Chapter 2 (week 2)

6 Feb 2023

Table of contents

“God is in every leaf of every tree”

- From Andrew Gelman (blog)
- “No problem is too small or too trivial if we really do something about it.” (Dyson (2005) quoting Richard Feynman)
- (an excuse for going down rabbit holes?)

Dyson, Freeman. 2005. “Wise Man.” *New York Review of Books*, October. https://www.nybooks.com/articles/2005/10/20/wise-man/.

feature selection

- *feature* ≈ a column of the model matrix
- termwise selection, e.g.
  - all columns associated with a categorical variable
  - all columns of a basis expansion (polynomial etc.) of a continuous variable
- columnwise selection
  - fine for prediction
  - silly for inference?
- selection maintaining the **principle of marginality** (Venables 1998)
  (i.e., don’t drop lower-order effects from a model containing interactions)
- ¿ a way to **merge categories** on the fly (based on rarity, correlation, predictive ability)?

Venables, W. N. 1998. “Exegeses on Linear Models.” In. 1998 International S-PLUS User Conference. Washington, DC. http://www.stats.ox.ac.uk/pub/MASS3/Exegeses.pdf.

why select?

- save memory
- save “flops” (floating-point operations)
- optimize bias-variance tradeoff

- optimize data collection
- parsimonious/simple explanations (e.g. `rms::fastbw` in R)

why select (2)?

- save memory: OK
- save flops, optimize B-V
  - which is best: soft (ridge), semi-soft (lasso/SCAD), hard (stepwise/subset) penalization?

selection: filters, wrappers, embedded methods

Jović, Brkić, and Bogunović (2015)

Jović, A., K. Brkić, and N. Bogunović. 2015. "A Review of Feature Selection Methods with Applications." In *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1200–1205. https://doi.org/10.1109/MIPRO.2015.7160458.

- **filters**: standalone recipes
  - e.g. minimum-redundancy maximum relevance (mrMR) (Peng, Long, and Ding 2005)
    - similar to stepwise forward, but no estimation done (compute mutual information)
    - **greedy**
  - general, low-cost

Peng, Hanchuan, Fuhui Long, and C. Ding. 2005. "Feature Selection Based on Mutual Information Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (8): 1226–38. https://doi.org/10.1109/TPAMI.2005.159.

- wrappers: applied around specific methods
  - e.g. stepwise regression
  - general, evaluates prediction
- embedded methods: integrate estimation and selection
  - e.g. lasso etc.
  - most efficient? can combine shrinkage and selection

stepwise abuse

- stepwise regression for **prediction** may be fine (Murtaugh 2009)
  - selection based on AIC etc. more sensible than with p-values

Murtaugh, Paul A. 2009. "Performance of Several Variable-Selection Methods Applied to Real Ecological Data." *Ecology Letters* 12 (10): 1061–68. https://doi.org/10.1111/j.1461-0248.2009.01361.x.

  - note $\Delta AIC \propto p-$ value, if using columnwise/1-df steps
    * $\Delta \log(L) \leftrightarrow \Delta AIC = 0 \leftrightarrow p = 0.16$
    * leave-one-out cross-validation (LOOCV) asymptotically equiv. to AIC (Stone (1977); but see CV)
- for **inference**, terrible if done naively (but see Blanchet, Legendre, and Borcard (2008))
  - see CrossValidated
  - unstable, biased estimates; overconfident inference ("snooping")
- ESL: stepwise as a jumping-off point/comparator for different

Stone, M. 1977. "An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion." *J. Royal Stat. Soc. B* 39 (1): 44–47. https://www.jstor.org/stable/2984877.

Blanchet, F. Guillaume, Pierre Legendre, and Daniel Borcard. 2008. "Forward Selection of Explanatory Variables." *Ecology* 89 (9): 2623–32. https://doi.org/10.1890/07-0986.1.

