Path Coding Penalties for Directed Acyclic Graphs

Julien Mairal and Bin Yu

University of California, Berkeley

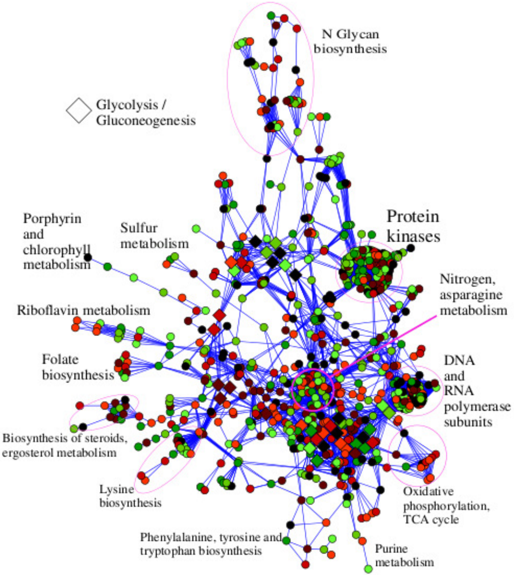
OPT'11, Sierra Nevada, 2011

What this work is about

- Feature selection in graphs.
- Structured sparsity.
- Non-convex and convex optimization.
- Network flow optimization.

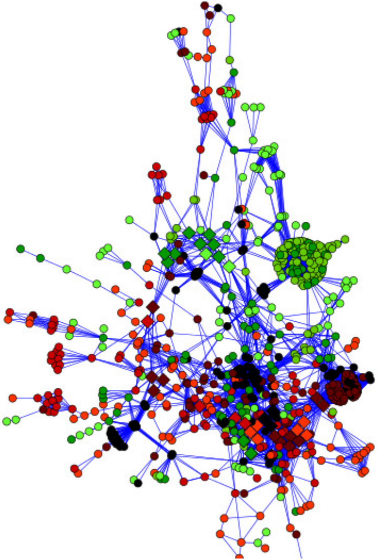
Metabolic network of the budding yeast

from Rapaport, Zinovyev, Dutreix, Barillot, and Vert [2007]



Metabolic network of the budding yeast

from Rapaport, Zinovyev, Dutreix, Barillot, and Vert [2007]



Sparse estimation problems

Where optimization/machine learning/signal processing meet

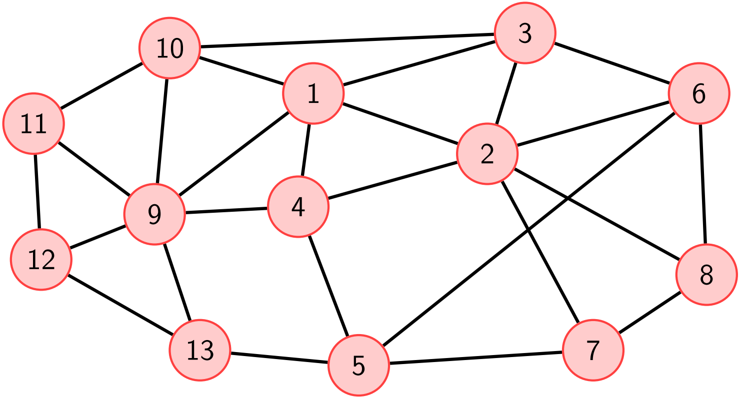
$$\min_{\mathbf{w}\in\mathbb{R}^p} \underbrace{g(\mathbf{w})}_{\text{convex, smooth}} + \underbrace{\lambda\Omega(\mathbf{w})}_{\text{regularization}},$$

Ω encodes some a priori knowledge on $\mathbf{w}$.

- squared ℓ_2-norm (ridge regression);
- ℓ_0-penalty;
- ℓ_1-norm [Tibshirani, 1996, Chen et al., 1999];
- Group Lasso [Turlach et al., 2005, Yuan and Lin, 2006];
- Hierarchical-norms [Zhao, Rocha, and Yu, 2009];
- **Structured sparsities** [Jenatton et al., 2009, Huang et al., 2009, Jacob et al., 2009, Baraniuk et al., 2010, Micchelli et al., 2011].

