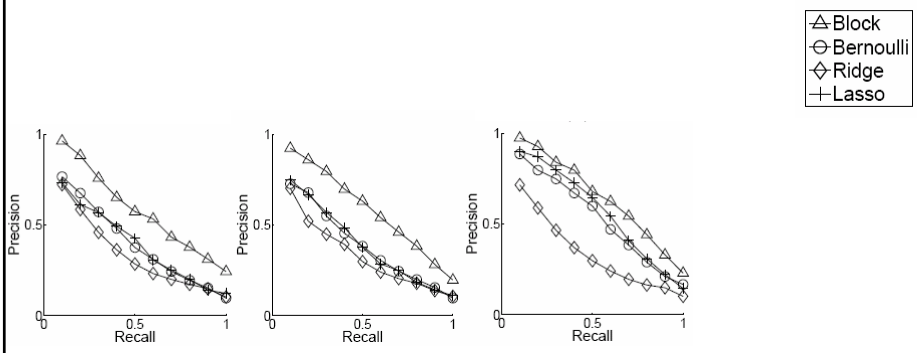
Precision and Recall

ρ = 0.05/kb

Low — Recombination Rate — High

School of Computer Science
Carnegie Mellon

Precision and Recall

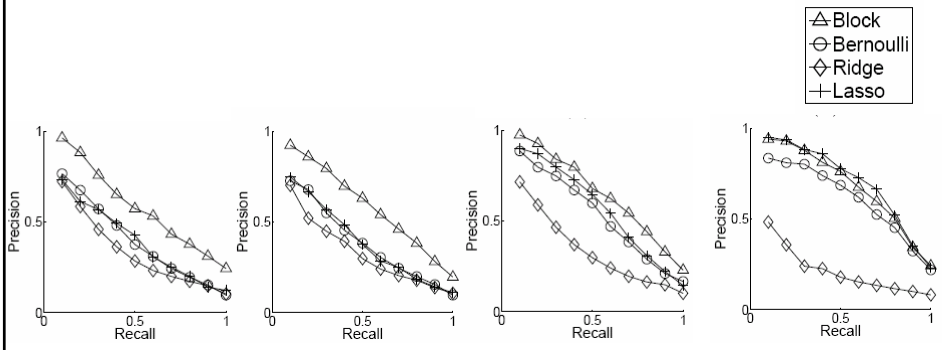


ρ = 0.05/kb ρ = 0.1/kb ρ = 0.5/kb ρ = 1.0/kb

Low — Recombination Rate — High

School of Computer Science
Carnegie Mellon

Precision and Recall



ρ = 0.05/kb ρ = 0.1/kb ρ = 0.5/kb ρ = 1.0/kb

Low Recombination Rate High

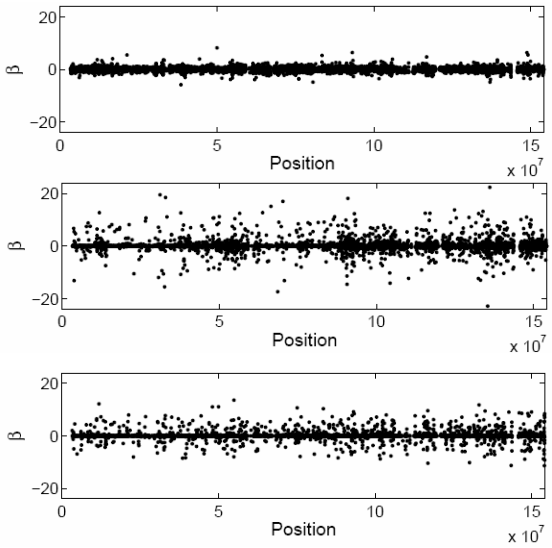
School of Computer Science
Carnegie Mellon

Mouse Data (BROAD institute)