POLLS

- did you learn to do stepwise regression in a class? Were you warned about its limitations?
- have you used stepwise regression? were you aware of its limitations at the time?
- have you used SR "in real life"? for prediction or inference?

contrasts for categorical variables

- expanding categorical variables to dummy variables
- automatically handled by `model.matrix()` in R (`StatsModels.jl:modelmatrix` in Julia)

```r
library(palmerpenguins)
library(tidyverse)
library(faux)
set.seed(101)
pp <- penguins[sample(nrow(penguins)), c("species", "island")]  ## scramble
head(model.matrix(~species+island, pp))
```

```
  (Intercept) speciesChinstrap speciesGentoo islandDream islandTorgersen
1           1                1             0           1               0
2           1                1             0           1               0
3           1                0             0           1               0
4           1                0             1           0               0
5           1                1             0           1               0
6           1                1             0           1               0
```

```
## faux makes nicer factors!
## rename variables/**idempotent** operations: f(f(x)) = f(x)  x
pp2 <- mutate(pp, across(where(is.factor), contr_code_treatment))
head(model.matrix(~species+island, pp2))
```

```
  (Intercept) species.Chinstrap-Adelie species.Gentoo-Adelie
1           1                        1                     0
2           1                        1                     0
3           1                        0                     0
4           1                        0                     1
5           1                        1                     0
6           1                        1                     0
  island.Dream-Biscoe island.Torgersen-Biscoe
1                   1                       0
2                   1                       0
3                   1                       0
4                   0                       0
5                   1                       0
6                   1                       0
```

```
colnames(model.matrix(~species*island, pp2))
```

```
[1] "(Intercept)"
[2] "species.Chinstrap-Adelie"
[3] "species.Gentoo-Adelie"
[4] "island.Dream-Biscoe"
[5] "island.Torgersen-Biscoe"
[6] "species.Chinstrap-Adelie:island.Dream-Biscoe"
[7] "species.Gentoo-Adelie:island.Dream-Biscoe"
```

```
[8] "species.Chinstrap-Adelie:island.Torgersen-Biscoe"
[9] "species.Gentoo-Adelie:island.Torgersen-Biscoe"
```

- identifiability constraints: leave out one category
  - post-hoc evaluation (e.g. `emmeans` R pkg)
  - penalized methods

regression, again

- hat matrix ($\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$) as *projection matrix* from $\mathbf{R}^N$ to $\mathbf{R}^p$
  - (what if we first transformed $\mathbf{X}$ to be orthonormal?)
- **non-full-rank** case ($\text{rank}(\mathbf{X}) < p$)
  - non-unique solutions
  - may break our linear algebra, depending on what we use

```r
X <- matrix(c(1:3, 2*(1:3)), ncol = 2)
y <- 1:3
Matrix::rankMatrix(X)
```

```
[1] 1
attr(,"method")
[1] "tolNorm2"
attr(,"useGrad")
[1] FALSE
attr(,"tol")
[1] 6.661338e-16
```

```r
try(solve(X %*% t(X)))
```

```
Error in solve.default(X %*% t(X)) :
  Lapack routine dgesv: system is exactly singular: U[2,2] = 0
```

```r
try(qr.solve(qr(X),y))
```

```
Error in qr.solve(qr(X), y) : singular matrix 'a' in solve
```

```r
lm.fit(X, y)$coefficients
```

```
x1 x2
 1 NA
```

Q: how would we do this with SVD (`svd`), or Cholesky decomposition (`chol`)?

