How to lasso positively, quickly and correctly

David Bryant

University of Otago

SUQ 2013

The Lasso (a.k.a. L_1 regularization)

Consider the linear inverse problem

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon.$$

An L_1-regularized solution takes a parameter δ and minimizes the penalized residual

$$\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \delta\|\boldsymbol{\beta}\|_1.$$

- This has the advantage that the solutions are typically sparse: used for variable selection.
- When $\delta = 0$ we obtain the least squares solution. As $\delta \to \infty$, the solutions approach $\mathbf{0}$.
- We can replace $\|\boldsymbol{\beta}\|$ with a weighted version $\sum_i w_i|\boldsymbol{\beta}_i|$.
- Introduced into statistics by Tibshirani (1996) under the name of ‘the LASSO

Computational challenge to efficiently compute LASSO solutions for a range of δ values (i.e. all?).

First observation (from KKT conditions): the set of LASSO solutions

$$\mathrm{argmin}_{\boldsymbol{\beta}}\{\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \delta\|\boldsymbol{\beta}\|_1\}$$

for $\delta \geq 0$ equals the set of solutions to

$$\mathrm{argmin}_{\boldsymbol{\beta}}\{\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 \text{ such that } \|\boldsymbol{\beta}\|_1 \leq \lambda\}$$

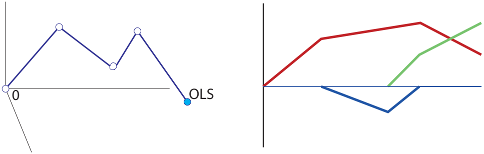
for $\lambda \geq 0$.

LARS - Lasso algorithm

Let $\boldsymbol{\beta}(\lambda)$ denote the optimal solution for a given λ. That is,

$$\boldsymbol{\beta}(\lambda) = \mathrm{argmin}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2^2 \text{ such that } \|\boldsymbol{\beta}\|_1 \leq \lambda.$$

It is not too hard to show that $\boldsymbol{\beta}(\lambda)$ is piecewise linear (as a function of λ).



Curve starts at **0** and finishes at the un-penalised solution.

Lasso computations

Osborne, Presnall and Turlach (2000)

394 citations

Efron, Hastie, Johnstone and Tibshirani (2004)

2909 citations

The positive Lasso

In many applications (including ours), we require variable coefficients which are non-negative.

$$\beta(\lambda) = \mathrm{argmin} \|\mathbf{X}\beta - \mathbf{y}\|^2 \text{ such that } \beta \geq \mathbf{0} \text{ and } \|\beta\| \leq \lambda.$$

LARS-LASSO

Efron et al. propose a 'Positive Lasso Lars' algorithm.

Let $\beta = \mathbf{0}$ and $\mathbf{c} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\beta)$.
while $\|\mathbf{c}\| > 0$
    $\mathcal{A} = \{i : \mathbf{c}_i \text{ is maximum}\}$.
    Choose the search direction $\mathbf{w}_{\mathcal{A}} = (\mathbf{X}'_{\mathcal{A}}\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{1}$
    Move β in direction $\mathbf{w}_{\mathcal{A}}$ until
        an entry becomes negative, OR
        new variable(s) join the set of those with maximum $\mathbf{c}_i$.
    Update $\mathbf{c} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\beta)$.
end

Problem: can fail if more than one variable leaves or enters $\mathcal{A}$ at any one time.

Multiple entries

Is this ‘one-at-a-time’ restriction a problem?

NO: you can always add random noise to break ties.

YES: With a positivity constraint ties appear as part of the algorithm (not just degenerate data). Also, we ran into problems with our degenerate models and problems.

