High-dimensional Data: Prediction, Variable Selection and Applications in Computational Biology

Peter Bühlmann

Seminar für Statistik, ETH Zürich

May 2006

High-dimensional data

$(X_1, Y_1), \ldots, (X_n, Y_n)$ i.i.d. or stationary

X_i p-dimensional predictor variable

Y_i univariate response variable, e.g. $Y_i \in \mathbb{R}$ or $Y_i \in \{0, 1\}$

high-dimensional: $p \gg n$

areas of application:

astronomy, imaging, marketing research, text classification,...

biology, e.g. gene expressions with $p \approx 10'000$; $n \approx 10 - 100$

High-dimensional data

$(X_1, Y_1), \ldots, (X_n, Y_n)$ i.i.d. or stationary

X_i p-dimensional predictor variable

Y_i univariate response variable, e.g. $Y_i \in \mathbb{R}$ or $Y_i \in \{0, 1\}$

high-dimensional: $p \gg n$

areas of application:

astronomy, imaging, marketing research, text classification,...

biology, e.g. gene expressions with $p \approx 10'000$; $n \approx 10 - 100$

Two examples from computational biology

Splice site detection in DNA sequences

- predictor variables: 7 DNA bases with values in $\{A, C, G, T\}^7$
  dimension: $4^7 = 16'384$
- response variable which encodes whether a site (position in DNA) is a splice site or not
- sample size is $n \approx 11'000$ but could be much lower (for other organisms than humans)

Alternative splicing in genes

- 5 (or 9) exons and knowledge whether they have spliced or not
  $\rightsquigarrow$ contingency table with 5 (or 9) factors each having two levels
  dimensionality: $2^5 = 32$ (but with empty cells already) or $2^9 = 512$
- sample size: $n \approx 170$

Two examples from computational biology

Splice site detection in DNA sequences

- predictor variables: 7 DNA bases with values in $\{A, C, G, T\}^7$
  dimension: $4^7 = 16'384$
- response variable which encodes whether a site (position in DNA) is a splice site or not
- sample size is $n \approx 11'000$ but could be much lower (for other organisms than humans)

Alternative splicing in genes

- 5 (or 9) exons and knowledge whether they have spliced or not
  $\rightsquigarrow$ contingency table with 5 (or 9) factors each having two levels
  dimensionality: $2^5 = 32$ (but with empty cells already) or $2^9 = 512$
- sample size: $n \approx 170$

High-dimensional linear models

$$Y_i = (\beta_0 +) \sum_{j=1}^{p} \beta_j X_i^{(j)} + \epsilon_i,\ i = 1, \ldots, n$$

$p \gg n$

in short: $Y = X\beta + \epsilon$

goals:

- prediction, e.g. squared prediction error
- variable selection
  estimating the effective variables
  (having corresponding coefficient $\neq 0$)

High-dimensional linear models

$$Y_i = (\beta_0 +) \sum_{j=1}^{p} \beta_j X_i^{(j)} + \epsilon_i,\ i = 1, \ldots, n$$

$p \gg n$

in short: $Y = X\beta + \epsilon$

goals:

- prediction, e.g. squared prediction error
- variable selection
  estimating the effective variables
  (having corresponding coefficient $\neq 0$)

Approaches include:

Variable selection via AIC, BIC, (g)MDL (in a forward manner)
Bayesian methods for regularization
...

for example with AIC (and known error variance $\sigma^2 = 1$):
for every sub-model $\mathcal{M}$

$$AIC(\mathcal{M}) = \sum_{i=1}^{n}(Y_i - \underbrace{X_{\mathcal{M}}\hat{\beta}_{OLS;\mathcal{M}}}_{\text{in model } \mathcal{M}})^2 + 2(\text{no. of parameters}\,(\mathcal{M}))$$

$$\text{best model} \;=\; \text{minimizer of } AIC(\mathcal{M})$$

but:

there are 2^p sub-models and
we “cannot easily” explore the space of possible sub-models
(this also applies to MCMC techniques in Bayesian statistics)

computational feasibility for high-dimensional problems $\rightsquigarrow$

$$\Leftrightarrow \quad \text{(quasi-) convex optimization}$$

$$\text{(relaxed)} \quad \underbrace{\text{Lasso}}_{\text{Tibshirani (1996)}}$$

Lasso for linear models

$$\hat{\beta}(\lambda) = \text{argmin}_\beta(n^{-1}\|Y - X\beta\|^2 + \underbrace{\lambda}_{\geq 0} \underbrace{\|\beta\|_1}_{\sum_{j=1}^p |\beta_j|})$$

Tibshirani (1996)

- convex optimization problem $\rightsquigarrow$ feasible to compute
- does variable selection, i.e.

$$\hat{\beta}_j(\lambda) = 0 \text{ for some } j\text{'s, depending on } \lambda$$

(because of ℓ^1-norm geometry)

Lasso $=$ convexization of computationally hard problem for variable selection

more on computation:
LARS algorithm (Efron, Hastie, Johnstone, Tibshirani (2004))
Lasso solutions for all λ's can be computed in

$$O(np \min(n, p)) \text{ essential operations}$$

linear in dimensionality p if $p \gg n$

instead of looking at all 2^p sub-models...!

