High Dimensional Statistics - Summary

General framework: (High dimensional data) $Z_1, \dots, Z_n$ i.i.d. $p = \dim(Z_i) \gg n$

examples: regression: $Z_i = (X_i, Y_i) \in \mathbb{R}^p \times \mathbb{R}$
classification: $Z_i = (X_i, Y_i) \in \mathbb{R}^p \times \{0,1\}$
graphical modeling: $Z_i = (X_i) \in \mathbb{R}^p$.

→ We have more $\underbrace{\text{parameters}}_{=:p}$ than $\underbrace{\text{sample size}}_{=:n}$ $(p \gg n)$

Asymptotic viewpoint: $p = p_n$ (function of n; to not lose the fact that $p \gg n$ as $n \to \infty$).
$p_n \to \infty$ as $n \to \infty$ (e.g. $p_n = n^\alpha$ for $\alpha > 1$)

→ We can write $p = p_n$ as abbreviation.

Landau-O-notation:
$p_n = O(a_n) \iff \frac{p_n}{a_n} \leq M < \infty \quad \forall n$
$p_n = o(a_n) \iff \frac{p}{a_n} = \frac{p_n}{a_n} \to 0$ (as $n \to \infty$)
$p_n \asymp a_n \iff p_n = O(a_n)$ and $p_n = o(a_n)$
$p_n \sim a_n \iff p_n / a_n \to 1$ (as $n \to \infty$)

nonparametric statistics: Also here $p \gg n$ is common
example $Y_i = f(X_i) + \varepsilon_i$, $X_i \in \mathbb{R}^1$, $Y_i \in \mathbb{R}$, $f(\cdot)$ smooth, $E[\varepsilon_i] = 0$.
Here $f(\cdot)$ represents an infinite dimensional parameter. Because given basis functions $\phi_i(x)$: $f(x) = \sum_{i=1}^{\infty} b_i \phi_i(x)$ → infinitely many parameters b_i.
→ because $f(\cdot)$ is smooth (typically required), it is possible to get a good estimator $\hat{f}(\cdot)$ for $f(\cdot)$.
example: smoothing splines uses $p \asymp n$, which need to be estimated → estimator has p parameters.

Sparsity Assumption: New view on how to deal with high dim. objects.
Take p parameters, but many of them are zero. (approx. zero)
This assumption also guarantees good estimators for $p \gg n$.

High-dimensional linear model: Prime example for a model to use.
$(Y_i = \sum_{j=1}^p \beta_j^0 X_i^{(j)} + \varepsilon_i)$ $Y = X\beta^0 + \varepsilon$, $Y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, $\beta^0 \in \mathbb{R}^p$, $\varepsilon \in \mathbb{R}^n$, $E[\varepsilon] = 0$
β^0 is the true parameter. X is seen as a fixed design matrix.
Sparsity: $S^0 = \text{supp}(\beta^0) = \{j \mid \beta_j^0 \neq 0\}$ → non-zero entries of parameter
the sparsity index s_0 should fulfill $s_0 = |S^0| \ll n$
→ in most applications assuming sparsity is reasonable.

We typically use centered and scaled variables, i.e.
$Y_i \leftarrow Y_i - \bar{Y}$ and $X_i^{(j)} \leftarrow \frac{X_i^{(j)} - \bar{X}^{(j)}}{\hat{\sigma}_j}$
with $\hat{\sigma}_j = \frac{1}{n} \sum_{i=1}^n (X_i^{(j)} - \bar{X}^{(j)})^2$

Lasso Estimator: Most popular estimator for high dim. lin. models.

Since $p>n$, $\text{rank}(X)<p$, so X is not of full rank.
→ ordinary least squares is not unique. It overfits and produces a residual sum of squares $=0$.
This happens by projecting Y onto the span $(X^{(1)},\dots,X^{(p)})$ of the columns of X, to receive $\hat{Y}_{OLS}$
→ Thus complexity needs to be regularized.

Lasso:
$$\hat{\beta}(\lambda) = \arg\min_{\beta} \left(\|y - X\beta\|_2^2/n + \lambda \cdot \|\beta\|_1 \right)$$

$\lambda \geq 0$ is the penalty parameter. (where $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$)

Note: this different from so-called ridge regression (Tikhonov regularization)
$$\hat{\beta}_{ridge}(\lambda) = \arg\min_{\beta} \left(\|y - X\beta\|_2^2 + \underbrace{\lambda}_{\geq 0} \cdot \|\beta\|_2^2 \right)$$

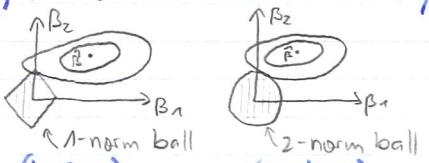
Properties of the Lasso:

(1) Lasso is a sparse estimator: $\hat{\beta}_j(\lambda) = 0$ for "many" j's. (depending on λ)
(For $\lambda \to 0$, only $\mathcal{O}(n)$ parameters will be non-zero, whereas if $\lambda = 0$, then many parameters will be non-zero → OLS)
→ thus "Lasso is doing variable selection".
→ LASSO = Least Absolute Shrinkage and Selection Operator.

(2) Lasso involves convex optimization. But for $p>n$ it is not strictly convex. Hence the Lasso solution is not unique. → but it is still a global minimum
But every local minimum is a global minimum.

(3) An equivalent version of the Lasso (1-to-1 correspondence $\lambda \sim R$) is the primal problem:
$$\hat{\beta}_{primal}(R) = \arg\min_{\beta : \|\beta\|_1 \leq R} \underbrace{\|y - X\beta\|_2^2/n}_{(*)}$$
→ not explicit, depends on data $(y_i, x_i)_{i=1,\dots,n}$

(4) The $l1$-norm is necessary for sparsity.



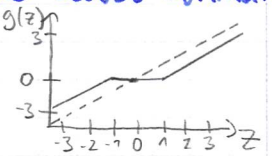
↑ 1-norm ball (Lasso) ↑ 2-norm ball (ridge)
→ R is minimal at non-sparse point (almost always)

→ 2-dim. example. The contour lines describe the values of $(*)$. The edges in the 1-norm ball enable the contour line (lowest R-value) to enter the norm-ball (at radius R) in a sparse point. (→ most of the time) This is not possible for l_q-norms with $q>1$ ($q<1$ is not a norm, but would work

Orthonormal design (of X): X is of orthonormal design, if $X^TX/n = I_p$
Note, this implies that $p \leq n$. We have orthonormal columns of X.
Then the Lasso has an explicit solution:
$$\hat{\beta}_j(\lambda) = g_{\lambda/2}(z_j) \quad \text{with} \quad z_j = (X^Ty)_j/n \; (= \hat{\beta}_{OLS,j})$$
→ since $\beta_{OLS} = (X^TX)^{-1}X^Ty = \left(\frac{X^TX}{n}\right)^{-1} X^Ty/n = X^Ty/n$

and $g_\lambda(z) = \text{sign}(z) \cdot (|z| - \lambda)_+ = \text{sign}(z) \cdot \max\{0, |z| - \lambda\}$