Our algorithm for the positive Lasso

Let $\boldsymbol{\beta} = \mathbf{0}$ and $\mathbf{c} = \mathbf{X}'\mathbf{y}$.
while $\|\mathbf{c}\| > 0$
    $\mathcal{A} = \{i : \mathbf{c}_i \text{ is maximum}\}$.
    Choose the search direction $\mathbf{w}_{\mathcal{A}} = (\mathbf{X}'_{\mathcal{A}}\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{1}$
    Find $\mathbf{v}$ minimizing $\|\mathbf{X}_{\mathcal{A}}(\mathbf{v}_{\mathcal{A}} - \mathbf{w}_{\mathcal{A}})\|_2$ such that
        $\mathbf{v}_i \geq 0$ when $\boldsymbol{\beta}_i = 0$.
    Move $\boldsymbol{\beta}$ in direction $\mathbf{v}$ until
        an entry goes negative, OR
        new variable(s) join the set $\mathcal{A}$
    Update $\mathbf{c} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$.
end

KKT conditions

For each λ we want to find

$$\min_{\boldsymbol{\beta}} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2 \text{ such that } \boldsymbol{\beta} \geq \mathbf{0} \text{ and } \|\boldsymbol{\beta}\|_1 = \lambda. \qquad (\dagger)$$

From KKT conditions:

$\boldsymbol{\beta}$ solves $(\dagger)$ if and only if $\boldsymbol{\beta}$ is feasible and $\mathbf{c}_i$ is maximal for all i with $\boldsymbol{\beta}_i > 0$.

One step

Consider moving $\boldsymbol{\beta}$ in direction $\mathbf{v}$. For γ define

$$\boldsymbol{\beta}^{\gamma} = \boldsymbol{\beta} + \gamma\mathbf{v}$$

so that

$$\mathbf{c}^{\gamma} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\beta^{\gamma}) = \mathbf{c} - \gamma\mathbf{X}'\mathbf{X}\mathbf{v}.$$

The conditions that $\boldsymbol{\beta}^{\gamma}$ and $\mathbf{c}^{\gamma}$ need to satisfy are then

- Feasibility: $\boldsymbol{\beta}^{\gamma} \geq 0$.
- Optimality: $\mathbf{c}_i^{\gamma}$ maximal for all i such that $\boldsymbol{\beta}_i^{\gamma} > 0$.
- Increasing: $\sum \boldsymbol{\beta}^{\gamma} > \sum \boldsymbol{\beta}$.

Defining the search direction

Define $\mathcal{A} = \{i : \mathbf{c}_i \text{ maximal}\}$. LARS-Lasso algorithm considers the (unconstrained) search direction $\mathbf{w}$, where $\mathbf{w}_\mathcal{A} = (\mathbf{X}'_\mathcal{A}\mathbf{X}_\mathcal{A})^{-1}\mathbf{1}$.

From above, the actual search direction should be $\mathbf{v}$ satisfying

- For all $i \in \mathcal{A}$ such that $\boldsymbol{\beta}_i = 0$, $\mathbf{v}_i \geq 0$.
- For all $i \in \mathcal{A}$ such that $\boldsymbol{\beta}_i > 0$ or $\mathbf{v}_i > 0$, $(\mathbf{X}'\mathbf{X}\mathbf{v})_i = 1$.
- For all $i \in \mathcal{A}$, $(\mathbf{X}'\mathbf{X}\mathbf{v})_i \geq 1$.
- For all $i \notin \mathcal{A}$, $\mathbf{v}_i = 0$.

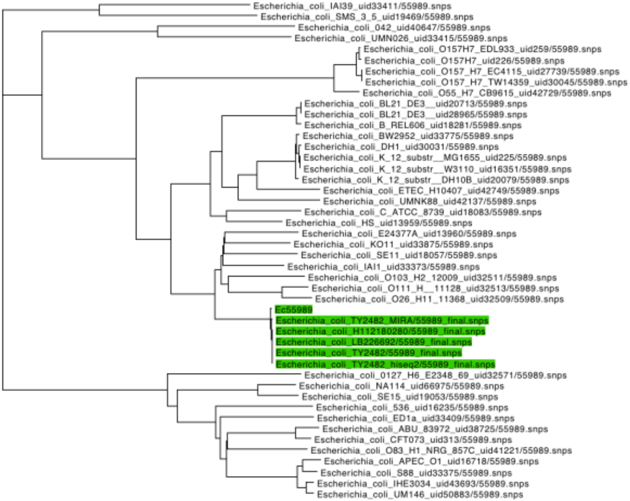
These are the KKT conditions for the constrained problem

$$\min_{\mathbf{v}} \|\mathbf{X}(\mathbf{v}_\mathcal{A} - \mathbf{w}_\mathcal{A})\| \text{ such that } \boldsymbol{\beta}_i = 0 \Rightarrow \mathbf{v}_i \geq 0$$