side note: Bessel's correction

- ESL gives $\hat{\sigma}^2 = \frac{1}{N-p-1} \cdot \text{RSS}$
  - **note** p doesn't include the constant term/intercept column
- note unbiased estimate of the residual variance
- MLE would give RSS/N
- unbiased estimate of resid std. error divides by $N - 1.5$; minimum MSE (for Normal distribution) divides by $N+1$ (!)
- bias is scale-dependent ($E(f(x)) \neq f(E(x))$ in general) and might not matter as much as you think

prostate cancer example

- data exploration: `pairs(., gap = 0)` (can be extended with `panel` function); `corrplot::corrplot.mixed(., lower="number", upper = "ellipse")`; `GGally::ggpairs()`. Can use `faraway::prostate`.

```r
## a bit of data exploration
pp <- (prostate
  |> mutate(across(
    where(~length(unique(.))<=4),
    factor))
)
ggpairs(pp)
corrplot::corrplot.mixed(cor(prostate),
              lower ='number', upper = 'ellipse')
```

train/test error

- hardly worth it for simple regression problems (measures like adjusted R^2 and AIC(c) give reasonable estimates of out-of-sample error)

Gauss-Markov theorem

- simple
- applicable as long as data are independent and homoscedastic (iid is stronger)
- MVUE (minimum-variance *unbiased* estimator)
- but **not** necessarily minimum MSE!

regression by orthogonalization (3.2.3)

- build up regression by successive orthogonalization
    - regress $\mathbf{x}_j$ on residuals of all previous columns $(\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_j)$ to get coefficients $\hat{\gamma}_{\ell j}$, residual $\mathbf{z}_j$.
    - regress $\mathbf{y}$ on $\mathbf{z}_p$ to get $\hat{\beta}_p$
    - order???
- Gram-Schmidt orthogonalization (successive projection)
- if $\mathbf{Z}$ is the residual columns and Γ is the (upper-triangular) matrix of $\gamma_{\ell j}$, then $\mathbf{X} = \mathbf{Z}\Gamma$
- if $\mathbf{D} = \text{Diag}(||\mathbf{z}_j||)$
- and $\mathbf{X} = \mathbf{Z}\mathbf{D}^{-1}\mathbf{D}\Gamma = \mathbf{Q}\mathbf{R}$ with $\mathbf{Q}$ orthonormal, $\mathbf{R}$ upper triangular
- $\rightarrow$ standard decomposition!

multiple outputs

- somewhat niche problem …
- changing $\mathbf{y}$ to $\mathbf{Y}$, β to $\mathbf{B}$, the algebra mostly stays the same
- separate coefficients for each problem
- if homoscedastic, no need to consider correlation of observations!

return to subset/stepwise selection

- still not sure it's worth it
- can update efficiently based on QR decomp
- forward-stagewise: **less** efficient
- **digression**: inefficiency as a virtue
    - improve bias-var tradeoff by *worsening* fit
    - early stopping, dropout, etc. etc.

shrinkage methods

ridge

- L2 penalty on coefficients
- predictors must be normalized! (scale of β_j depends on scale of x_j)
- equivalence between penalty ($+\lambda \sum \beta^2$) and constraint ($\sum \beta^2 \leq t$)
  ("one-to-one correspondence" between λ and t, but not simple!)
- add $\lambda \mathbf{I}$ in the normal equations
- works for non-full-rank problems