→ this corresponds to a sparsification of the OLS estimator, by truncating values of β to zero, if they are close to zero.
→ But this sparsification introduces bias into the Lasso estimator. Since values outside of the truncation area are still shifted by λ. Optimally they shouldn't be shifted at all. → "hard thresholding"



---: no modification of z (OLS)
—: Lasso

→ We can formally define this hard thresholding by (with parameter $\sqrt{\lambda}$)
(Assume again $X^TX/n = I_p$)

$$\hat{\beta}_{L_0}(\lambda) = \operatorname{argmin}_\beta \left(\|y - X\beta\|_2^2/n + \lambda \|\beta\|_0 \right)$$

(→ with L_0 penalization)

where $\|\beta\|_0 = |\{j \mid \beta_j \neq 0\}|$ (# non-zero components)
In principle very good, but it is hard to compute.
Hard-thresholding exhibits less bias than Lasso, but is still biased (because of the truncating).

Prediction with the Lasso estimator: Assume we want to predict (estimate) values of $f(x) = E[y \mid X = x] = \sum_{j=1}^p \beta_j^0 x_j = (\beta^0)^T x$
Thus, given the Lasso estimator $\hat{\beta}$, our prediction is

$$\hat{f}(x) = \hat{\beta}(\lambda)^T x \quad \text{and choose } \lambda \text{ via cross validation.}$$

Note: Sometimes the prediction y from the last layer of a NN is very high dimensional $\phi(x_i) \in \mathbb{R}^d$, and we can use Lasso to predict a y from $\phi(x_i)$. $\hat{\beta}(\lambda) = \|y - (\phi(x_1)^T, \dots, \phi(x_n)^T)^T\|_2^2/n + \|\lambda\|_1$.

Measuring prediction quality: In practice, to measure prediction quality of a Lasso method, we look at the cross-validation (CV) error.
From a theory point of view, we look at

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n \quad (\rightarrow \text{how well we estimate our regression surface } X\beta^0)$$

, or the "Prediction error": $E[\|X(\hat{\beta} - \beta^0)\|_2^2/n] = E[\|y - X\hat{\beta}\|_2^2/n] + \sigma_\varepsilon^2$

Asymptotic theory and the Lasso: Remember the linear model, now for any n:

$$y_{n,i} = \sum_{j=1}^{p_n} \beta_{n,j}^0 X_{n,i}^{(j)} + \varepsilon_{n,i}, \quad E[\varepsilon_{n,i}] = 0.$$

for $i = 1, \dots, n$ (rows), $n = 1, 2, \dots$, $X^{(j)}$ a column of X.
With growing n, the data evolves as follows

$$\begin{matrix} (X,Y)_{n;1}, \dots, (X,Y)_{n;n} \\ (X,Y)_{n+1;1}, \dots, (X,Y)_{n+1;n}, (X,Y)_{n+1;n+1} \\ (X,Y)_{n+2;1}, \dots, (X,Y)_{n+2;n}, (X,Y)_{n+2;n+1}, (X,Y)_{n+2;n+2} \end{matrix}$$

→ we call this triangular array asymptotics.

Preview

We have the following asymptotic results for the Lasso:
(1) (For fixed design X) If $\|\beta^0\|_1 = o(\sqrt{n/\log(p)})$ (→ some degree of sparseness) then $\|X(\hat{\beta} - \beta^0)\|_2^2/n = o_P(1)$
(2) (For fixed design X) If a certain "compatibility condition" with constant $\phi_0^2 > 0$ is fulfilled, then

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n = O_P\left(\frac{s_0 \log(p)}{n} \frac{1}{\phi_0^2}\right) \quad \text{and} \quad \|\hat{\beta} - \beta^0\|_1 = O_P\left(s_0 \sqrt{\frac{\log(p)}{n}} \frac{1}{\phi_0^2}\right)$$

where $s_0 = |S_0| = |\{j \mid \beta_j^0 \neq 0\}|$
→ ϕ_0^2 close to zero means highly correlated columns of X.

→ We will discuss these two results in more detail later.

Corollary 6.1: Assume $\varepsilon \sim \mathcal{N}_n(0, \sigma^2 I)$, and scaled columns (+fixed design X) (result (1) from previous page)
i.e. $\hat{\sigma}_j^2 \equiv 1 \ \forall j$ (otherwise Lasso and Ridge are nonsensical)
with $\hat{\sigma}_j = \frac{1}{n}\sum_{i=1}^n (x_i^{(j)})^2$ (note the x_i are centered) → along each row

Then for $\lambda = 4\hat{\sigma}\sqrt{\frac{t^2 + 2\log(p)}{n}}$ with $\hat{\sigma}$ an estimator of σ.

with probability $1-\alpha$ with $\alpha = 2\exp(-t^2/2) + \mathbb{P}[\hat{\sigma} < \sigma]$
we have that $\|X(\hat{\beta} - \beta^0)\|_2^2/n \leq \frac{3}{2}\lambda\|\beta^0\|_1$ → not asymptotic statement!

Implication of corollary: It turns out, the "proper" ("best") λ fulfills $\lambda \asymp \sqrt{\log(p)/n}$ (e.g. take $t^2 \asymp \log(p)$ above). → asymptotically we want $\lambda \to 0$ (because $\log(p) \ll n$)

Then by the corollary above

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n = \mathcal{O}_{\mathbb{P}}\Big(\underbrace{\lambda}_{\asymp\sqrt{\log(p)/n}}\|\beta^0\|_1\Big) = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\log(p)/n}\,\underbrace{\|\beta^0\|_1}_{\text{↳ corresponds to sparsity of } \beta^0}\Big)$$

→ This means that if $\sqrt{\frac{\log(p)}{n}}$ goes to zero, then the (sparsity of β^0 i.e.) $\|\beta^0\|_1$ can even grow.
→ For the very sparse case $\|\beta^0\|_1 = \mathcal{O}(1)$, then the total convergence rate is only $\mathcal{O}_{\mathbb{P}}(\sqrt{\log(p)/n})$, whis is very slow.
(Note: We have made no assumptions for X, e.g. uncorrelated columns) → this is exactly the problem

To understand why the convergence rate is so slow, we consider the benchmark OLS-oracle (OLS, and it knows which param. are zero)
Then $\|X(\hat{\beta}_{\text{ols-oracle}} - \beta^0)\|_2^2/n = \mathcal{O}_{\mathbb{P}}\left(\frac{s_0}{n}\right)$, $s_0 = |S^0| = |\{j \mid \beta_j^0 \neq 0\}|$
this is a much faster rate.