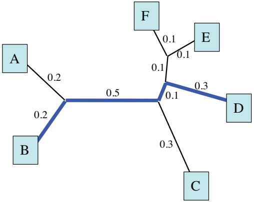
where $\mathbf{v}_i = 0$ for all $i \notin \mathcal{A}$.

Let $\boldsymbol{\beta} = \mathbf{0}$ and $\mathbf{c} = \mathbf{X}'\mathbf{y}$.
while $\|\mathbf{c}\| > 0$
    $\mathcal{A} = \{i : \mathbf{c}_i \text{ is maximum}\}$.
    Choose the search direction $\mathbf{w}_{\mathcal{A}} = (\mathbf{X}'_{\mathcal{A}}\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{1}$
    Find $\mathbf{v}$ minimizing $\|\mathbf{X}_{\mathcal{A}}(\mathbf{v}_{\mathcal{A}} - \mathbf{w}_{\mathcal{A}})\|_2$ such that
        $\mathbf{v}_i \geq 0$ when $\boldsymbol{\beta}_i = 0$.
    Move $\boldsymbol{\beta}$ in direction $\mathbf{v}$ until
        an entry goes negative, OR
        new variable(s) join the set $\mathcal{A}$
    Update $\mathbf{c} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$.
end

Phylogenetics



Each edge in the tree corresponds to a split (bipartition) of the objects into two parts. These splits and their weights determine evolutionary distances between the objects:

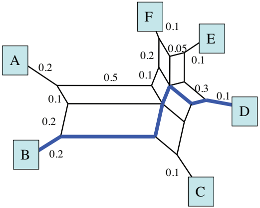


- $\mathbf{y}$ contains the observed distances between objects;
- $\mathbf{X}$ indicates which splits/branches separate which pairs;
- β is the vector of split/branch weights to be inferred.

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

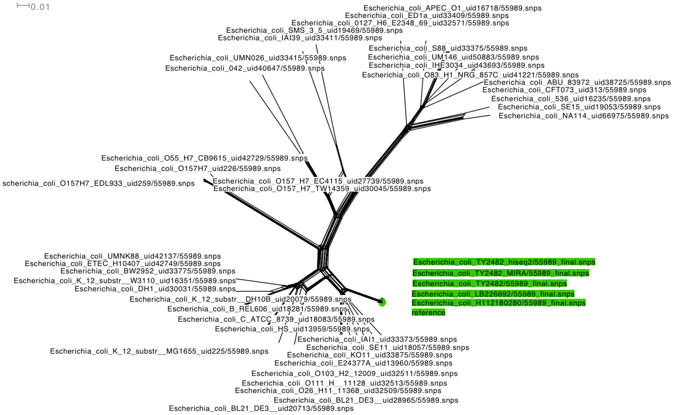
Application: phylogenetic networks

Most collections of splits do not encode a tree, however they can be represented using a split network.

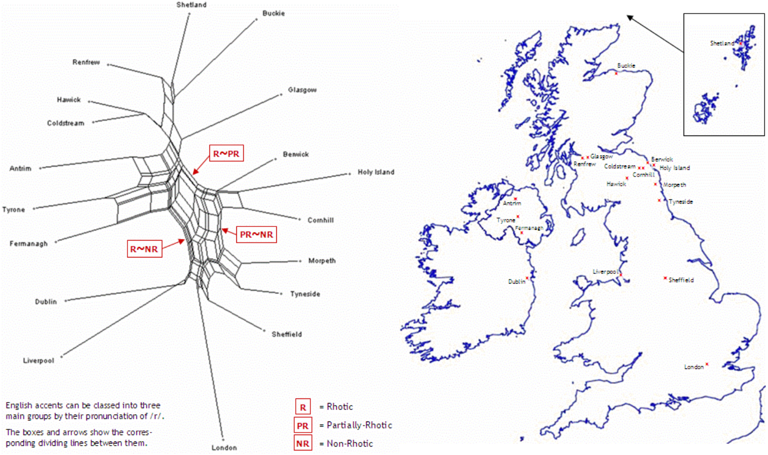


Useful for data exploration since we can depict conflicting signals, and represent the amount of noise*.