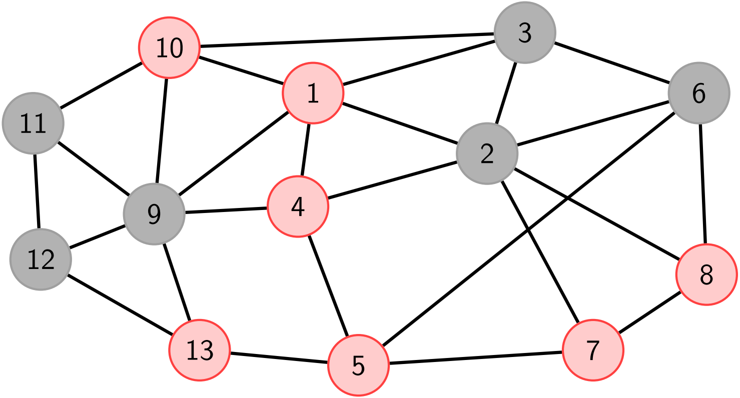
Graph sparsity

$G = (V, E)$, with $V = \{1, \ldots, p\}$



Graph sparsity

Encouraging patterns with a small number of connected components



Structured sparsity for graphs

the non-convex penalty of Huang, Zhang, and Metaxas [2009]

$$\varphi_{\mathcal{G}}(\mathbf{w}) \triangleq \min_{\mathcal{J} \subseteq \mathcal{G}} \Big\{ \sum_{g \in \mathcal{J}} \eta_g \ \text{ s.t. } \ \mathrm{Supp}(\mathbf{w}) \subseteq \bigcup_{g \in \mathcal{J}} g \Big\}.$$

$\mathcal{G}$ is a pre-defined set of groups (subsets) of variables in $\{1, \ldots, p\}$.

- the penalty is **non-convex**.
- is **NP-hard** to compute (set cover problem).
- The pattern of non-zeroes in $\mathbf{w}$ is a **union** of (a few) groups.

It can be rewritten as a boolean linear program:

$$\varphi_{\mathcal{G}}(\mathbf{w}) = \min_{\mathbf{x} \in \{0,1\}^{|\mathcal{G}|}} \Big\{ \boldsymbol{\eta}^\top \mathbf{x} \ \text{ s.t. } \ \mathbf{N}\mathbf{x} \geq \mathrm{Supp}(\mathbf{w}) \Big\}.$$

Structured sparsity for graphs

convex relaxation and the penalty of Jacob, Obozinski, and Vert [2009]

The penalty of Huang et al. [2009]:

$$\varphi_{\mathcal{G}}(\mathbf{w}) = \min_{\mathbf{x} \in \{0,1\}^{|\mathcal{G}|}} \left\{ \boldsymbol{\eta}^\top \mathbf{x} \ \text{ s.t. } \ \mathbf{N}\mathbf{x} \geq \text{Supp}(\mathbf{w}) \right\}.$$

A convex LP-relaxation:

$$\psi_{\mathcal{G}}(\mathbf{w}) \triangleq \min_{\mathbf{x} \in \mathbb{R}_+^{|\mathcal{G}|}} \left\{ \boldsymbol{\eta}^\top \mathbf{x} \ \text{ s.t. } \ \mathbf{N}\mathbf{x} \geq |\mathbf{w}| \right\}.$$

Lemma: $\psi_{\mathcal{G}}$ is the penalty of Jacob et al. [2009] with the ℓ_∞-norm.

Structured sparsity for graphs

Group structure for graphs.

Natural choices to encourage connectivity in the graph is to define $\mathcal{G}$ as