Block-regularized regression

Independent Bernoulli prior

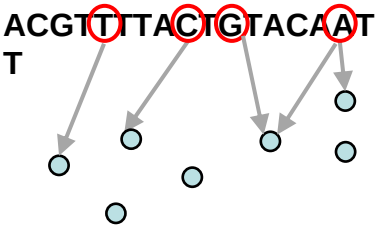
Lasso



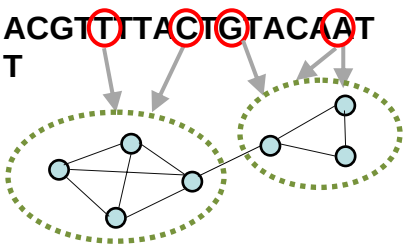
School of Computer Science
Carnegie Mellon

Multiple-trait Association

Simple (univariate) traits



Complex (multivariate) traits



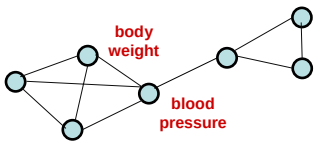
$$\hat{\mathbf{B}}^{\text{lasso}} = \text{argmin} \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T \cdot (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \sum_{k,j} |\beta_{kj}| \quad \Rightarrow$$

Need additional constraints on **B** apart from sparsity

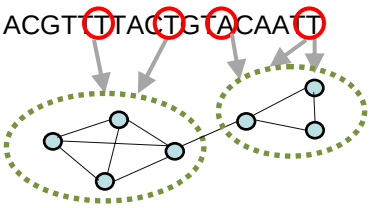
School of Computer Science
Carnegie Mellon

Multiple-trait Association

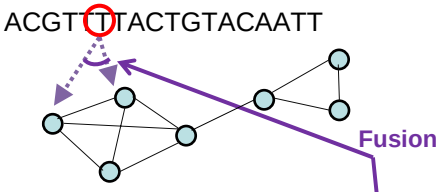
Step 1: Thresholded correlation graph of phenotypes



Overall effect of the fusion penalty



Step 2: Graph-constrained fused lasso



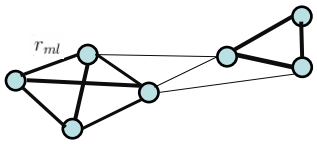
$$\hat{\mathbf{B}}^{\text{GC}} = \text{argmin} \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T \cdot (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \sum_k \sum_j |\beta_{jk}| + \gamma \sum_{(m,l)\in E} \sum_j |\beta_{jm} - \text{sign}(r_{ml})\beta_{jl}|$$

ML Lunch @ CMU, 11/ 24, 2008

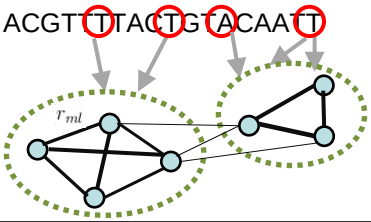
School of Computer Science
Carnegie Mellon

Multiple-trait Association

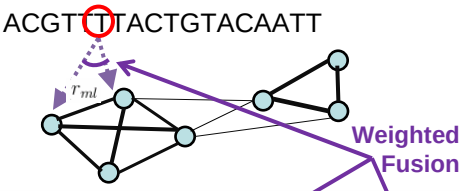
Step 1: Thresholded correlation graph of phenotypes with **weights**



Overall effect of the weighted fusion penalty



Step 2: Graph-weighted fused lasso



$$\hat{\mathbf{B}}^{\mathrm{GW}} = \operatorname{argmin} \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T \cdot (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)$$
$$+\lambda \sum_k \sum_j |\beta_{jk}| + \gamma \sum_{(m,l)\in E} f(r_{ml}) \sum_j |\beta_{jm} - \operatorname{sign}(r_{ml})\beta_{jl}|$$