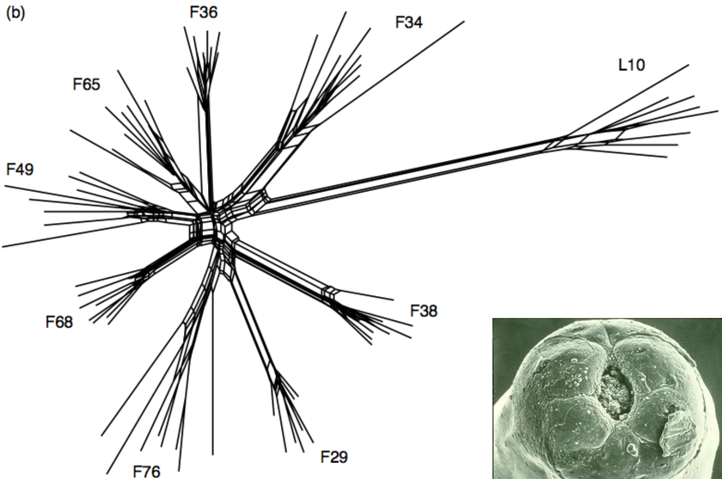
Phylogenetic networks



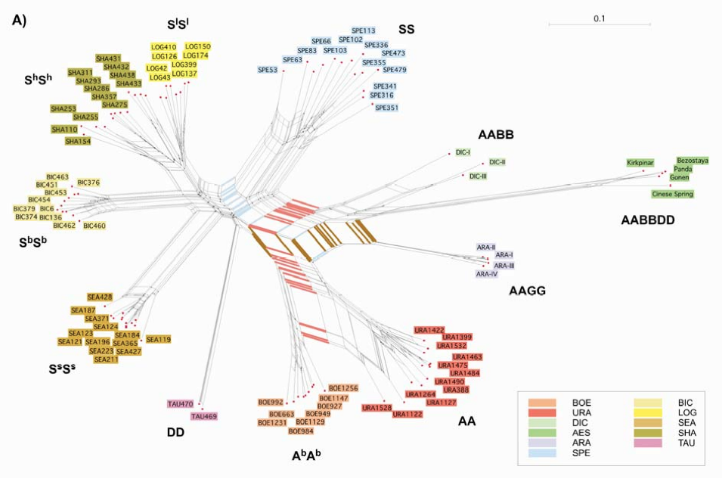
From English accents...



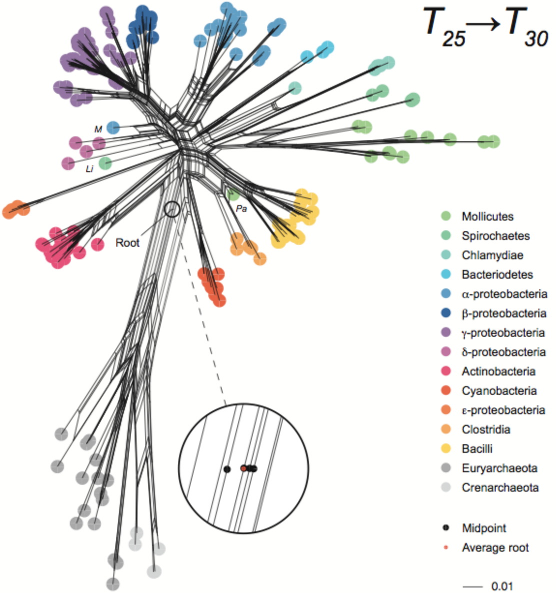
...to Swedish worms.