Bayesian analogue

- analogous to setting iid Gaussian prior on individual β parameters

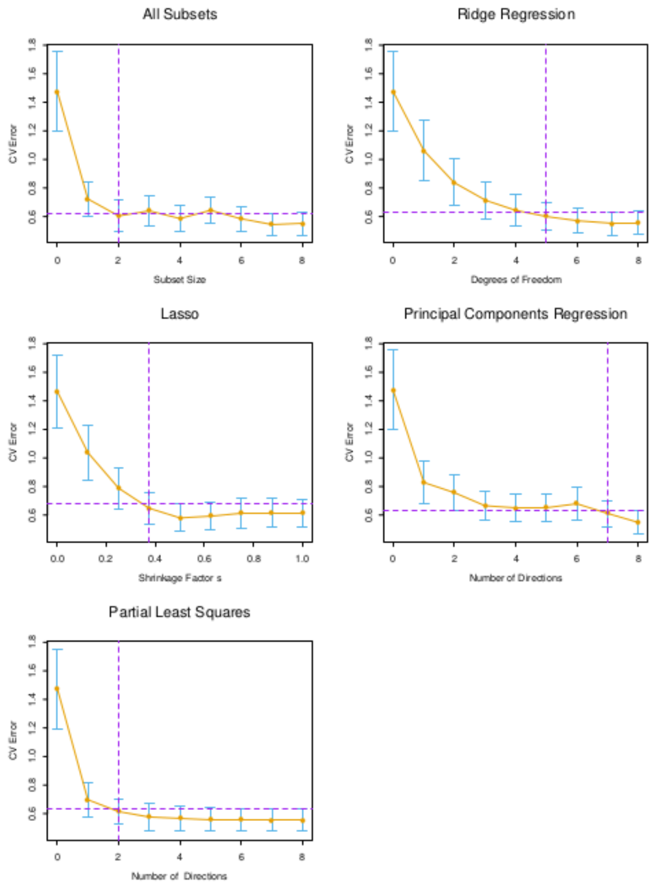

Figure 1: ESL fig 3.7

- log-posterior = log-likelihood + log-prior $\propto$ σ^2RSS + $\lambda \sum \beta^2$
- MAP (maximum *a posteriori*) estimate, **not** "proper" Bayesian est (mode, not mean, of posterior)

solving ridge by QR

- note that we can solve ridge regression by introducing *pseudo-observations* (*data augmentation*)
- set

$$\mathbf{B} = \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{pmatrix}$$

- and $\mathbf{y}^* = (\mathbf{y}\, 0)$
- so that $\mathbf{B}^\top\mathbf{B} = \mathbf{X}^\top\mathbf{X} + \lambda I$ and the residual sum of squares is unchanged
- and solving $(\mathbf{B}^\top\mathbf{B})\beta = \mathbf{B}\mathbf{y}^*$ by QR decomposition (Atlas 2013)
- ¿¿ a trick for solving for successive λ values faster ... ?

Atlas. 2013. "QR Factorization for Ridge Regression." *Mathematics Stack Exchange*. https://math.stackexchange.com/questions/299481/qr-factorization-for-ridge-regression.

singular value decomposition

- if $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ then

$$\begin{aligned}\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} &= \mathbf{U}\mathbf{D}\mathbf{V}^\top(\mathbf{V}\mathbf{D}\mathbf{U}\top \cdot \mathbf{U}\mathbf{D}\ \mathbf{V}^\top)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^\top\mathbf{y} \\ &= \mathbf{U}\mathbf{D}\mathbf{V}^\top(\mathbf{V}\mathbf{D}^2\mathbf{V}^\top)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^\top\mathbf{y} \\ &= \mathbf{U}\mathbf{U}^\top\mathbf{y}\end{aligned}$$

- and ridge translates to $\sum \mathbf{u}_j \frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^\top\mathbf{y}$
- i.e. **shrinking the j^{th} principal component** by $\frac{d_j^2}{d_j^2+\lambda}$
- (if inputs are orthonormal all coefficients are shrunk equally)

effective df

- this also shows that effective df = trace of hat matrix = $\sum \frac{d_j^2}{d_j^2+\lambda}$
- see also Hastie (2020)

Hastie, Trevor. 2020. "Ridge Regularization: An Essential Concept in Data Science." *Technometrics* 62 (4): 426–33. https://doi.org/10.1080/00401706.2020.1791959.

ridge projection

`from STAT 897D/Applied Data Mining https://online.stat.psu.edu/stat857/node/155`