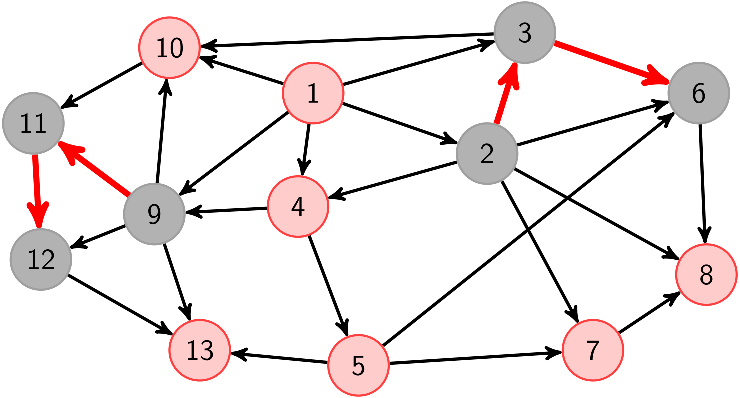
1. pairs of vertices linked by an arc. **only models local interactions**;
2. all connected subgraphs up to a size L. **cumbersome/intractable**;
3. all connected subgraphs. **intractable**.

Question

Can we replace connected subgraphs by another structure which (i) is rich enough to model long-range interactions in the graph, and (ii) leads to computationally feasible penalties?

Our solution when the graph is a DAG

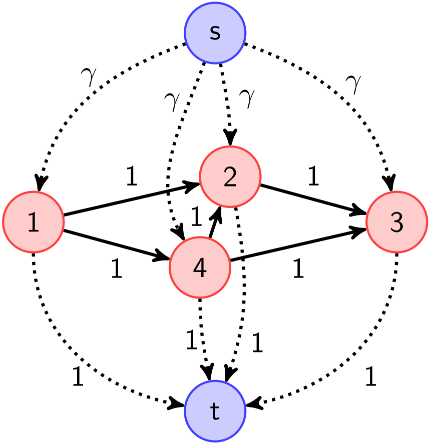
1. Define $\mathcal{G}$ to be the **set of all paths in the DAG**.
2. Define η_g to be $\gamma + |g|$ (the cost of selecting a path g).

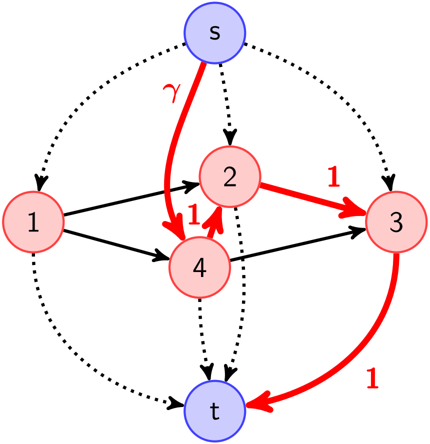


$$\varphi_{\mathcal{G}}(\mathbf{w}) = (\gamma + 3) + (\gamma + 3)$$

Graph sparsity for DAGs

Decomposability of the weights $\eta_g = \gamma + |g|$



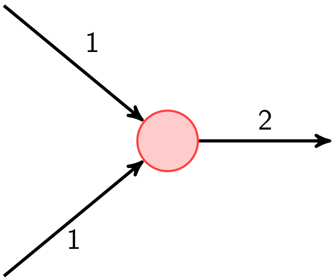


Quick introduction to network flows

References:

- Ahuja, Magnanti and Orlin. Network Flows, 1993
- Bertsekas. Network Optimization, 1998

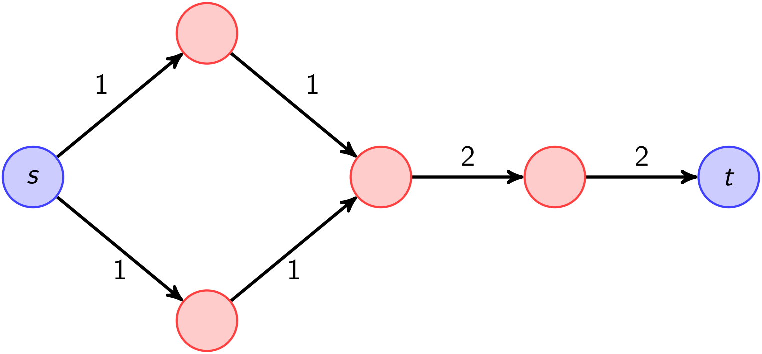
A flow f in $\mathcal{F}$ is a non-negative function on arcs that respects conservation constraints (Kirchhoff's law)