why solutions for all λ's?
$\rightsquigarrow$ cross-validation to pick a good λ
(and we consider all possible candidate values of λ)

in summary:

- Lasso is computationally great
- statistical properties and justification...? $\rightsquigarrow$ next minutes

The prediction problem

statistical notion of
high-dimensionality is relative to sample size n

mathematical formulation and conceptually useful:
dimension $p = p_n$

if p_n is fast growing function in $n \Leftrightarrow$ “high-dimensional”

Theorem (Greenshtein & Ritov, 2004)

- linear model with $p = p_n = O(n^\alpha)$ for some $\alpha < \infty$ (high-dimensional)
  e.g. $n = 100$, $p = p_n = 10'000$
- $\|\beta\|_1 = \|\beta_n\|_1 = \sum_{j=1}^{p_n} |\beta_{j,n}| = o((n/\log(n))^{1/4})$ (sparse)
  e.g. number of effective variables not growing too fast
- other minor conditions

Then, for suitable $\lambda = \lambda_n$,

$$\mathbb{E}_X[(\underbrace{\hat{f}(X)}_{\hat{\beta}(\lambda)^T X} - \underbrace{f(X)}_{\beta^T X})^2] \longrightarrow 0 \text{ in probability } (n \to \infty)$$

Choice of λ in practice for prediction: use cross-validation

and Lasso performs “quite well” for prediction

binary lymph node classification using gene expressions:
a high noise problem
$n = 49$ samples, $p = 7130$ gene expressions

cross-validated misclassification error (2/3 training; 1/3 test)

Lasso	L_2Boosting	FPLR	Pelora	1-NN	DLDA	SVM
21.1%	17.7%	35.25%	27.8%	43.25%	36.12%	36.88%

multivariate gene selection

best 200 genes (Wilcoxon test)
no additional gene selection

Lasso selected on CV-average 13.12 out of $p = 7129$ genes

The variable selection problem

$$Y_i = (\beta_0 +) \sum_{j=1}^{p} \beta_j X_i^{(j)} + \epsilon_i, \; i = 1, \ldots, n$$

goal: find the effective predictor variables
i.e. the set $\mathcal{E}_{true} = \{j;\ \beta_j \neq 0\}$

use the Lasso: $\hat{\mathcal{E}}(\lambda) = \{j;\ \hat{\beta}_j(\lambda) \neq 0\}$

as mentioned before: computationally very efficient
for binary lymph node classification with $n = 49,\ p = 7130$
computation of Lasso solutions for all λ's:

CPU time: 2.609 seconds using `lars` in `R`

Properties of $\hat{\mathcal{E}}(\lambda)$

Theorem (Meinshausen & PB, 2004)

- Y, $X^{(j)}$'s Gaussian (not crucial)
- LfV condition (LfV = **L**asso **f**or **V**ariable selection)
  see also Zhao & Yu (2006)
- $p(n) = O(n^{\alpha})$ for some $0 < \alpha < \infty$ (high-dimensionality)
- $|\mathcal{E}_{true,n}| = O(n^{\kappa})$ for some $0 < \kappa < 1$ (sparsity)
- other minor conditions

Then: for suitable $\lambda = \lambda_n$,

$$\mathbb{P}[\hat{\mathcal{E}}(\lambda) = \mathcal{E}_{true}] = 1 - O(\exp(-Cn^{1-\delta})) \longrightarrow 1 \ (n \to \infty)$$

statistical (asymptotic) justification of convexization of computationally hard problem for variable selection

Properties of $\hat{\mathcal{E}}(\lambda)$

Theorem (Meinshausen & PB, 2004)

- Y, $X^{(j)}$'s Gaussian (not crucial)
- LfV condition (LfV = **L**asso **f**or **V**ariable selection)
  see also Zhao & Yu (2006)
- $p(n) = O(n^{\alpha})$ for some $0 < \alpha < \infty$ (high-dimensionality)
- $|\mathcal{E}_{true,n}| = O(n^{\kappa})$ for some $0 < \kappa < 1$ (sparsity)
- other minor conditions

Then: for suitable $\lambda = \lambda_n$,

$$\mathbb{P}[\hat{\mathcal{E}}(\lambda) = \mathcal{E}_{true}] = 1 - O(\exp(-Cn^{1-\delta})) \longrightarrow 1 \ (n \to \infty)$$

statistical (asymptotic) justification of convexization of computationally hard problem for variable selection

the method and theory immediately generalizes to
Gaussian Graphical Modeling

i.e. the Lasso can be used to estimate
high-dimensional Gaussian graphical models

LfV condition is restrictive

sufficient and necessary for consistent model selection with Lasso

it fails to hold if design matrix is “too correlated”

$\Rightarrow$ Lasso is not consistent anymore for selecting the true model

The LfV condition: a condition on the covariance of X

$$\underbrace{\text{LfV condition}}_{\text{Meinshausen \& PB (2004)}} \quad \Leftrightarrow \quad \underbrace{\text{Irrepresentable condition}}_{\text{Zhao \& Yu (2006)}}$$