From the origin of modern wheat....



to the origin of life.



Networks and overfitting

With phylogenetic networks we intentionally over-fit the data.

In practice, many variables (splits) are eliminated using NNLS.

A large component of my student Alethea Rea's Ph.D. thesis was devoted to methods for cleaning up the remainder: the Lasso was an obvious choice.

Numerical issues

Let n be the number of objects. Then

- $\mathbf{X}$ is $\binom{n}{2} \times \binom{n}{2}$.
- $\mathbf{X}'\mathbf{X}$ typically poorly conditioned.
- $\mathbf{X}$ not sparse, but structured, so efficient algorithms for $\mathbf{X}\mathbf{v}$, $\mathbf{X}'\mathbf{v}$.
- $(\mathbf{X}'\mathbf{X})^{-1}$ sparse.

Let $\boldsymbol{\beta} = \mathbf{0}$ and $\mathbf{c} = \mathbf{X}'\mathbf{y}$.
while $\|\mathbf{c}\| > 0$
    $\mathcal{A} = \{i : \mathbf{c}_i \text{ is maximum}\}$.
    Choose the search direction $\mathbf{w}_{\mathcal{A}} = (\mathbf{X}'_{\mathcal{A}}\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{1}$
    Find $\mathbf{v}$ minimizing $\|\mathbf{X}_{\mathcal{A}}(\mathbf{v}_{\mathcal{A}} - \mathbf{w}_{\mathcal{A}})\|_2$ such that
        $\mathbf{v}_i \geq 0$ when $\boldsymbol{\beta}_i = 0$.
    Move $\boldsymbol{\beta}$ in direction $\mathbf{v}$ until
        an entry goes negative, OR
        new variable(s) join the set $\mathcal{A}$
    Update $\mathbf{c} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$.
end

Algorithm steps with numerical issues

Let $\boldsymbol{\beta} = \mathbf{0}$ and $\mathbf{c} = \mathbf{X}'\mathbf{y}$.
while $\|\mathbf{c}\| > 0$
    $\mathcal{A} = \{i : \mathbf{c}_i \text{ is maximum}\}$.
    Choose the search direction $\mathbf{w}_{\mathcal{A}} = (\mathbf{X}'_{\mathcal{A}}\mathbf{X}_{\mathcal{A}})^{-1}\mathbf{1}$
    Find $\mathbf{v}$ minimizing $\|\mathbf{X}_{\mathcal{A}}(\mathbf{v}_{\mathcal{A}} - \mathbf{w}_{\mathcal{A}})\|_2$ such that
        $\mathbf{v}_i \geq 0$ when $\boldsymbol{\beta}_i = 0$.
    Move $\boldsymbol{\beta}$ in direction $\mathbf{v}$ until
        an entry goes negative, OR
        new variable(s) join the set $\mathcal{A}$
    Update $\mathbf{c} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$.
end

Strategies

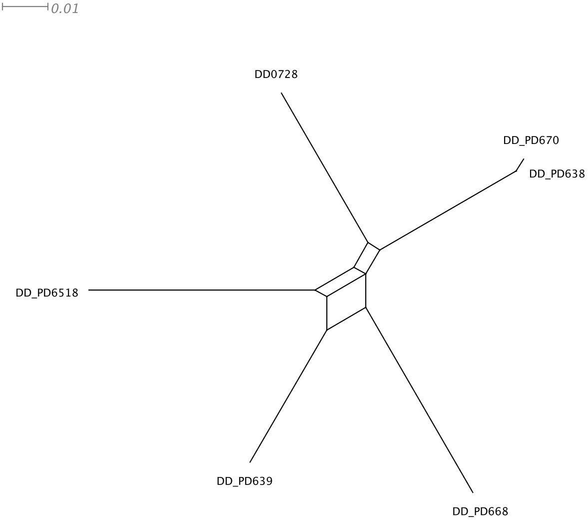
The key computation is the choice of search direction:

$$\min_{\mathbf{v}} \|\mathbf{X}_{\mathcal{A}}(\mathbf{v}_{\mathcal{A}} - \mathbf{w}_{\mathcal{A}})\|_2 \text{ such that } \beta_i = 0 \Rightarrow \mathbf{v}_i \geq 0.$$

Started with PGCG (thanks to John) but had problems with conditioning of $(\mathbf{X}'_{\mathcal{A}}\mathbf{X}_{\mathcal{A}})$ and with degeneracy

'Regressed' to an active set method. Made use of the sparseness of $(\mathbf{X}'\mathbf{X})^{-1}$, PCG and the Woodbury formula to solve sub-problems.