Quick introduction to network flows

Properties

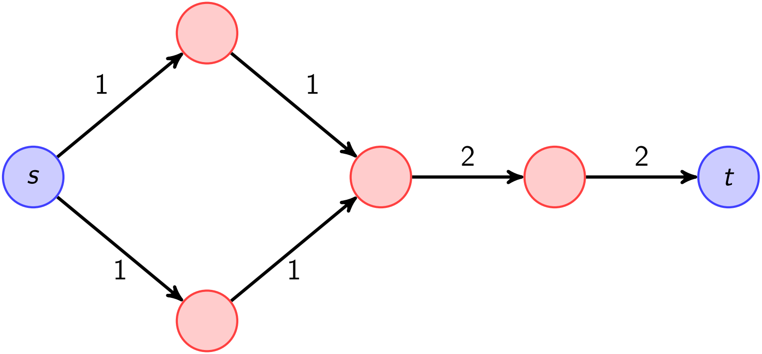
Flows usually go from a source node s to a sink node t.



Quick introduction to network flows

Properties

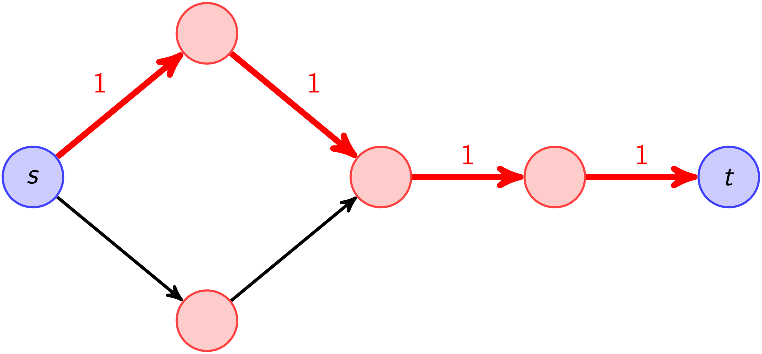
A flow on a DAG can be decomposed into “path-flows”.



Quick introduction to network flows

Properties

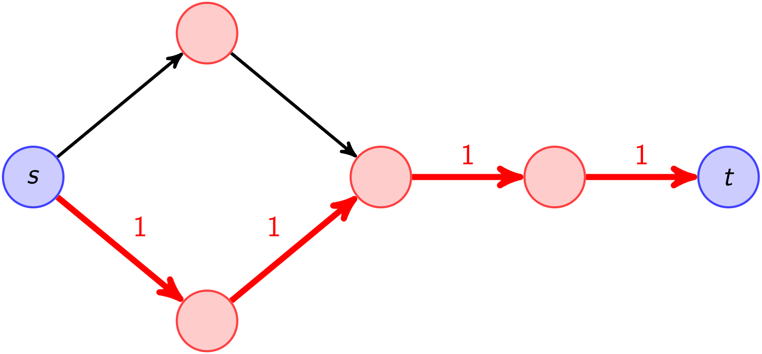
A flow on a DAG can be decomposed into “path-flows”.



Quick introduction to network flows

Properties

A flow on a DAG can be decomposed into “path-flows”.



Quick introduction to network flows

An optimization problem on paths might be transformed into an equivalent flow problem.

Proposition 1

$$\varphi_{\mathcal{G}}(\mathbf{w}) = \min_{f \in \mathcal{F}} \sum_{(u,v)\in E'} f_{uv} c_{uv} \ \text{ s.t. } \ s_j(f) \geq 1, \ \forall j \in \text{Supp}(\mathbf{w}),$$

Proposition 2

$$\psi_{\mathcal{G}}(\mathbf{w}) = \min_{f \in \mathcal{F}} \sum_{(u,v)\in E'} f_{uv} c_{uv} \ \text{ s.t. } \ s_j(f) \geq |\mathbf{w}_j|, \ \forall j \in \{1, \ldots, p\},$$

$\varphi_{\mathcal{G}}(\mathbf{w})$, $\psi_{\mathcal{G}}(\mathbf{w})$ and similarly the proximal operators, the dual norm of $\psi_{\mathcal{G}}$ **can be computed in polynomial time** using network flow optimization.

Application 1: Breast Cancer Data

The dataset is compiled from van't Veer et al. [2002] and the experiment follows Jacob et al. [2009].