$''\Leftrightarrow''$ Lasso is consistent for variable selection

Irrepresentable condition $\Leftrightarrow |\hat{\Sigma}_{noise;eff}\hat{\Sigma}^{-1}_{eff;eff}\text{sign}(\beta_{eff})| \leq 1 - \eta$

it holds for

- $\hat{\Sigma}_{ij} \leq \rho^{|i-j|}$ $(0 \leq \rho < 1)$ power decay correlations
- dictionaries with $\underbrace{\text{coherence}}_{\text{max. correlation}} < (2p_{eff} - 1)^{-1}$
  (notion of coherence: Donoho, Elad & Temlyakov (2004))
- easy to construct examples where condition fails to hold

Choice of λ

first (not so good) idea: choose λ to optimize prediction
e.g. via some cross-validation scheme

but: for prediction oracle solution

$$\lambda^* = \text{argmin}_\lambda \mathbb{E}[(Y - \sum_{j=1}^{p} \hat{\beta}_j^{(}\lambda) X^{(j)})^2]$$

$\mathbb{P}[\hat{\mathcal{E}}(\lambda^*) = \mathcal{E}_{true}] < 1 \ (n \to \infty)$ (or $= 0$ if $p_n \to \infty \ (n \to \infty)$)

asymptotically: prediction optimality yields too large models
(Meinshausen & PB, 2004; related example by Leng et al., 2004)

in summary:

- prediction optimal solution yields asymptotically too large models
- if LfV condition fails to hold (and assuming weaker conditions)
  Lasso yields models which contain the true model

$\rightsquigarrow$ Lasso can be used as
a "filter for variable selection" i.e. true model is contained in selected models from Lasso

Binary lymph node classification in breast cancer: $n = 49$ $p = 7130$

5-fold CV tuned Lasso selects 23 genes (on whole data set)

note (in practice): identifiability problem among highly correlated predictor variables

$\rightsquigarrow$ an ad-hoc approach:
keep the 23 plus all its highly correlated genes for further modeling, interpretation etc...

From filtering to selection of variables

with Lasso, we obtain sequence of sub-models

$$\widehat{\mathcal{SUB}} = \{\hat{\mathcal{E}}(\lambda_r);\ 1 \leq r \leq \underbrace{r_{max}}_{=O(\min(n,p))}\},\ \lambda_1 = 0 < \lambda_2 < \ldots < \lambda_{max}$$

i.e. not very many sub-models anymore

typically

$$\hat{\mathcal{E}}(\lambda_{max}) \subset \ldots \subset \hat{\mathcal{E}}(\lambda_2) \subset \hat{\mathcal{E}}(\lambda_1)$$

assuming the LfV and other conditions:
with high probability,

$$\mathcal{E}_{true} \in \widehat{\mathcal{SUB}},$$
$$(\text{and } \mathcal{E}_{true} \subseteq\ \hat{\mathcal{E}}(\lambda^*))$$

$\rightsquigarrow$ we only need a good selector within $\widehat{\mathcal{SUB}}$

first (empirically not so good idea): choose best model in $\widehat{\mathcal{SUB}}$ using BIC or related method

better:
use the Lasso again for the models in $\widehat{\mathcal{SUB}}$:

$$\underbrace{\hat{\mathcal{E}}(\lambda_{max})}_{\rightsquigarrow \text{Lasso again}} \quad \underbrace{\hat{\mathcal{E}}(\lambda_{r_{max}-1})}_{\rightsquigarrow \text{Lasso again}} \quad \cdots \quad \underbrace{\hat{\mathcal{E}}(\lambda_2)}_{\rightsquigarrow \text{Lasso again}} \quad \underbrace{\hat{\mathcal{E}}(\lambda_1)}_{\rightsquigarrow \text{Lasso again}}$$

this is the Relaxed Lasso (Meinshausen, 2005)

first (empirically not so good idea): choose best model in $\widehat{\mathcal{SUB}}$ using BIC or related method

better:
use the Lasso again for the models in $\widehat{\mathcal{SUB}}$:

$$\underbrace{\hat{\mathcal{E}}(\lambda_{max})}_{\rightsquigarrow \text{Lasso again}} \quad \underbrace{\hat{\mathcal{E}}(\lambda_{r_{max}-1})}_{\rightsquigarrow \text{Lasso again}} \quad \cdots \quad \underbrace{\hat{\mathcal{E}}(\lambda_2)}_{\rightsquigarrow \text{Lasso again}} \quad \underbrace{\hat{\mathcal{E}}(\lambda_1)}_{\rightsquigarrow \text{Lasso again}}$$

this is the Relaxed Lasso (Meinshausen, 2005)