→ We will see that with additional assumptions on X, we can achieve a convergence rate close to the OLS-oracle: (→ see "oracle inequality" p.6)

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n = \mathcal{O}_{\mathbb{P}}\left(\log(p)\,\frac{s_0}{n}\right) \quad \text{(if } \beta_0^2 \text{ bounded away from zero)}$$

→ Main takeaway from corollary: $\lambda_n = C \cdot \sigma\sqrt{\log(p_n)/n}$

↳ For σ unknown, take $\hat{\sigma}$ s.t. $\mathbb{P}(C' > \hat{\sigma} \geq \sigma) \to 1$ (as $n \to \infty$)
Then $\|X(\hat{\beta}(\lambda_n) - \beta^0\|_2^2/n \to 0$ in probability (as $n \to \infty$)

Note: The proving technique for the corollary is based on a decoupling of the problem into a deterministic and probabilistic part. First prove the statement deterministically on a set J, and then show that the probability of this set J is high. ($\mathbb{P}(J)$). (→ Def. of J in "oracle inequality" on p.6)
The deterministic part remains the same for other probabilistic structures such as:

- heteroscedastic errors with $\mathbb{E}[\varepsilon_i] = 0$, $\mathrm{Var}(\varepsilon_i) = \sigma_i^2 \neq \text{const.}$ → $\varepsilon \sim \mathcal{N}_n(0, D)$, $D = \mathrm{diag}(\sigma_1^2, \dots, \sigma_n^2)$. Assume $\sigma_i^2 \leq \sigma^2 < \infty$ (← some positive constant). Then the above corollary is true with σ^2.
- dependent observations (fixed design, dependent errors)
- non-gaussian errors
- random design: ε independent of X. Because otherwise conditioning on X (i.e. $Y|X \to \varepsilon|X$), which is necessary for prediction, this would prevent us from assuming that design X is fixed. ↳ necessary assumption of corollary.

Compatibility Condition: For identifying and estimating the true regression parameter β^0, we need to make more assumptions. Because if e.g. the columns $X^{(1)} = X^{(2)}$, then we cannot distinguish between β_1^0 and β_2^0 (components of β_0). Then typically we assume $\text{rank}(X) = p$. But this is not possible for us, since $p > n$ (and $\text{rank}(X)$ can at most be n).

→ β_1^0, β_2^0 can be interchanged, to receive a different valid β^0.

→ To address the problem of identifiability, we want to show that $X\theta = X\beta^0 \Rightarrow \theta = \beta^0$. For this we can use eigenvalues:

$$0 = \|X(\theta - \beta^0)\|_2^2 / n = (\theta - \beta^0)^T \hat{\Sigma} (\theta - \beta^0) \geq \lambda_{\min}^2(\hat{\Sigma}) \|\theta - \beta^0\|_2^2$$

→ If $\lambda_{\min}^2 \neq 0$, then $\theta = \beta^0$ follows.

where $\lambda_{\min}^2(\hat{\Sigma})$ is the minimal eigenvalue of $\hat{\Sigma} = X^T X / n$.
The last inequality follows from definition: (↑ NOT squared eigenvalue)

$$\lambda_{\min}^2(\hat{\Sigma}) = \min_{u: u \neq 0} \frac{u^T \hat{\Sigma} u}{\|u\|_2^2}$$

But note that for $p > n$: $\lambda_{\min}^2(\hat{\Sigma}) = 0$. So the above bound is "useless."
Because $\lambda_{\min}^2(\hat{\Sigma}) > 0 \Rightarrow \|\theta - \beta^0\|_2^2 = 0 \Rightarrow \theta = \beta^0$.
In a sense, $\lambda_{\min}^2$ measures the identifiability of β^0. If $\lambda_{\min}^2$ is closer to zero, then it is less identifiable.
(↳ for higher dimensions)

Sparse eigenvalues: We can still solve the above problem for high dimensions ($p \gg n$) by restricting to small sub-matrices.
For this, we first define

$$\phi_{\min}^2(m) = \min_{\substack{S \subseteq \{1,\dots,p\} \\ |S| \leq m}} \lambda_{\min}^2(\hat{\Sigma}_S) \quad \text{or} \quad \Leftrightarrow \phi_{\min}^2(m) = \min_{\beta \neq 0;\, \|\beta\|_0 \leq m} \frac{\beta^T \hat{\Sigma} \beta}{\|\beta\|_2^2}$$

where S denotes a subset of columns of X, to construct $\hat{\Sigma}_S$.

Now assume we have any other sparse vector θ. Define $s_\theta = \|\theta\|_0$ and $s_0 = \|\beta^0\|_0$. If we then require that $\phi_{\min}^2(s_\theta + s_0) > 0$ we receive

$$0 = \|X(\theta - \beta^0)\|_2^2 / n \geq \phi_{\min}^2(s_\theta + s_0) \|\theta - \beta^0\|_2^2$$

(↳ since $\|\theta - \beta^0\|_2^2 \leq s_\theta + s_0$)

$$\Rightarrow \theta = \beta^0.$$

(↳ both θ and β^0 are sparse. + submatrices are sufficiently well conditioned.)

→ if we check identifiability among other sparse vectors ($\phi_{\min}^2$), then we are successful.
→ If we restrict to sparse vectors θ with at most sparsity of β^0, i.e. $\|\theta\|_0 = s_\theta \leq \|\beta^0\|_0 = s_0$
Then we can identify the regression parameter vector β^0, if $\phi_{\min}^2(2s_0) > 0$.
→ If $\phi_{\min}^2(m) > 0$ for $m \gg s_0$, then Lasso identifies β^0 with high probability (→ because with high probability $\|\hat{\beta}(\lambda)\|_0 \asymp \|\beta^0\|_0 = s_0$ for suitable λ)

Restricted eigenvalues: In practice, instead of using sparse eigenvalues, Lasso can identify β^0 under weaker conditions.
The idea is to add the restriction of a cone condition:
For a $S \subseteq \{1,\dots,p\}$:

$$\|\beta_{S^c}\|_1 \leq 3 \|\beta_S\|_1 \qquad (*)$$

(↳ arbitrary.)

Then define the "restricted eigenvalue":

$$\kappa^2(m, 3) = \min_{S;\, |S| \leq m} \min_{\substack{\beta \neq 0; \\ \beta \text{ fulfills } (*)}} \frac{\beta^T \hat{\Sigma} \beta}{\|\beta_S\|_2^2}$$

We can then show that for Lasso, with high probability (i.e. on $\mathcal{T}$), the cone cond. $(*)$ is fulfilled for $\hat{\beta} - \beta^0$, $S = S_0$
i.e. on $\mathcal{T}$: $\|(\hat{\beta} - \beta^0)_{S_0^c}\|_1 = \|\hat{\beta}_{S_0^c}\|_1 \leq 3 \cdot \|(\hat{\beta} - \beta^0)_{S_0}\|_1$.
→ If $\kappa^2(s_0, 3) > 0$, then β^0 is indentifiable with the Lasso.