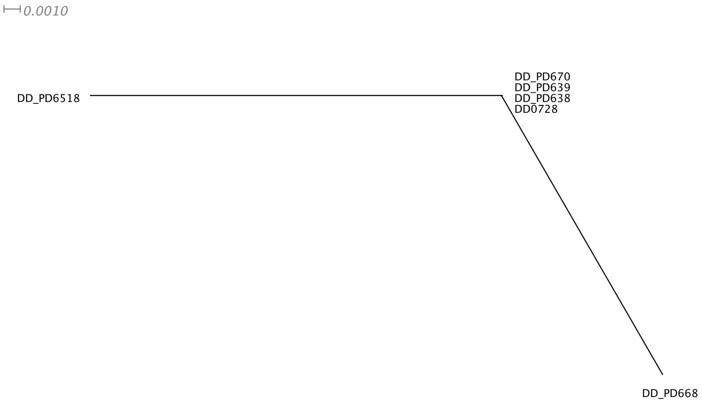
Simple example: full network



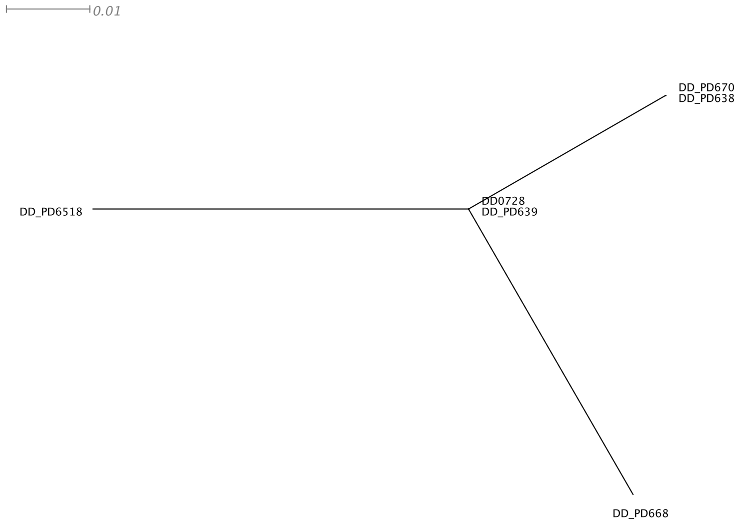
Simple example: lasso networks

Simple example: lasso networks

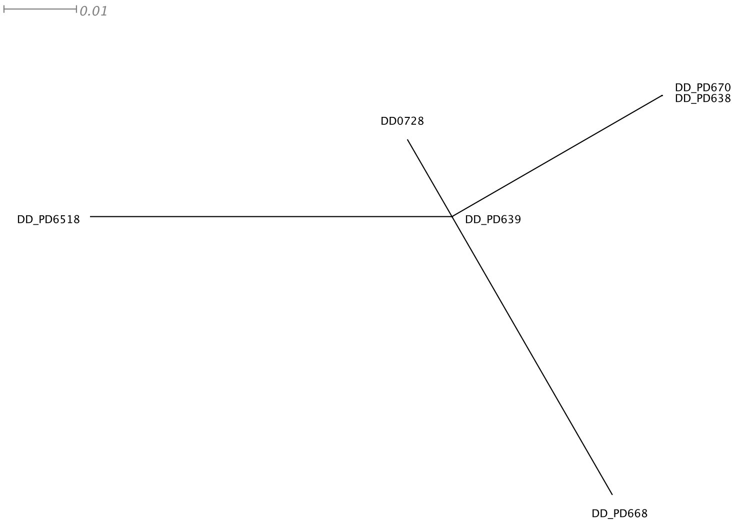
Simple example: lasso networks



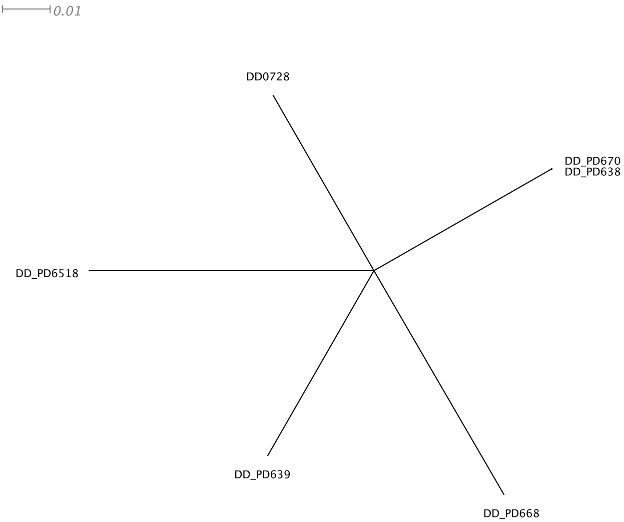
Simple example: lasso networks



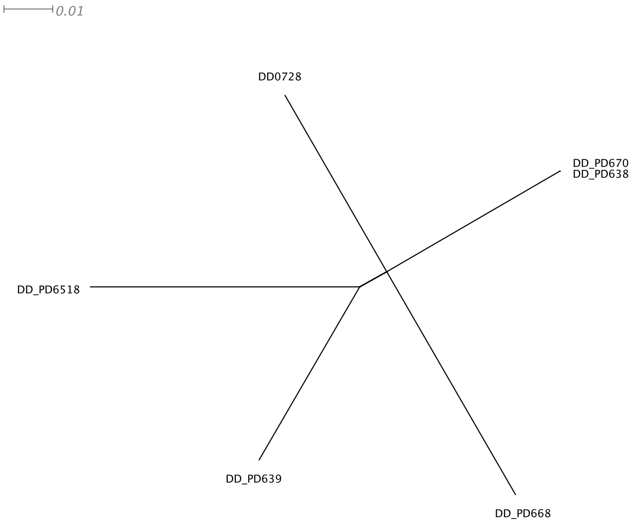
Simple example: lasso networks



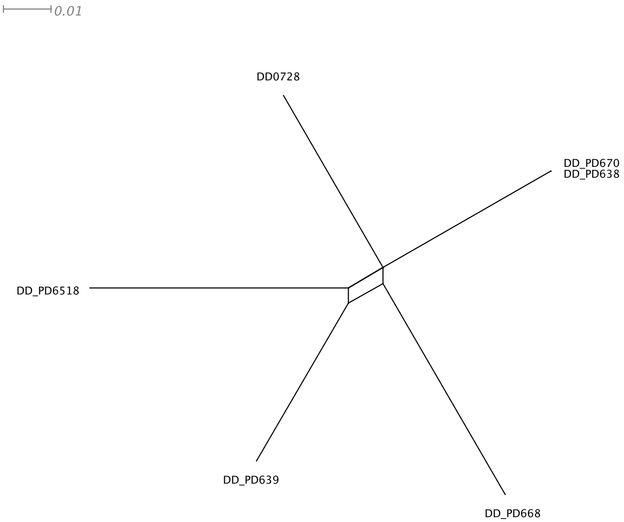
Simple example: lasso networks



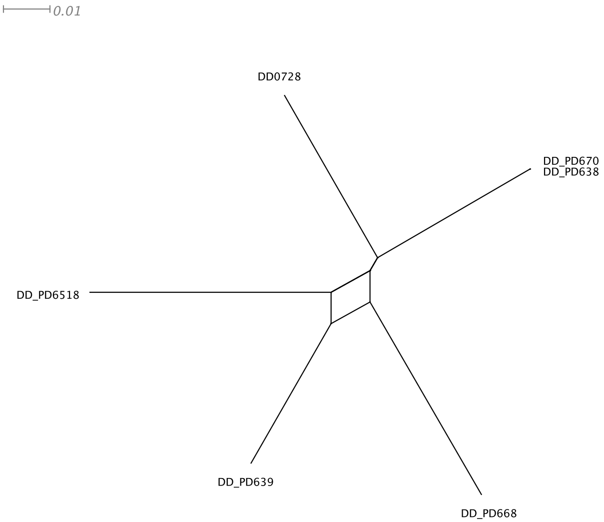
Simple example: lasso networks



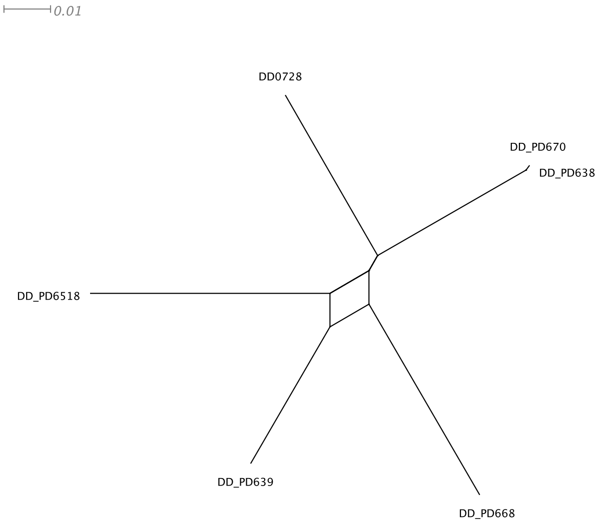
Simple example: lasso networks