Relaxed Lasso

for $\lambda \geq 0,\ 0 \leq \phi \leq 1$

$$\hat{\beta}_{\lambda,\phi} = \text{argmin}_\beta n^{-1} \sum_{i=1}^{n} (Y_i - \sum_{j \in \hat{\mathcal{E}}(\lambda)} \beta_j X_i^{(j)})^2 + \phi\lambda \|\beta\|_1$$

for $\phi = 0$: OLS on selected variables from Lasso(λ)
for $\phi = 1$: Lasso(λ)

amount of computation for finding all solutions over λ and ϕ:
often, the same computational complexity as for Lasso/LARS:

$$O(np \min(n,p)) = O(p) \text{ if } p \gg n$$

worst case: $O(np \min(n,p)^2) = O(p)$ if $p \gg n$ still linear in p

this is "quasi-convex" optimization
two levels of a convex problem

Relaxed Lasso

for $\lambda \geq 0,\ 0 \leq \phi \leq 1$

$$\hat{\beta}_{\lambda,\phi} = \text{argmin}_\beta n^{-1} \sum_{i=1}^{n} (Y_i - \sum_{j \in \hat{\mathcal{E}}(\lambda)} \beta_j X_i^{(j)})^2 + \phi\lambda \|\beta\|_1$$

for $\phi = 0$: OLS on selected variables from Lasso(λ)
for $\phi = 1$: Lasso(λ)

amount of computation for finding all solutions over λ and ϕ:
often, the same computational complexity as for Lasso/LARS:

$$O(np \min(n,p)) = O(p) \text{ if } p \gg n$$

worst case: $O(np \min(n,p)^2) = O(p)$ if $p \gg n$ still linear in p

this is "quasi-convex" optimization
two levels of a convex problem

Relaxed Lasso

for $\lambda \geq 0,\ 0 \leq \phi \leq 1$

$$\hat{\beta}_{\lambda,\phi} = \text{argmin}_\beta n^{-1} \sum_{i=1}^{n} (Y_i - \sum_{j \in \hat{\mathcal{E}}(\lambda)} \beta_j X_i^{(j)})^2 + \phi\lambda \|\beta\|_1$$

for $\phi = 0$: OLS on selected variables from Lasso(λ)
for $\phi = 1$: Lasso(λ)

amount of computation for finding all solutions over λ and ϕ:
often, the same computational complexity as for Lasso/LARS:

$$O(np \min(n,p)) = O(p) \text{ if } p \gg n$$

worst case: $O(np \min(n,p)^2) = O(p)$ if $p \gg n$ still linear in p

this is "quasi-convex" optimization
two levels of a convex problem

Properties of the relaxed Lasso

from Meinshausen (2005):

assume the LfV and other conditions

prediction optimal tuned relaxed Lasso
is consistent for variable selection

$\rightsquigarrow$ can use cross-validation to estimate λ
and such CV-estimated $\hat{\lambda}_{CV}$ is good for variable selection

for very high-dimensional case
($p = p_n \sim C_1 \exp(C_2 n^{1-\xi})$ $(0 < \xi < 1)$)

relaxed Lasso has much lower prediction error than Lasso

Properties of the relaxed Lasso

from Meinshausen (2005):

assume the LfV and other conditions

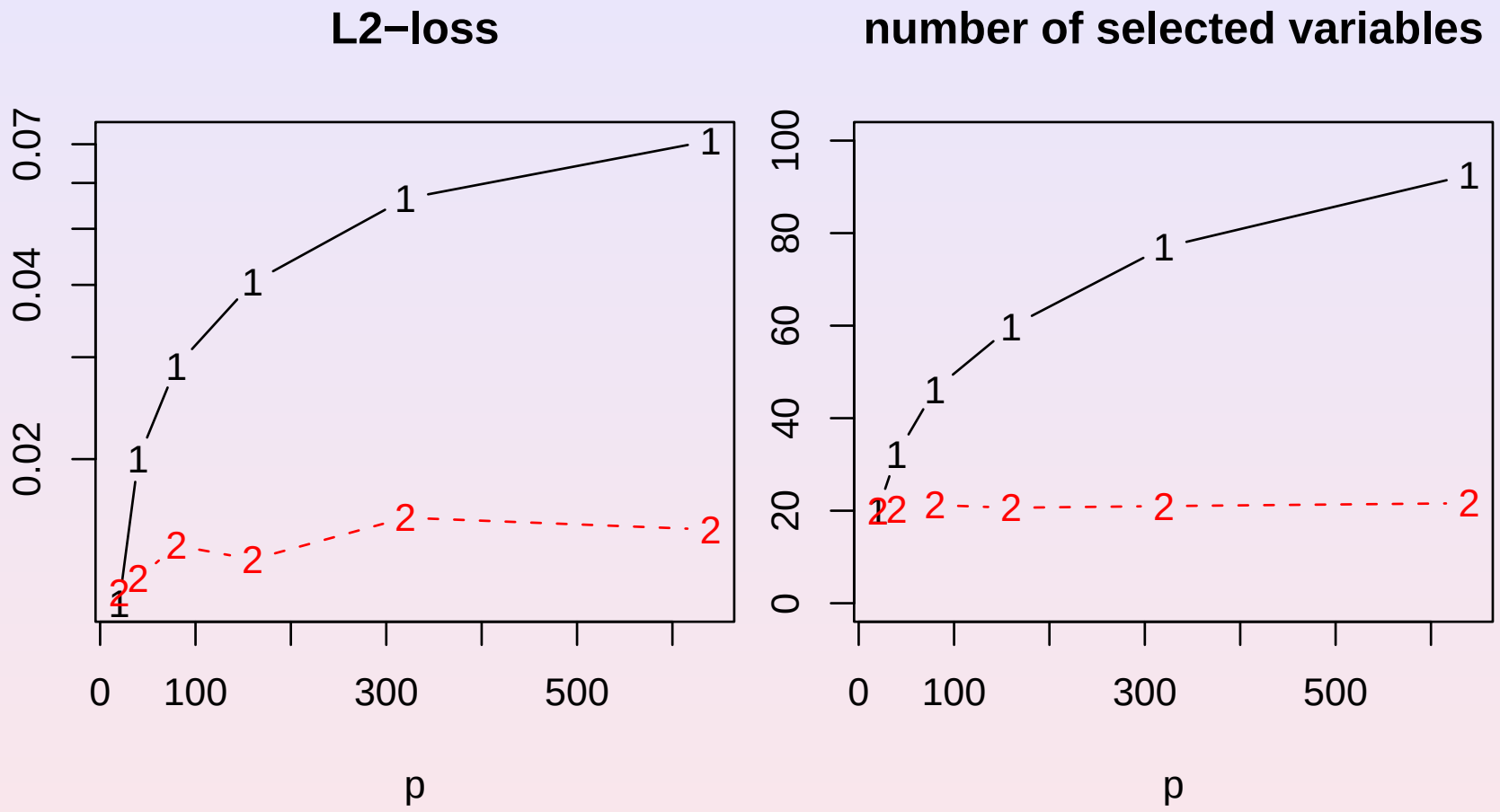
prediction optimal tuned relaxed Lasso
is consistent for variable selection