Compatibility constant: A slightly weaker condition (than the restricted eigenvalue) for the Lasso to identify the true parameter β^0 can be stated using the so-called compatibility constant for the set S_0:

$$\phi_0^2 = \min_{\beta \neq 0} \min_{\|\beta_{S_0^c}\|_1 \leq 3\|\beta_{S_0}\|_1} \frac{s_0 \cdot \beta^T \hat{\Sigma} \beta}{\|\beta_{S_0}\|_1^2} \qquad (s_0 = |S_0| = \|\beta^0\|_0).$$

Using Cauchy-Schwarz we can show that

$$\phi_0^2 \geq \kappa^2(S_0, 3) \qquad \rightarrow \text{thus it is a weaker condition.}$$

Finally, if the **compatibility condition** $\phi_0^2 > 0$ holds, then the true parameter β^0 is identifiable using the Lasso.

Note: For restricted eigenvalues (thus also compatibility constant) it holds

$$\|\hat{\beta} - \beta^0\|_2^2 = \mathcal{O}(s_0 \cdot \log(p)/n) \xrightarrow{} 0$$
$$\hookrightarrow \text{if } s_0 = o(n/\log(p))$$

$$\|\hat{\beta} - \beta^0\|_1 = \mathcal{O}(s_0 \cdot \sqrt{\log(p)/n}) \xrightarrow{} 0$$
$$\hookrightarrow \text{if } s_0 = o(\sqrt{n/\log(p)})$$

→ The compatibility condition is the weakest assumption (among restricted and sparse eigenvalues) which still allows to achieve (near) statistical optimality of Lasso.

Oracle inequality for Lasso: Assume the compatibility condition holds with compatibility constant ϕ_0^2 ($\geq L > 0$). Then on $\mathcal{T} = \{2 \cdot \max_{j=1,\dots,p} |\varepsilon^T X^{(j)}/n| \leq \lambda_0\}$ (→ set with high probability if $\lambda_0 \asymp \sqrt{\log(p)/n}$) it holds that

→ we call λ_0 the "noise level" (choose λ_0 appropriately)

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n + \lambda \cdot \|\hat{\beta} - \beta^0\|_1 \leq 4\lambda^2 s_0/\phi_0^2 \qquad \text{for } \lambda \geq 2\lambda_0$$

→ corollary (6.8): Assume the row vectors of X are i.i.d. sampled from a sub-gaussian distribution with mean zero, and cov-matrix Σ. If $\lambda_{\min}^2(\Sigma) > 0$ and $s_0 = |S^0| = \mathcal{O}(\sqrt{n/\log(p)})$. Then for some $C > 0$:

$$\phi_0^2 \geq C \lambda_{\min}^2 > 0 \xrightarrow{n \to \infty} 1 \qquad \text{with high probability.}$$

Results for convergence of prediction/Lasso:

→ If the compatibility condition holds with $\phi_0^2 \geq L > 0$, and $\lambda (= 2\lambda_0) \asymp \sqrt{\log(p)/n}$ (as $p \geq n \to \infty$), then we receive a fast rate of convergence:

$$\|X(\hat{\beta} - \beta^0)\|_2^2/n = \mathcal{O}_P(s_0 \cdot \log(p)/n)$$ → minimax optimal rates. No other method can do better.

- If S^0 with $s_0 = o(n)$ was known (oracle), then $\|X(\hat{\beta}_{OLS} - \beta^0)\|_2^2/n = \mathcal{O}_P(\frac{s_0}{n})$. Thus $\log(p)$ is the (small) price for not knowing S_0 (→ set)
- Further, the estimation error for β^0 in terms of L_1-norm fulfills: $\|\hat{\beta} - \beta^0\|_1 = \mathcal{O}_P(s_0\sqrt{\log(p)/n})$ → minimax optimal rates. No other method can do better.

→ If instead we assume the restricted eigenvalue condition with $\kappa^2(3, s_0) \geq L > 0$. Then we receive a "better" result for the estimation error for β^0 in terms of the L_2-norm:

$$\|\hat{\beta} - \beta^0\|_2 = \mathcal{O}_P(\sqrt{s_0 \log(p)/n})$$

Also: $\|\hat{\beta} - \beta^0\|_2 \xrightarrow{P} 0$ if $s_0 = o(n/\log(p))$
$\|\hat{\beta} - \beta^0\|_1 \xrightarrow{P} 0$ if $s_0 = o(\sqrt{n/\log(p)})$ } consistency

Variable screening: Remember the active set $S_0 = \{j \mid \beta_j^0 \neq 0\}$.
Generally the probability that the estimated active set $\hat{S}_0 = S_0$ is very low. So instead we can show the following:

If $\min_{j \in S_0} |\beta_j^0| > 4\lambda s_0 / \phi_0^2$ "beta-min condition" → if the non-zero coefficients are sufficiently large

then $\mathbb{P}(S_0 \subseteq \hat{S}_0) \geq \mathbb{P}(T)$ (large probability).

Thus with high probability, the Lasso selects a superset of the active set S_0 (→ doesn't miss an active variable). } in theory!

→ In theory if we assume (1) compatibility condition for fixed design X, (2) beta-min condition, (3) i.i.d. Gaussian errors, then

$$\mathbb{P}(S_0 \subseteq \hat{S}_0) \to 1$$

→ Further, we receive that for Lasso $|\hat{S}| \leq \min(n, p)$. Hence in the case of $p \gg n$, we achieve a huge dimensionality reduction. → typically less

→ In practice $\mathbb{P}(S_0 \subseteq \hat{S}_0)$ might not be very large. Even if we choose λ very small (which would make $\hat{s}_0$ larger).
Possible reasons for this are (1) that the compatibility constant ϕ_0^2 is very small (eg. higly correlated columns of X), or (2) that the errors are strongly non-gaussian (heavy tailed) and thus need large λ to have reasonable probability of T.
Both of these cases would require a strong beta-min-condition, to be solved.
↳ It is "empirically evident" though, that all coefficients with large values are captured (with high probability). → small-valued coefficients are difficult to capture

→ Under much more restrictive conditions on the design X (irrepresentable condition or neighborhood stability condition), and assuming the beta-min condition $\min_{j \in S_0} |\beta_j^0| \gg \sqrt{s_0 \log(p)/n}$, then $\mathbb{P}(\hat{S}_0 = S_0) \to 1$ as $n \to \infty$.
It turns out the irrepresentable condition is sufficient and essentially necessary for consistent variable selection ($\|\hat{\beta} - \beta^0\| \to 0$), but it is in practice often not fulfilled.

→ Thus variable selection with Lasso is unrealistic. Variable screening is realistic though (screen variables with high values). So Lasso should instead mean: LASSO = Least Absolute Shrinkage and Screening Operator → Tibshirani agrees.