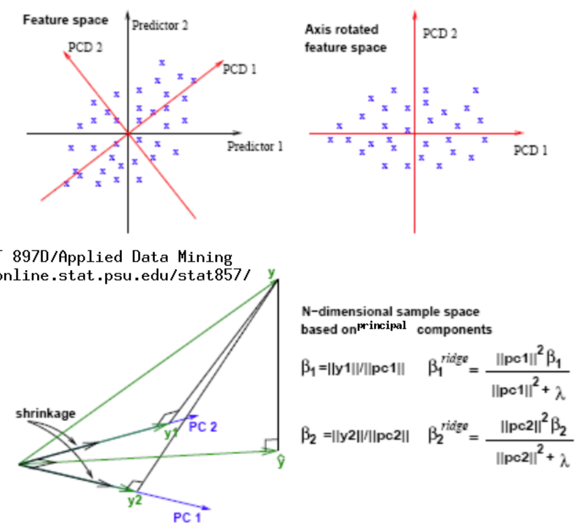
Feature space
Predictor 2
PCD 2
PCD 1
Predictor 1
Axis rotated feature space
PCD 2
PCD 1
y
N-dimensional sample space based on principal components
$\beta_1 = \|y1\|/\|pc1\|$
$\beta_1^{ridge} = \frac{\|pc1\|^2 \beta_1}{\|pc1\|^2 + \lambda}$
$\beta_2 = \|y2\|/\|pc2\|$
$\beta_2^{ridge} = \frac{\|pc2\|^2 \beta_2}{\|pc2\|^2 + \lambda}$
shrinkage
y1
PC 2
$\hat{y}$
y2
PC 1

Figure 2: The Geometric interpretation of principal components and shrinkage by ridge regression.

lasso

- L1 regularization
- **sparsity-inducing**
- least-angle regression (LARS): nice, but superseded (also, doesn't work for GLMs)
- `glmnet` et al. use cyclic/pathwise coordinate descent (Friedman, Hastie, and Tibshirani 2010) (also in Julia analogue)
    - plus "warm-start" algorithm

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1–22. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2929880/.

pathwise coordinate descent

- ESL § 3.8.6
- $\tilde{\beta}_k(\lambda)$ is **current** estimate of $\beta_k(\lambda)$. Then

$$R = \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \sum_{k \neq j} x_{ik} \tilde{\beta}_k(\lambda) - x_{ij}\beta_j \right)^2 + \lambda \sum_{k \neq j} |\tilde{\beta}_k(\lambda)| + \lambda |\beta_j|$$

- i.e. univariate lasso on j with k parameters fixed
- or lasso on *partial residual* $(y_i - \tilde{y}_i^{(j)}) = y_i - \sum_{k \neq j} \tilde{\beta}_k(\lambda)$
- **solution**:

$$\tilde{\beta}_j(\lambda) \leftarrow S\left(\sum_{i=1}^{N} x_{ij}(y_i - \tilde{y}_i^{(j)}), \lambda \right)$$

- where $S(t, \lambda) = \text{sign}(t)(|t| - \lambda)$
- can't do all λ automatically, but **warm start** algorithm works quickly
  - start with large λ such that all coefficients $\to 0$
  - reduce in small steps, using values from previous λ to initialize
- ¿how much worse does this get for other loss functions (e.g. GLMs)?

other penalties

- could use L_p penalization with $1 < p < 2$ (equivalent to a *generalized normal* or *exponential power* prior: $\propto \exp\left(|(x-\mu)/s|^p\right)$ (`gnorm` package)
- **elastic-net** (penalty $\propto \alpha \sum \beta^2 + (1-\alpha) \sum |\beta|$)
  - computationally nicer and sparsity-inducing

ridge vs lasso vs best-subset vs elastic net

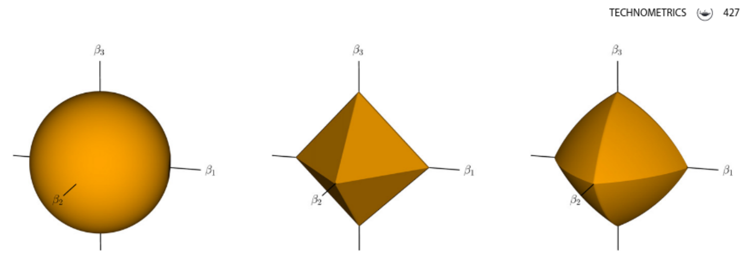

Figure 1. Constraint balls for ridge, lasso, and elastic-net regularization. The sharp edges and corners of the latter two allow for variable selection as well as shrinkage.