$\rightsquigarrow$ can use cross-validation to estimate λ
and such CV-estimated $\hat{\lambda}_{CV}$ is good for variable selection

for very high-dimensional case
($p = p_n \sim C_1 \exp(C_2 n^{1-\xi})$ $(0 < \xi < 1)$)

relaxed Lasso has much lower prediction error than Lasso

$n = 300, p = 20, \ldots 650, p_{eff} = 20$



1: Lasso 2: relaxed Lasso

additional pure noise variables are much less damaging with the relaxed Lasso than for Lasso

for prediction:

Relaxed Lasso never substantially worse than the Lasso

the price for the flexibility of the relaxed Lasso is

the larger search space $0 \leq \phi \leq 1$ (Lasso: $\phi = 1$)

for variable selection:

Relaxed Lasso (almost) always sparser than Lasso

Binary lymph node classification in breast cancer: $n = 49$ $p = 7130$

5-fold CV tuning for each method

cross-validated quantities (2/3 training; 1/3 test)

	misclassif. error	number of selected genes
Lasso	21.1%	13.12
Relaxed Lasso	20.1%	7.3

Binary lymph node classification in breast cancer: $n = 49$ $p = 7130$

5-fold CV tuning for each method

cross-validated quantities (2/3 training; 1/3 test)

	misclassif. error	number of selected genes
Lasso	21.1%	13.12
Relaxed Lasso	20.1%	7.3

DNA splice site detection

DNA sequence

$$\ldots ACGGC \ldots \quad \underbrace{NNN \quad \underbrace{GC}_{\text{potential donor site}} \quad NNNN}_{\text{3 positions exon } GC \text{ 4 positions intron}} \quad \ldots AAC \ldots$$

response $Y \in \{0, 1\}$: splice or non-splice site
predictor variables: 7 factors each having 4 levels
(full dimension: $4^7 = 16'384$)
data: $p = 16'384$, $n = 11'220$

training:	$5'610$ true splice sites
	$5'610$ non-splice sites
	plus an unbalanced validation set
test data:	$4'208$ true splice sites
	$89'717$ non-splice sites

logistic regression:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \text{ main effects} + \text{first order interactions} + \ldots$$

with sum to zero constraints

use “Lasso” which selects whole terms
instead of selection of dummy indicator variables

e.g. the interaction term between factor 2 and 5 (which is encoded with 9 free parameters/dummy indicators)

$\rightsquigarrow$ Group Lasso (Yuan and Lin (2006), for Gaussian regression)