School of Computer Science
Carnegie Mellon

Convex Optimization

- Quadratic programming formulation
  - Graph-constrained fused lasso

$$\hat{\mathbf{B}}^{\mathrm{GC}} = \operatorname{argmin} \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T \cdot (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)$$
$$\text{s. t.} \quad \sum_k \sum_j |\beta_{jk}| \le s_1 \text{ and } \sum_{(m,l)\in E} \sum_j |\beta_{jm} - \operatorname{sign}(r_{ml})\beta_{jl}| \le s_2$$

  - Graph-weighted fused lasso

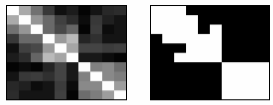
$$\hat{\mathbf{B}}^{\mathrm{GW}} = \operatorname{argmin} \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T \cdot (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)$$
$$\text{s. t.} \quad \sum_k \sum_j |\beta_{jk}| \le s_1 \text{ and } \sum_{(m,l)\in E} f(r_{ml}) \sum_j |\beta_{jm} - \operatorname{sign}(r_{ml})\beta_{jl}| \le s_2$$

- Many publicly available software packages for solving convex optimization problems can be used

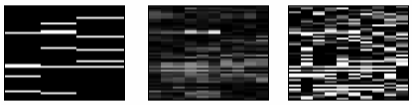
School of Computer Science
Carnegie Mellon

Simulated Data

Phenotype Correlation Structure

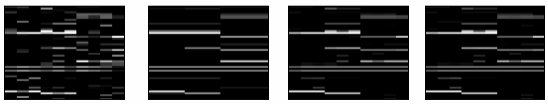


A B



C D E

Regression Coefficients



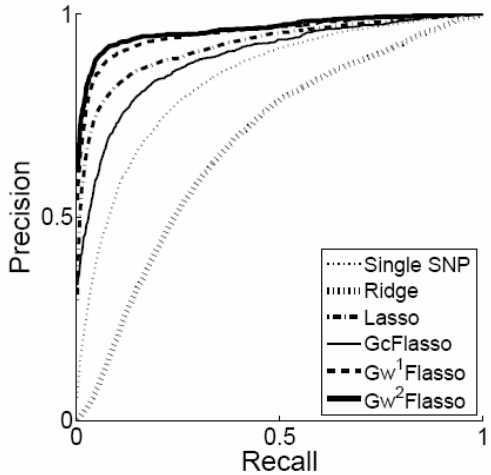
F (Lasso) G (G_cFlasso) H (G_wFlasso) I

A: The correlation matrix of phenotypes, B: the edges of the phenotype correlation graph obtained at threshold 0.3 are shown as white pixels, C: The true regression coefficients used in simulation. Rows correspond to SNPs and columns to phenotypes. D: -log(p-value). Absolute values of the estimated regression coefficients are shown for E: ridge regression, F: lasso, G: G_cFlasso, H: G_1^wFlasso, and I: G_2^wFlasso.

93

School of Computer Science
Carnegie Mellon

ROC on Simulated Data



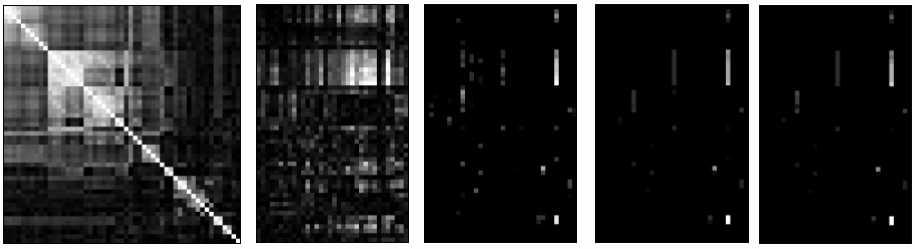
The same trend is observed under:

- Different sample sizes,
- Association strength
- Edge weight cut-offs

94

School of Computer Science
Carnegie Mellon

Asthma Multiple-trait Association



Phenotype Correlation Structure

Single-marker Single-trait test

Lasso

Graph-constrained Fused lasso

Graph-weighted Fused lasso

95

School of Computer Science
Carnegie Mellon

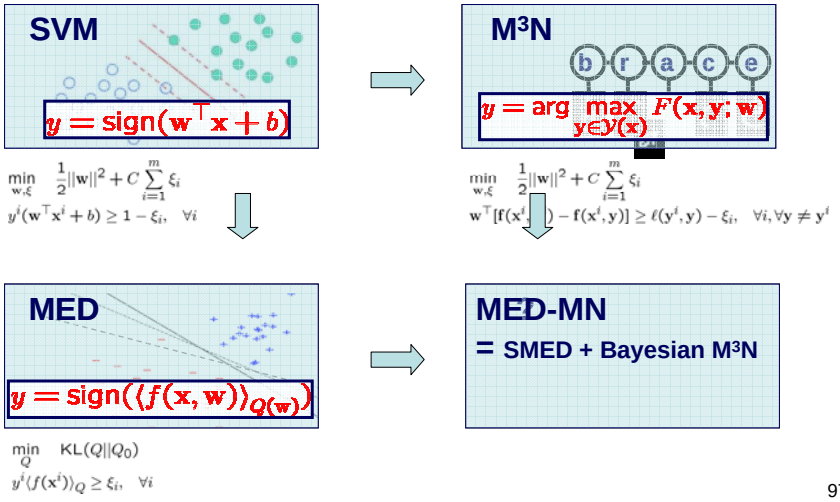
Summary

- Likelihood-based Structured Input or Structured Output
  - Block-regularized regression makes use of the prior knowledge on the block structure such as distance and recombination rate between adjacent SNPs.
  - Graph-guided fused lasso framework incorporates correlation information among phenotypes to detect pleiotropic effect of genotypic variations.
- Future Work
  - Combine structural information in both genome and phenome in a single statistical method

96

School of Computer Science
Carnegie Mellon

Margin-Based Discriminative Learning Paradigms



School of Computer Science
Carnegie Mellon

Maximum Entropy Discrimination Markov Networks

- Structured MaxEnt Discrimination (SMED):

$$\text{P1}: \quad \min_{p(\mathbf{w}),\xi} KL(p(\mathbf{w})\|p_0(\mathbf{w})) + U(\xi)$$
$$\text{s.t.} \quad p(\mathbf{w}) \in \mathcal{F}_1, \ \xi_i \geq 0, \forall i.$$

generalized maximum entropy or *regularized* KL-divergence

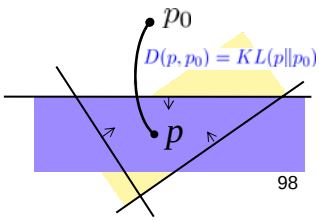
- Feasible subspace of weight distribution:

$$\mathcal{F}_1 = \{p(\mathbf{w}) : \int p(\mathbf{w})[\Delta F_i(\mathbf{y};\mathbf{w}) - \Delta\ell_i(\mathbf{y})]\,d\mathbf{w} \geq -\xi_i, \ \forall i, \forall \mathbf{y} \neq \mathbf{y}^i\},$$

expected margin constraints.

- Average from distribution of M³Ns

$$h_1(\mathbf{x}; p(\mathbf{w})) = \arg\max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \int p(\mathbf{w}) F(\mathbf{x},\mathbf{y};\mathbf{w}) d\mathbf{w}$$



School of Computer Science
Carnegie Mellon

Solution to MaxEnDNet

- Theorem:
  - Posterior Distribution:

$$p(\mathbf{w}) = \frac{1}{Z(\alpha)} p_0(\mathbf{w}) \exp\{\sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})[\Delta F_i(\mathbf{y};\mathbf{w}) - \Delta \ell_i(\mathbf{y})]\}$$