Recap overview table:

property	condition on design X	size of non-zero coefficients
slow prediction convergence rate	/	/
fast prediction convergence rate	compatibility cond.	/
estimation error bound ($\|\hat{\beta} - \beta^0\|_1$)	compatibility condition	/
variable screening	compatibility condition or restricted eigenvalue	beta-min- condition or weaker version
variable selection	neighborhood stability $\Leftrightarrow$ irrepresentable condition	beta-min condition

Lasso-OLS hybrid: This describes a method which is computable for general X, and with lower bias than Lasso.
(1) First, run Lasso, to receive $\hat{S}_0$ (note: $|\hat{S}_0| \leq \min(n,p)$)
(2) Run OLS only on $\hat{S}_0$, to receive $\hat{\beta}^{\hat{S}_0}_{OLS}$.
This method works well, if $S_0 \subseteq \hat{S}_0$.

Adaptive Lasso: This method was derised to address the bias problem of the Lasso. It works as follows:

(1) Get initial estimator $\hat{\beta}_{init}$ (with eg. the Lasso)
(2) Compute Lasso, but with re-weighted l_1-penalty:

$$\hat{\beta}_{adapt}(\lambda) = \operatorname{argmin}_\beta \left(\|y - X\beta\|_2^2/n + \lambda \sum_{j=1}^p \frac{|\beta_j|}{|\hat{\beta}_{init,j}|} \right).$$

Note that if $\hat{\beta}_{init,j}(\lambda)=0$, then also $\hat{\beta}_{adapt,j}(\lambda)=0$, because we have infinite penalty on $|\beta_j|$.

→ Usually the steps are run only twice. After each run-through, the model becomes more sparse
→ The adaptive Lasso often works well in practice, since it is more sparse than Lasso. If the truth is assumed to be sparse, then it has better variable screening (and selection) properties than Lasso (in theory).
→ Generally, the adaptive Lasso should be prefered to the Lasso.

Lasso regularization path: We can compute the values of the components of $\hat{\beta}(\lambda)$ for all λ. (Specifically we plot each coefficient against $\|\hat{\beta}(\lambda)\|_1 / \max_\lambda \|\hat{\beta}(\lambda)\|_1$, where larger values correspond to smaller λ-values.) (components)
→ We notice that in general $\hat{\beta}_j(\lambda)$ <u>are not monotone in the non-zeros</u>.
So for example: $\hat{\beta}_j(\lambda) \neq 0$ and $\hat{\beta}_j(\lambda') = 0$ for $\lambda' < \lambda$.
→ correlation of columns causes non-monotonicity.

Generalized linear model (GLM): $Y_1, \dots, Y_n \in \mathbb{R}$ i.i.d. response variables with some distribution, $X \in \mathcal{X} \subseteq \mathbb{R}^p$ covariates. The GLM is

$$g(E[Y_i \mid X_i = x]) = \mu + \sum_{j=1}^p \beta_j x^{(j)} \quad (= f(x) = f_{\mu,\beta}(x))$$

where $g(\cdot)$ is some known, real-valued "link function". μ is the intercept term. It is important, and cannot be ignored by eg. centering the respons
→ The Lasso is defined as l_1-norm penalized negative-log likelihood (where μ is not penalized).
→ example: logistic (penalized) regression. $Y \in \{0,1\}$, $\pi(x) = E[Y \mid X=x] = P(Y=1 \mid X=x$
logistic function: $g(\pi) = \log(\pi/(1-\pi))$ for $\pi \in (0,1)$.
further denote $\pi_i = P(Y_i = 1 \mid X_i)$.
Then $\log(\pi_i/(1-\pi_i)) = \mu + X_i^T\beta \Rightarrow \pi_i = \frac{\exp(\mu + X_i^T\beta)}{1+\exp(\mu+X_i^T\beta)}$.
Lasso: log-likelihood $l(\mu,\beta) = \sum_{i=1}^n \log(\pi_i^{y_i}(1-\pi_i)^{1-y_i}) = \sum_{i=1}^n (y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i))$

$$= \sum_{i=1}^n \Big(y_i \underbrace{\log(\pi_i/(1-\pi_i))}_{\mu + X_i^T\beta} + \underbrace{\log(1-\pi_i)}_{\log(1+\exp(\mu+X_i^T\beta))} \Big)$$

the negative log-likelihood $-l(\mu,\beta)$ is convex in μ and β. Then the Lasso for linear logistic regression is:

$$\hat{\mu}, \hat{\beta} = \operatorname{argmin}_{\mu,\beta} \left(-l(\mu,\beta) + \lambda \|\beta\|_1 \right)$$

→ μ is not penalized.

classification with deep neural networks: Often used nowadays method for classification with DNN is

$$\log(\pi_i/(1-\pi_i)) = \mu + \underbrace{x^T\beta^{(1)}}_{\text{from NN with linear connection}} + \underbrace{w_\theta(x)^T\beta^{(2)}}_{\text{features from a last NN layer.}}$$

($w_\theta(x)$: high-dimensional)

estimator:

$$\hat{\mu}, \hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\theta} = \operatorname{argmin} -L(\mu, \beta^{(1)}, \beta^{(2)}, \theta) + \lambda(\|\beta^{(1)}\|_1 + \|\beta^{(2)}\|_1)$$

But this is a highly non-convex function in θ. Given w_θ (i.e. a trained NN), then we can use logistic Lasso.

Group Lasso: Let $G_1, \dots, G_q$ be a partition of $\{1, \dots, p\}$. We call them groups. Write the parameter: $\beta = (\beta_{G_1}, \dots, \beta_{G_q})^T$.

The goal is to find an estimator, which is group sparse, i.e.

For all $j = 1, \dots, q$ $\quad \hat{\beta}_{G_j} \equiv 0 \quad$ OR $\quad (\hat{\beta}_{G_j})_r \neq 0 \ \forall r \in G_j$. → either entirely zero, or entirely non-zero.

This is achieved by the group Lasso:

$$\hat{\beta}(\lambda) = \operatorname{argmin}_\beta (\|y - X\beta\|_2^2/n + \lambda \sum_{j=1}^q m_j \|\beta_{G_j}\|_2)$$

where typically $m_j = \sqrt{|G_j|}$.

Group sparsity is fulfilled because the objective function is non-differentiable at $\|\beta_{G_j}\|_2 = 0 \Leftrightarrow \beta_{G_j} \equiv 0 \ (j = 1, \dots, q)$

Sparse Group Lasso: allow for sparsity within groups if $\alpha > 0$ → helpful if groups are large.

$$\hat{\beta}(\lambda, \alpha) = \operatorname{argmin}_\beta (\|y - X\beta\|_2^2/n + (1-\alpha)\lambda \sum_{j=1}^q m_j \|\beta_{G_j}\|_2 + \alpha\lambda\|\beta\|_1)$$

This is a mix between Lasso and Group Lasso.