and more penalties

- fit unrestricted (linear regression or other) model on lasso-selected variables (why??) (Zhao, Witten, and Shojaie 2021)
- **relaxed lasso**: re-fit lasso on selected variables (why??)
- **smoothly clipped absolute deviation** (SCAD): $\lambda|\beta| \to J_\alpha(\beta, \lambda)$, with

$$\frac{dJ_a(\beta,\lambda)}{d\beta} = \lambda \cdot \text{sign}(\beta)\left[I(|\beta| \leq \lambda) + \frac{(a\lambda - |\beta|)_+}{(a-1)\lambda} I(||\beta| > \lambda)\right]$$

  for $a \geq 2$
- **adaptive lasso** $\approx |\beta|^{1-\nu}$

Zhao, Sen, Daniela Witten, and Ali Shojaie. 2021. "In Defense of the Indefensible: A Very Naïve Approach to High-Dimensional Inference." *Statistical Science* 36 (4): 562–77. https://doi.org/10.1214/20-STS815.

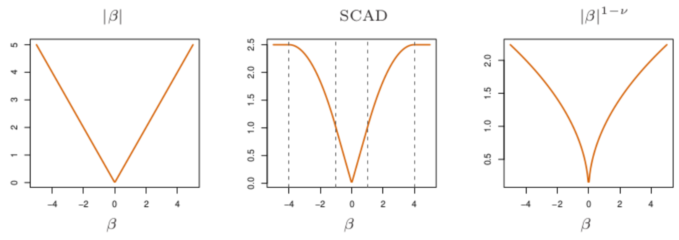

FIGURE 3.20. *The lasso and two alternative non-convex penalties designed to penalize large coefficients less. For SCAD we use $\lambda = 1$ and $a = 4$, and $\nu = \frac{1}{2}$ in the last panel.*

grouped lasso

- ESL § 3.8.4; Yuan and Lin (2006)

Yuan, Ming, and Yi Lin. 2006. "Model Selection and Estimation in Regression with Grouped Variables." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68 (1): 49–67. https://doi.org/10.1111/j.1467-9868.2005.00532.x.

- lasso on groups of parameters: compute $||\beta_\ell||_2$ by group (β_ℓ is the sub-vector of parameters in group ℓ, of length p_ℓ)
- RSS criterion plus penalty

$$\lambda \sum_{\ell=1}^{L} \sqrt{p_\ell} ||\beta_\ell||_2$$

* reduces to lasso if every parameter is in a separate group ($||c||_2 = |c|$ if c is a scalar) * ESL: “encourages sparsity at both the group and individual levels” * ¿ridge-like within groups, lasso-like between groups¿