$$\text{penalty: } \lambda \sum_{\text{term } j} \|\beta_j\|_2$$

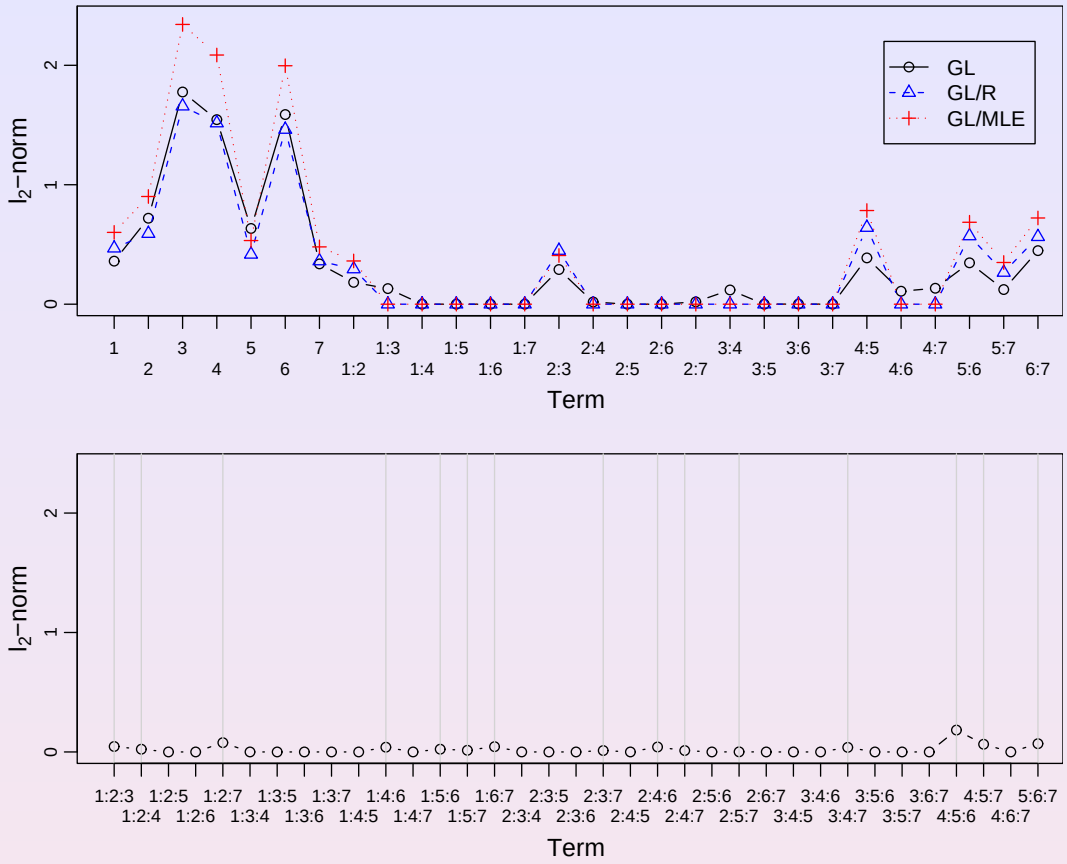
Group Lasso penalty:

- invariant under orthogonal reparametrization
- if term j has dimension 1: $\|\beta_j\|_2 = \|\beta_j\|_1$

- new efficient algorithms are needed for Group Lasso with binomial likelihood
  $\rightsquigarrow$ Block gradient descent with tight approximations for the Hessian
- theory and methodology for high-dimensions: "similar" as for the Lasso

(Meier, v.d. Geer & PB, 2006)

Group Lasso/Ridge: in spirit of the Relaxed Lasso
1st stage: Group Lasso for logistic regression
2nd stage: Ridge logistic regression on models from 1st stage
$\rightsquigarrow$ allows for hierarchical model fitting
$\rightsquigarrow$ better term selection and better prediction than Group Lasso



- mainly neighboring DNA positions show interactions (has been “known” and “debated”)
- no interaction among exons and introns

  (with Group Lasso/Ridge)
- no second-order interactions

  (with Group Lasso/Ridge)

predictive power:
competitive with “state to the art” maximum entropy modeling from Yeo and Burge (2004)

correlation between true and predicted class

Logistic Group Lasso/Ridge	0.6593
max. entropy (Yeo and Burge)	0.6589

- our model is simple (not necessarily the method/algorithm) and has clear interpretation
- it is as good or better than many of the complicated non-Markovian stochastic process models (e.g. Zhao, Huang and Speed (2004))

Alternative DNA splicing

DNA sequence: for a single gene

exon1 intron1 exon2 intron2 . . . exon5 intron5

"regular" splicing $\rightsquigarrow$ exon1 exon2 ... exon5
alternative splicing $\rightsquigarrow$ only some exons are spliced
(or spliced in a different order)

5 exons from gene "itpr1":
we know whether exons have been spliced or not
data from full length cDNA libraries

tissue from adult cerebrum in rats and different developmental stages of cerebellum in rats
(Emerick & Agnew, Johns Hopkins)

$\rightsquigarrow$ contingency table(s) with 5 factors (from 5 exons)
each having two levels (spliced or not)
the table has many empty cells ("high-dimensional")
(other problems involve 9 exons)