Generalized Group Lasso penalty: Generally it is helpful to be invariant to reparametrization of the design X. For this we have the generalized Group Lasso penalty:

$$\text{pen}(\beta) = \lambda \sum_{j=1}^q m_j \underbrace{\sqrt{\beta_{G_j}^T A_j \beta_{G_j}}}_{\text{for of the penalty}} \quad \text{for } A_j \text{ positive semi definite.}$$

Then we can find the estimator from the above penalty, using the standard group Lasso with the transformation:

$$\tilde{\beta}_{G_j} = A_j^{1/2} \cdot \beta_{G_j} \rightarrow \text{pen}(\tilde{\beta}) = \lambda \sum_{j=1}^q m_j \|\tilde{\beta}_{G_j}\|_2$$

and

$$X\beta = \sum_{j=1}^q \tilde{X}_{G_j} \tilde{\beta}_{G_j} =: \tilde{X}\tilde{\beta} \quad \text{with} \quad \tilde{X}_{G_j} = X_{G_j} A_j^{\tilde{-1/2}}$$

Then we can simply solve the $\tilde{\beta}$ problem: $\hat{\beta}_{G_j} = A_j^{-1/2} \cdot \hat{\tilde{\beta}}_{G_j}$

Groupwise prediction penalty: To achieve invariance we can use the penalty

$$\text{pen}(\beta) = \lambda \sum_{j=1}^q m_j \|X_{G_j}\beta_{G_j}\|_2 = \lambda \sum_{j=1}^q m_j \sqrt{\beta_{G_j}^T \underbrace{X_{G_j}^T X_{G_j}}_{=A_j \text{ from above}} \beta_{G_j}}$$

where typically $X_{G_j}^T X_{G_j}$ is positive definite for $|G_j| < n$.

→ this penalty is invariant under arbitrary reparameterizations within every group G_j. (! important!)
→ When using an orthogonal parameterization (design), i.e. $X_{G_j}^T X_{G_j} = I$. Then we are left with the standard group Lasso.
→ With categorical variables we can use groupwise orthogonalized design, or groupwise prediction penalty.

Additive model: is an extension of the linear model:

$$Y_i = \mu + \sum_{j=1}^{p} f_j(X_i^{(j)}) + \varepsilon_i$$

with f_j smooth, $E[\varepsilon_i]=0$, $\varepsilon_i \perp X_i$ (if X_i random). → f^0 is true fkt.
Typically we assume so-called identification: $\sum_{i=1}^{n} f_j^0(X_i^{(j)}) \equiv 0 \ \forall j$

For $p<n$ we get the classical additive model, which escapes the curse of dimensionality, i.e. writing $Y_i := f^0(X_i^{(1)}, \dots, X_i^{(p)}) + \varepsilon_i$.
(→in high dimensions it is extremely difficult to fit an arbitrary smooth function.)
(↳This becomes much easier to solve when we break it up into sums like above.)

In the case of $p \gg n$ it is possible to solve additive models, but we need to add sparsity among $\{f_j^0(\cdot);\ j=1,\dots,p\}$ (→i.e. many zero functions).
The additivity avoids the curse of dimensionality here aswell.
Approach: use basis expansion $h_{j,k}(\cdot)$ (eg. splines, Fourier, wavelets, ...)

$$\to f_j(\cdot) = \sum_{k=1}^{K} \underbrace{\beta_{j,k}}_{\text{unknown parameters}} \cdot h_{j,k}(\cdot) \approx f_j^0(\cdot)$$

(↳ "=" for $K \to \infty$)

Denote $(H_j)_{i,k} = h_{j,k}(X_i^{(j)})$, $\beta_j = (\beta_{j,1}, \dots, \beta_{j,K})^T$, $\beta = (\beta_1, \dots, \beta_p)^T \in \mathbb{R}^{pK}$ (large vector), $pK \times 1$

Then $\sum_{j=1}^{p} \underbrace{f_j(X_i^{(j)})}_{\approx f_j^0(X_i^{(j)})} = \sum_{j=1}^{p} H_j \beta_j \ (= H\beta)$ with $H = [H_1 \dots H_p]$

For estimation we then want that either $X^{(j)}$ is selected (i.e. $\hat\beta_j \neq 0$ with $\hat\beta_{j,k} \neq 0 \ \forall k$) or not selected (i.e. $\hat\beta_j \equiv 0$).
Using group Lasso we propose the sparse additive model (SpAM):

$$\hat\beta = \operatorname*{argmin}_{\beta} \| Y - \sum_{j=1}^{p} H_j \beta_j \|_2^2 / 2 + \text{pen}_\lambda(\beta)$$

where $\text{pen}_\lambda(\beta) = \lambda \sum_{j=1}^{p} \cancel{m_j} \cdot \| H_j \beta_j \|_2 / \sqrt{n}$ (right scaling)

↳don't use m_j, because all groups have size K → group wise prediction penalty → with $|G_j| = K \ \forall j$

(Note: (why scaling is correct) $\text{pen}_\lambda(\beta) = \lambda \sum_{j=1}^{p} \|f_j\|_n$, $f_j = (f_j(X_1^{(j)}), \dots, f_j(X_n^{(j)}))$, $\|f_j\|_n^2 = f_j^T f_j / n = \|f_j\|_2^2 / n$)

→ This simple approach does not take smoothness of different $f_j^0(\cdot)$ into account (some f_j^0 are more smooth, than others, i.e. different K for different f_j^0). This is too hard to compute.

Natural cubic splines for additive model: Taking smoothness of the $f_j^0(\cdot)$ into account is convenient when using splines.
Instead of using an explicit basis expansion, use

$$(\hat f_1, \hat f_2, \dots, \hat f_p) = \operatorname*{argmin}_{f_1, \dots, f_p \in \mathcal{F}} \left\{ \| Y - \sum_{j=1}^{p} f_j \|_2^2 / n + \lambda_1 \sum_{j=1}^{p} \|f_j\|_n + \lambda_2 \sum_{j=1}^{p} I(f_j) \right\}$$

"SpS penalty" = sparsity penalty + smoothness penalty

→ $I(f_j) = \int_a^b (f_j''(x))^2 dx$ ↳describes smoothness of f_j

where $\mathcal{F} = \{ f:[a,b] \to \mathbb{R} \mid f$ twice cont. diff.able, and $\int_a^b (f''(x))^2 dx < \infty \}$
↳"Sobolev space"
→ Now instead of choosing $K_1, \dots, K_p$, we only choose λ_1, λ_2.