  - Dual Optimization Problem:

$$\begin{aligned} \text{D1}: \quad & \max_{\alpha} \; -\log Z(\alpha) - U^\star(\alpha) \\ & \text{s.t.} \;\; \alpha_i(\mathbf{y}) \geq 0, \; \forall i, \; \forall \mathbf{y}, \end{aligned}$$

$U^\star(\cdot)$ is the conjugate of the $U(\cdot)$, i.e., $U^\star(\alpha) = \sup_\xi \left(\sum_{i,y} \alpha_i(y)\xi_i - U(\xi)\right)$

99

School of Computer Science
Carnegie Mellon

Gaussian MaxEnDNet (reduction to M³N)

- Theorem
  - Assume $F(\mathbf{x},\mathbf{y};\mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x},\mathbf{y}), U(\xi) = C\sum_i \xi_i, \text{and } p_0(\mathbf{w}) = \mathcal{N}(\mathbf{w}|0,I)$

  - Posterior distribution: $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mu_{\mathbf{w}}, I), \text{ where } \mu_{\mathbf{w}} = \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})\Delta\mathbf{f}_i(\mathbf{y})$
  - Dual optimization:

$$\begin{aligned} & \max_{\alpha} \; \sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})\Delta\ell_i(\mathbf{y}) - \frac{1}{2}\|\sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})\Delta\mathbf{f}_i(\mathbf{y})\|^2 \\ & \text{s.t.} \;\; \sum_{\mathbf{y}} \alpha_i(\mathbf{y}) = C; \; \alpha_i(\mathbf{y}) \geq 0, \; \forall i, \; \forall \mathbf{y}, \end{aligned}$$

  - Predictive rule:

$$h_1(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \int p(\mathbf{w}) F(\mathbf{x},\mathbf{y};\mathbf{w})\, d\mathbf{w} = \arg\max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \mu_{\mathbf{w}}^\top \mathbf{f}(\mathbf{x},\mathbf{y})$$

- Thus, MaxEnDNet subsumes M³Ns and admits all the merits of max-margin learning
- Furthermore, MaxEnDNet has at least three advantages …

100

School of Computer Science
Carnegie Mellon

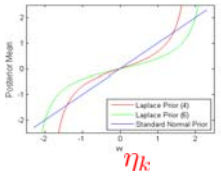
Three Advantages

- An averaging Model: PAC-Bayesian prediction error guarantee

$$\Pr_Q(M(h,\mathbf{x},\mathbf{y}) \le 0) \le \Pr_D(M(h,\mathbf{x},\mathbf{y}) \le \gamma) + O\left(\sqrt{\frac{\gamma^{-2}KL(p||p_0)\ln(N|\mathcal{Y}|) + \ln N + \ln \delta^{-1}}{N}}\right).$$

- Entropy regularization: Introducing useful biases
  - Standard Normal prior => reduction to standard M^{3}N (we've seen it)
  - Laplace prior => Posterior shrinkage effects (sparse M^{3}N)

$$\forall k,\ \langle w_k \rangle_p = \frac{2\eta_k}{\lambda - \eta_k^2}$$



η_k

- Integration of Generative and Discriminative principles
  - Incorporate latent variables and structures (PoMEN)
  - Semisupervised learning (with partially labeled data)

School of Computer Science
Carnegie Mellon

Key Challenges

- Extremely high dimensionality and low data volume
  - d ~ 1M
  - N ~ 1K
  - Sample complexity with bounded error?
- Sparsity bias of the model
  - Often <100 features out of the !M are relevant
  - Regularization schemes to enforce sparsity
- Structures and hidden variables
  - Inputs and outputs often bear intricate structures (e.g., chain or graphical dependencies)
  - How to capture other latent structures between unobserved variables
- Generalizability and scalability
  - Move efficient convex opt solver and Bayesian inference algorithms
- Provable theoretical guarantees
  - Consistency and sparsistency
  - Stability, convergence rate, etc.