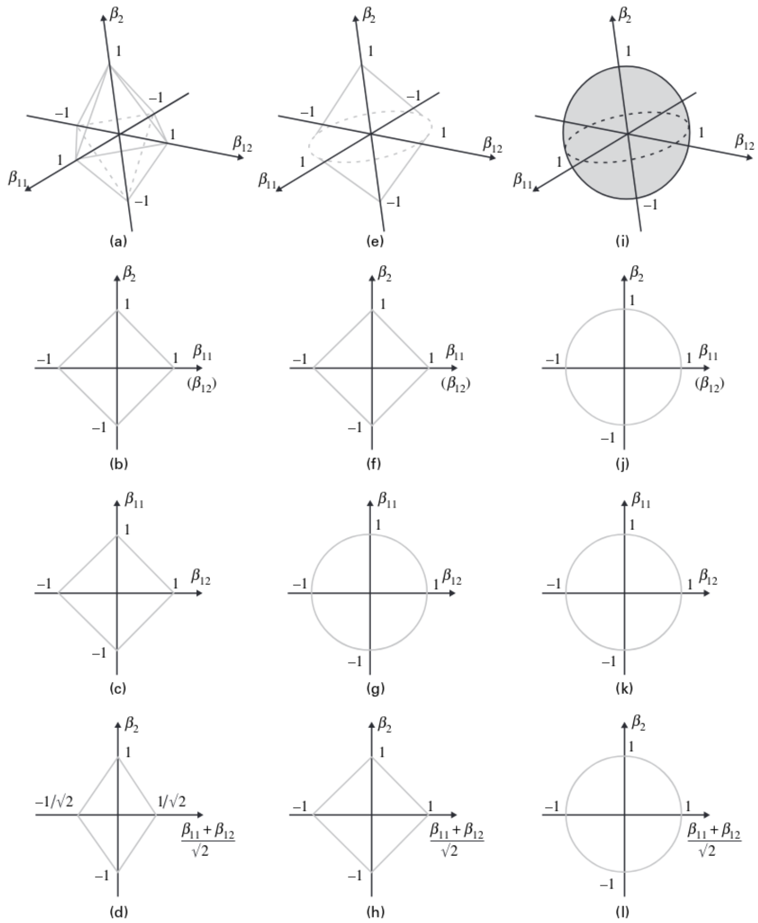

Fig. 1. (a)–(d) l_1-penalty, (e)–(h) group lasso penalty and (i)–(l) l_2-penalty

- ¿has someone written a formula-to-groupedlasso interface¿
- **sparse grouped lasso**: like elastic net (convex combination) but for regular lasso + grouped lasso

finding packages

```r
a1 <- available.packages()
grep("lasso", rownames(a1), ignore.case = TRUE, value = TRUE)
```

```
 [1] "abglasso"           "ALassoSurvIC"       "BayesianGLasso"    
 [4] "biglasso"           "bolasso"            "BTdecayLasso"      
 [7] "BTLLasso"           "CARlasso"           "CDLasso"           
[10] "cglasso"            "clogitLasso"        "covglasso"         
[13] "CVglasso"           "DIFlasso"           "DLASSO"            
[16] "DWLasso"            "elasso"             "extlasso"          
[19] "gamlss.lasso"       "genlasso"           "gglasso"           
[22] "glamlasso"          "glasso"             "glassoFast"        
[25] "glmmLasso"          "GPCMlasso"          "grplasso"          
[28] "grplassocat"        "hglasso"            "higlasso"          
[31] "ipflasso"           "islasso"            "LassoBacktracking"
[34] "LassoGEE"           "LassoNet"           "lassopv"           
[37] "lassoshooting"      "LassoSIR"           "lglasso"           
[40] "mglasso"            "MSGLasso"           "MWLasso"           
[43] "nnlasso"            "PabonLasso"         "PACLasso"          
[46] "palasso"            "pcLasso"            "PCLassoReg"        
[49] "ppmlasso"           "prioritylasso"      "sealasso"          
[52] "sglasso"            "slasso"             "smoothedLasso"     
[55] "SSLASSO"            "SummaryLasso"       "Tlasso"            
[58] "vennLasso"          "VSOLassoBag"       
```

- also see `sos` package

arm-waving

- optimization: scaling/robustness vs speed
- how do we decide on a ‘best’ model?

- run everything and compare on a test set? (Do we need another level of nested cross-validation?)
- appropriate metrics: fit quality? fit quality/time or within a time threshold?
- interpretability?
- analogue of no free lunch theorem: "any two optimization algorithms are equivalent when their performance is averaged across all possible problems" (Wolpert and Macready 1997; Giraud-Carrier and Provost 2005)

Wolpert, D. H., and W. G. Macready. 1997. "No Free Lunch Theorems for Optimization." *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82. https://doi.org/10.1109/4235.585893.

Giraud-Carrier, Christophe, and Foster Provost. 2005. "Toward a Justification of Meta-Learning: Is the No Free Lunch Theorem a Show-Stopper?" *Proceedings of the ICML-2005 Workshop on Meta-Learning*, January.