→ Assume $a, b \in \mathbb{R}$, $a < \min_{i,j}(X_i^{(j)})$, $b > \max_{i,j}(X_i^{(j)})$. $\mathcal{F}$ as above.
Then $\hat f_j$'s are natural cubic splines with knots at $X_i^{(j)}$ $i=1,\dots,n$.
⇒ The optimization over functions is exactly representable as a parametric problem with dim. $\approx 3np$ (namely cubic splines).
Thus, since $\|f_j\|_n = \sqrt{\beta_j^T H_j^T H_j \beta_j / n}$, $I(f_j) = \sqrt{\beta_j^T W_j \beta_j}$, $W_j = (H_j'')^T H_j''$ ↳second derivative
we have a convex problem:

$$\hat\beta = \operatorname*{argmin}_{\beta} \left(\|Y - H\beta\|_2^2 / n + \lambda_1 \sum_{j=1}^{p} \sqrt{\beta_j^T H_j^T H_j \beta_j / n} + \lambda_2 \sum_{j=1}^{p} \sqrt{\beta_j^T W_j \beta_j} \right)$$

$$\left((W_j)_{k,l} = \int h_{j,k}''(x) h_{j,l}''(x)\, dx \right)$$

SpS Group Lasso penalty: To make computation of the above estimator easier, instead of using the SpS penalty $\lambda_1 \sum_j \|f_j\|_n + \lambda_2 \sum_j I(f_j)$ we use the alternative:

SpS Group Lasso penalty $= \lambda_1 \sum_j \sqrt{\|f_j\|_n^2 + \lambda_2 I^2(f_j)}$

in parameterized form:

$$\lambda_1 \sum_{j=1}^p \sqrt{\|H_j \beta_j\|_2^2/n + \lambda_2^2 \beta_j^T W_j \beta_j} = \lambda_1 \cdot \sum_{j=1}^p \sqrt{\beta_j^T \underbrace{(H_j^T H_j/n + \lambda_2^2 W_j)}_{A_j \text{ from generalized Group Lasso penalty}} \beta_j}$$

→notice: for any λ_2 we have a generalized group Lasso penalty.

Additive Models in high dimensions: If the problem is sparse and smooth, i.e. only a few $X^{(j)}$ influence Y (only few non-zero f_j^0) and the non-zero f_j^0 are smooth.

→Then we can often afford to model and fit additive functions in high dimensions. (→ how many variables need to be solved for)

That is, because in that case (1) dimensionality is of order $\dim = \mathcal{O}(p \cdot n)$, and $\log(\dim)/n = \mathcal{O}((\log(p) + \log(n))/n)$ (→small) and because (2) sparsity and smoothness then lead to the fact that if each f_j^0 is twice continuously differentiable

$$\|\hat{f} - f^0\|_2^2/n = \mathcal{O}_p\Big(\underbrace{\text{sparsity}}_{\#\text{non-zero } f_j} \cdot \sqrt{\log(p)} \cdot n^{-4/5}\Big)$$

Uncertainty quantification: Because of the complexity of the Lasso, it is very difficult any kind of confidence intervals or p-values. But, we cannot handle the (asymptotic) distribution of the Lasso $\hat{\beta}(\lambda)$ under the null hypotheses (H_0: β_j component is zero).

→Many sparse estimators show so-called "super efficiency", where they are better than most estimators when $\beta_j^0 = 0$, but arbitrarily bad when they are non-zero.

→ using standard bootstrapping or subsampling to get eg. a p-value has no strong theoretical nor practical support. (should not be used)

→Instead it makes more sense to de-sparsify or de-bias the Lasso.

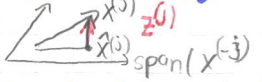
de-sparsified / de-biased Lasso: We can de-sparsify and de-bias the Lasso, by looking at partial regression of Y onto residuals Z_j. Here is a recap for $p<n$ and $\text{rank}(X)=p$:

$$\hat{\beta}_{ols,j} = y^T z^{(j)}/(X^{(j)})^T z^{(j)} = y^T z^{(j)}/\|z^{(j)}\|_2^2$$

→ project y onto the residual vector z_j and scale it.

where $z^{(j)} = X^{(j)} - X^{(-j)} \cdot \hat{\gamma}_j^{(j)}$ with $X^{(-j)} = \{X^{(k)} \mid k \neq j\}$ → columns without j.

and $\hat{\gamma}_j^{(j)} = \text{argmin}_\gamma \|X^{(j)} - X^{(-j)}\gamma\|_2^2$ →



$z^{(j)} \perp \hat{X}^{(j)}$
$z^{(j)} \perp X^{(-j)}$

→The idea for the high dimensional setting is to use the Lasso for the residuals $z^{(j)}$. Thus consider

$$z^{(j)} = X^{(j)} - X^{(-j)}\hat{\gamma}_j^{(j)} \quad \text{with } \hat{\gamma}_j^{(j)} = \text{argmin}_\gamma \|X^{(j)} - X^{(-j)}\gamma\|_2^2 + \lambda_j \|\gamma\|_1$$

Now, like before project Y onto $z^{(j)}$:

$$y^T z^{(j)}/X^{(j)} z^{(j)} \underset{Y = X\beta^0 + \varepsilon}{=} \beta_j^0 + \underbrace{\sum_{k \neq j} \frac{(X^{(k)})^T z^{(j)}}{(X^{(j)})^T z^{(j)}} \beta_k^0}_{\text{bias}} + \underbrace{\frac{\varepsilon^T z^{(j)}}{(X^{(j)})^T z^{(j)}}}_{\text{expected value zero.} \rightarrow \text{"fluctuation term"}}$$

To estimate this bias, we estimate β_k^0 using Lasso → $\hat{\beta}_k$.

Then we receive the de-sparsified / de-biased Lasso estimator

$$\hat{b}_j = \frac{y^T z^{(j)}}{(X^{(j)})^T z^{(j)}} - \sum_{k \neq j} \frac{(X^{(k)})^T z^{(j)}}{(X^{(j)})^T z^{(j)}} \hat{\beta}_k \qquad j = 1, \dots, p$$

↖from standard Lasso

→ subtract the estimated bias.

(de-sparsified / de-biased Lasso)

→ this new estimator gives us a not sparse solution (not any component will ever be exactly zero). Because at each estimation we are only looking at one component of β^0 (sparseness in total does not matter) (one 1-dim. component at a time!)

We can also write the de-biased (/ de-sparsified) Lasso:

$$\hat{b}_j = \underbrace{\hat{\beta}_j}_{\text{standard Lasso.}} + (y - X\hat{\beta})^T z^{(j)} / (x^{(j)})^T z^{(j)} \qquad \rightarrow \text{asymptotically unbiased.}$$

→ In total we run $p+1$ Lasso fits (standard Lasso + for each component)
This has high computational complexity: $\mathcal{O}(p^2 n^2)$.