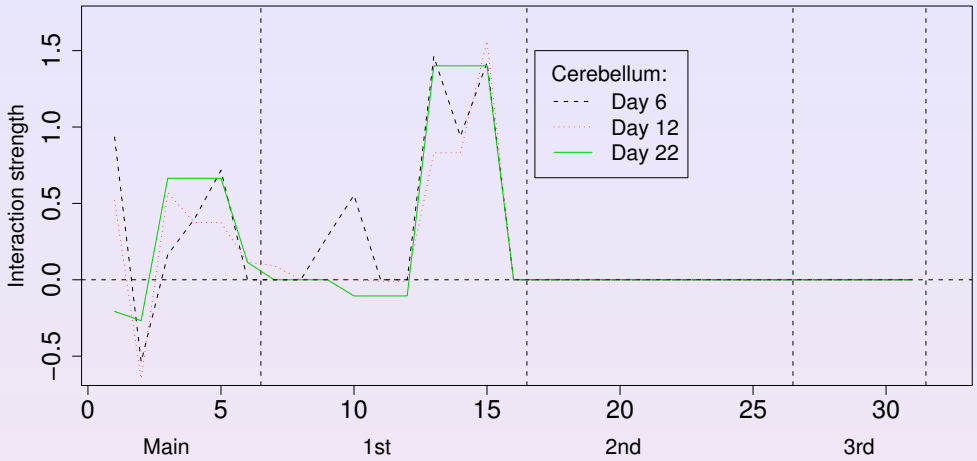
log-linear model for cell probabilities

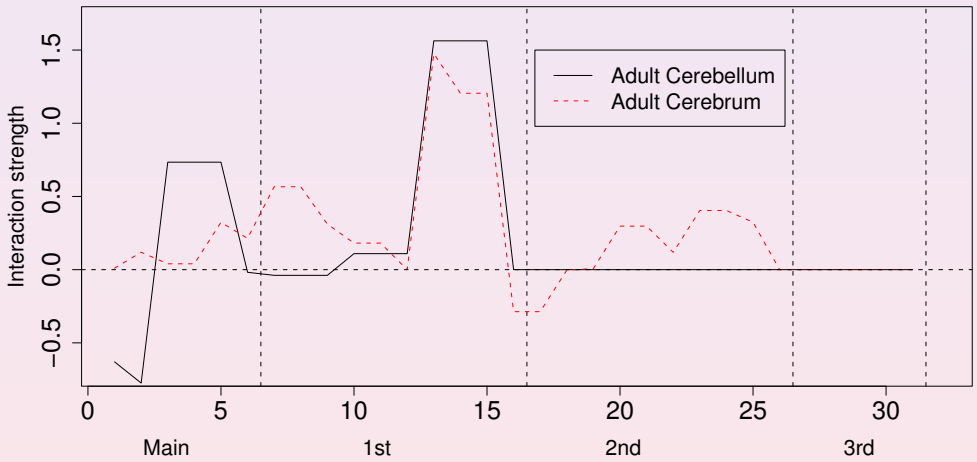
$$\log(\text{cell probability}) = \text{intercept} + \text{main effects} + \text{interaction terms}$$

with sum to zero constraints

use the relaxed Lasso for estimation $\rightsquigarrow$ selection of terms

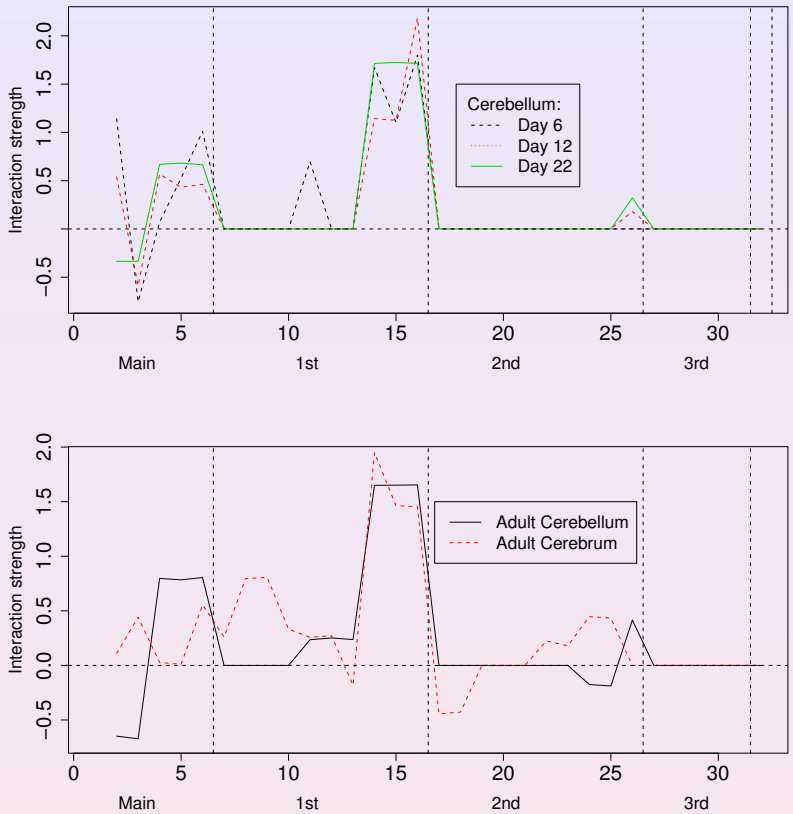
Dahinden, Emerick, Parmigiani & PB (2006)





- no second or higher-order interactions
- interaction pattern well conserved over different developmental stages

with hierarchical Bayesian modeling (a lot of computing...!)