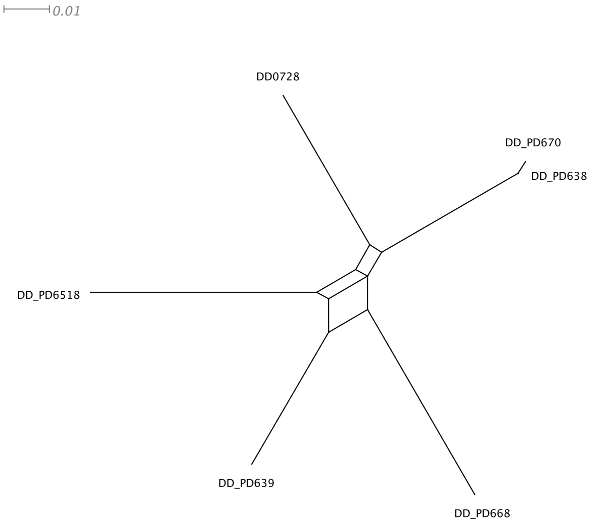
Simple example: lasso networks



Simple example: lasso networks



Simple example: lasso networks



Open problems

1. Making a choice of λ.
2. Weights within the penalty function (adaptive lasso?)
3. More general error distributions.

LASSO sampling

What I like most about the LARS and LARS-Lasso algorithm is that you effectively get an estimate for *all* possible values of λ: these are the points on the path $\boldsymbol{\beta}$.

Lasso-LARS sampler for

$$\pi(\boldsymbol{\beta}|\mathbf{y}, \lambda)$$

would produce a 'nice' function $\boldsymbol{\beta} : \Re \longrightarrow \Re^n$ such that for each λ, $\beta(\lambda)$ has the conditional marginal distribution

$$\boldsymbol{\beta}(\lambda) \sim \pi(\boldsymbol{\beta}|\mathbf{y}, \lambda).$$

Goal:

1. Sample $\boldsymbol{\beta}(\lambda_0)$ from $\pi(\boldsymbol{\beta}|\mathbf{y}, \lambda_0)$.
2. Remainder of $\boldsymbol{\beta}(\lambda)$ computed deterministically from $\boldsymbol{\beta}(\lambda_0)$ (e.g. numerically).

Summary

We propose a way to 'correct' the LARS-positive lasso algorithm to account for degeneracies.

Motivation was applications to phylogenetic networks, though we are exploring other applications.