Data description

- gene expression data of $p = 7910$ genes.
- $n = 295$ tumors, 78 metastatic, 217 non-metastatic.
- a graph between the genes was compiled by Chuang et al. [2007]. We arbitrary choose arc directions and heuristically remove cycles.

For each run, we keep 20% of the data as a test set, select parameters by 10-fold cross validation on the remaining 80% and retrain on 80%.

Application 1: Breast Cancer Data

Results

Results after 20 runs.

	Ridge	Lasso	Elastic-Net	Groups-pairs	ψ (convex)
error in %	31.0	36.0	31.5	35.9	30.2
error std.	6.1	6.5	6.7	6.8	6.8
nnz	7910	32.6	929	68.4	69.9
connex	58	30.9	355	13.1	1.3
stab	100	7.9	30.9	6.1	32

stab represents the percentage of genes selected in more than 10 runs.

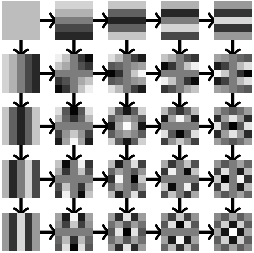
$\approx$ six proximal operators per second on our laptop cpu.

Application 2: Image denoising

Recipe, similarly to Elad and Aharon [2006]

- Extract all 10×10 overlapping patches from a noisy image.
- Obtain a sparse approximation of every patch.
- Average the estimates to obtain a clean image.

We use an orthogonal **DCT dictionary**:



Application 2: Image denoising

- Classical old-fashioned image processing dataset of 12 images.
- 7 levels of noise.
- Parameters optimized on the first 3 images.

σ	5	10	15	20	25	50	100
ℓ_0	**37.04**	33.15	31.03	29.59	28.48	25.26	22.44
ℓ_1	36.42	32.28	30.06	28.59	27.51	24.48	21.96
$\varphi_{\mathcal{G}}$	37.01	**33.22**	**31.21**	**29.82**	**28.77**	**25.73**	**22.97**
$\psi_{\mathcal{G}}$	36.32	32.17	29.99	28.54	27.49	24.54	22.12

PSNR: higher is better.

$\approx$ 4000 proximal operators per second on our laptop cpu.

Advertisement

- **Review monograph on sparse optimization:**
  F. Bach, R. Jenatton, J. Mairal and G. Obozinski. Optimization with Sparsity-Inducing Penalties. to appear in Foundation and Trends in Machine Learning.
- **SPAMS toolbox (C++)**
  - proximal gradient methods for ℓ_0, ℓ_1, elastic-net, fused-Lasso, group-Lasso, tree group-Lasso, tree-ℓ_0, sparse group Lasso, overlapping group Lasso...
  - ...for square, logistic, multi-class logistic loss functions.
  - handles sparse matrices, intercepts, provides duality gaps.
  - (block) coordinate descent, OMP, LARS-homotopy algorithms.
  - dictionary learning and matrix factorization (NMF).
  - fast projections onto some convex sets.
  - **soon: this work!**

Try it! `http://www.di.ens.fr/willow/SPAMS/`

References I

R. G. Baraniuk, V. Cevher, M. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 2010. to appear.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1999.

H.Y. Chuang, E. Lee, Y.T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Molecular systems biology*, 3(1), 2007.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 54(12):3736–3745, December 2006.

J. Huang, Z. Zhang, and D. Metaxas. Learning with structured sparsity. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

References II

L. Jacob, G. Obozinski, and J.-P. Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

R. Jenatton, J-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, 2009. preprint arXiv:0904.3523v1.

C.A. Micchelli, J.M. Morales, and M. Pontil. Regularizers for structured sparsity. *preprint arXiv:1010.0556v2*, 2011.

F. Rapaport, A. Zinovyev, M. Dutreix, E. Barillot, and J.P. Vert. Classification of microarray data using gene networks. *BMC bioinformatics*, 8(1):35, 2007.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

B. A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

References III

L.J. van't Veer, H. Dai, M.J. van de Vijver, Y.D. He, AA Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530, 2002.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67, 2006.

P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. 37(6A):3468–3497, 2009.