- for suitable choice of the (one) hyperparameter maximum a posteriori (MAP) similar to relaxed Lasso
- for other choices of the hyperparameter: markedly different
  $\rightsquigarrow$ tune Bayesian model such that MAP $\approx$ relaxed Lasso

Bayesian model

$$\mathbf{n} \sim \text{Multinom}(\mathbf{p}), \quad \log(\mathbf{p}) = X\beta$$
$$\beta_j|\gamma_j \sim (1-\gamma_j)I_0 + \gamma_j \mathcal{N}(0, \sigma^2) \text{ independent for all } j\text{'s}$$
$$\gamma_j \sim \text{Bernoulli}(1/2) \text{ independent for all } j\text{'s}$$
$$\sigma^2 = 1 \quad (\text{or } \sigma^2 \sim \Gamma^{-1}(2,3))$$

design matrix X encoded with dummies
sum-to-zero constraints for parameters

for hierarchical models:

- zero coefficients can be interpreted in terms of conditional independence
- invariant under reparametrization
  zero term remains zero term

in both biology problems:

we are “in a better position” to estimate
whether higher-order interactions exist or not

without good regularization and variable selection methods:
difficult to answer

Your own high-dimensional problem...

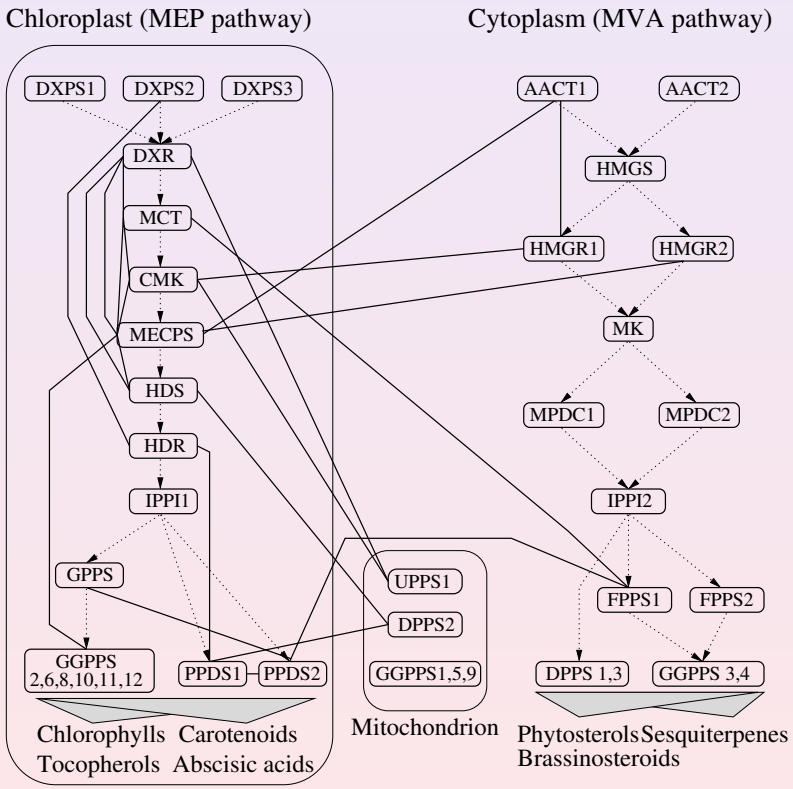
Two biosynthesis pathways in Arabidopsis Thaliana: associations among 39 genes from $n = 118$ microarray exper.

Wille et al. (2004)

Your own high-dimensional problem...

Two biosynthesis pathways in Arabidopsis Thaliana: associations among 39 genes from $n = 118$ microarray exper.

Wille et al. (2004)



Conclusions

especially for high-dimensional data:

- Lasso useful for variable filtering
  it is computationally attractive: linear in dimensionality p
  the "true model" is contained in the solution set of Lasso

- Relaxed Lasso (or similar two stage procedures)
  often better prediction than Lasso
  optimal penalty for prediction $\rightsquigarrow$ consistent model selection
  sparser solutions than Lasso

- Software: efficient implementations in R
  LARS algorithm for linear models (Hastie)
  Group Lasso and Lasso for generalized linear models (Meier)

Conclusions

especially for high-dimensional data:

- Lasso useful for variable filtering
  it is computationally attractive: linear in dimensionality p
  the "true model" is contained in the solution set of Lasso

- Relaxed Lasso (or similar two stage procedures)
  often better prediction than Lasso
  optimal penalty for prediction $\rightsquigarrow$ consistent model selection
  sparser solutions than Lasso

- Software: efficient implementations in R
  LARS algorithm for linear models (Hastie)
  Group Lasso and Lasso for generalized linear models (Meier)

Conclusions

especially for high-dimensional data:

- Lasso useful for variable filtering
  it is computationally attractive: linear in dimensionality p
  the "true model" is contained in the solution set of Lasso

- Relaxed Lasso (or similar two stage procedures)
  often better prediction than Lasso
  optimal penalty for prediction $\rightsquigarrow$ consistent model selection
  sparser solutions than Lasso

- Software: efficient implementations in R
  LARS algorithm for linear models (Hastie)
  Group Lasso and Lasso for generalized linear models (Meier)

Conclusions

especially for high-dimensional data:

- Lasso useful for variable filtering
  it is computationally attractive: linear in dimensionality p
  the "true model" is contained in the solution set of Lasso

- Relaxed Lasso (or similar two stage procedures)
  often better prediction than Lasso
  optimal penalty for prediction $\rightsquigarrow$ consistent model selection
  sparser solutions than Lasso

- Software: efficient implementations in R
  LARS algorithm for linear models (Hastie)
  Group Lasso and Lasso for generalized linear models
  (Meier)