→ We can now make a very helpful statement about $\hat{b}_j - \beta_j^0$.
Assume (1) $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, (2) $\lambda_j = c_j \cdot \sqrt{\log(p)/n}$ (where $|(x^{(k)})^T z^{(j)}/n| \le \lambda_j/2 \ \forall k \neq j$) and $\|z^{(j)}\|_2^2/n \ge L > 0$, (3) $s_0 = o(\sqrt{n}/\log(p))$ (a bit more sparse than "usual"), (4) $\|\hat{\beta} - \beta^0\|_1 = O_P(s_0 \sqrt{\log(p)/n})$ (i.e. $\phi_0^2 \ge L > 0$). Then

$$\underbrace{\sigma^{-1}\sqrt{n}\, \frac{(x^{(j)})^T z^{(j)}/n}{\|z^{(j)}\|_2/\sqrt{n}}}_{\text{normalizing constant.}} \cdot (\hat{b}_j - \beta_j^0) \Rightarrow \mathcal{N}(0,1) \qquad (j = 1, \dots, p)$$

($\sigma^{-1}\sqrt{n}$: σ unknown)

- Note that we don't know σ. So plug in an estimator $\hat{\sigma}^2$
  e.g. $\hat{\sigma}^2 = \|y - X\hat{\beta}\|_2^2/n$ or $\hat{\sigma}^2 = \|y - X\hat{\beta}\|_2^2/(n - \|\hat{\beta}\|_0)$ ← number of non-zero entries of $\hat{\beta}$

⇒ Using this estimator of $\hat{\sigma}$ now allows us to compute confidence intervals for the estimation of β_j^0:

$$\hat{b}_j \pm \hat{\sigma}\, \frac{1}{\sqrt{n}}\, \|z^{(j)}\|_2/\sqrt{n} \,/\, |(x^{(j)})^T z^{(j)}/n| \cdot \Phi^{-1}(1 - \tfrac{\alpha}{2})$$

→ this works quite well in practice.
→ the interval shrinks with a rate of $\frac{1}{\sqrt{n}}$ (rather fast)

Choice of tuning parameters to influence the confidence interval: (→ de-biased Lasso)

First notice that $\underbrace{\sqrt{n}\, \frac{(x^{(j)})^T z^{(j)}/n}{\|z^{(j)}\|_2/\sqrt{n}}}_{\text{→ scaling factor}} (\hat{b}_j - \beta_j^0) = W_j + \Delta_j$

where $(W_1, \dots, W_p)^T \sim \mathcal{N}_p(0, \sigma^2 \Omega)$ is jointly gaussian, and $\max_{j=1,\dots,p} |\Delta_j| = o_P(1)$ ($\xrightarrow{P} 0$).
While usually we test hypotheses $H_{0,j}: \beta_j^0 = 0$ against $H_{a,j}: \beta_j^0 \neq 0$, we can also test for group hypotheses (since it makes sense to group correlated columns into one hypothesis). Then we have
For $G \subseteq \{1, \dots, p\}$: $H_{0,G}: \beta_j^0 = 0 \ \forall j \in G$ against $H_{A,G}: \beta_j^0 \neq 0 \ \exists j \in G$.
Under $H_{0,G}$: $\max_{j \in G} \sigma^{-1}\sqrt{n}\, \frac{|(x^{(j)})^T z^{(j)}/n|}{\|z^{(j)}\|_2/\sqrt{n}} |\hat{b}_j| = \max_{j \in G} |W_j + \Delta_j| \asymp \max_{j \in G} |W_j|$ (→ bounded in probability)

Let λ_j be the Lasso parameter for each $z^{(j)}$, then we have:

$$\text{variance} = \sigma^2 n^{-1} \frac{\|z^{(j)}\|_2^2/n}{|(x^{(j)})^T z^{(j)}/n|^2} \asymp \frac{\sigma^2}{\|z^{(j)}\|_2^2}$$

then, if $\lambda_j \searrow 0$, then $\|z^{(j)}\|_2^2 \searrow 0$, and we get a large variance.
Also |bias estimation error| $\le \sqrt{n}\, \frac{\lambda_j/2}{|(x^{(j)})^T z^{(j)}/n|} \|\hat{\beta} - \beta^0\|_1 \propto \frac{\lambda_j}{\|z^{(j)}\|_2^2/n}$

then, if $\lambda_j \searrow 0$ (but not too small!), then bias estimation error $\searrow 0$ (→ smaller)

→ In total, we receive that by reducing λ_j we can decrease the error due to bias estimation (i.e. type I error), at the price of inflating the variance (i.e. slightly increasing power). Note that this also increases the length of the confidence interval.

Asymptotic efficiency (de-biased Lasso): For the de-biased Lasso to "work", we require (1) $s_0 = o(\sqrt{n}/\log(p))$ (sparsity) (which cannot be beaten in a minimax sense), (2) compatibility condition for X. If we then further require (3) $X^{(j)}$ versus $X^{(-j)}$ is sparse, i.e. $s_j \ll n/\log(p)$, then the variance of the de-biased Lasso achieves the Cramér-Rao lower bound (asymptotic optimality) i.e. $\sqrt{n}(\hat{b}_j - \beta_j^0) \Rightarrow \mathcal{N}(0, \sigma^2 \Theta_{j,j})$ → CRLB.

→ the estimator $\hat{\beta}^{(j)}$ for the 2(i) should be sp

Multiple testing adjustment: If we test all hypotheses for $j = 1, \dots, p$ $H_{0,j}: \beta_j^0 = 0$ against $H_{1,j}: \beta_j^0 \neq 0$, then we have to adjust for multiple testing. We can use different type I error measures like $FWER = P(V > 0)$ where $V = \#$ false positives (falsely rejected), OR $FDR = E[V/R]$ with $R = \#$ rejections.
Depending on the adjustment method, we can input the raw p-values p_j from each test, and then output $p_{corrected,j}$ such that reject $H_{0,j} \Leftrightarrow p_{corrected,j} \leq \alpha$.
Then it holds $FWER \leq \alpha$ or $FDR \leq \alpha$ (depending on the method of adjustment.

Stability selection: Generally it is still quite difficult to estimate discrete quantities (eg. "relevant" variables in a GLM). But there are methods which try to improve on this.
Setup: $Z_1, \dots, Z_n$ i.i.d. (eg. $Z_i = (X_i, Y_i)$), $\hat{S}_\lambda$ a "feature selection" method among eg. $\{1, \dots, p\}$ features.
We can try to assign some "relevance" to the selected features in $\hat{S}_\lambda$:
→ One such approach is subsampling: 1. I^* is a random sub-sample of $\{1, \dots, n\}$ of size $\lfloor n/2 \rfloor$, 2. compute $\hat{S}_\lambda(I^*)$, 3. repeat this B-times to receive $\hat{S}_\lambda(I^{*1}), \dots, \hat{S}_\lambda(I^{*B})$, 4. Consider their overlap.
Then we regard $\hat{\Pi}_K(\lambda) = \mathbb{P}^*(K \in \hat{S}_\lambda(I^*))$ e.g. $\hat{\Pi}_j(\lambda)$ → how often was j^{th} feature selected

→ compute $\hat{S}_\lambda$ eg. using Lasso

From this we can define the set of stable features

$$\hat{S}_{stable} = \{j \mid \max_{\lambda \in \Lambda} \hat{\Pi}_j(\lambda) \geq \pi_{thr}\}$$ for some threshold π_{thr}

which selects features without "too many" false positives (if π_{thr} chosen well)
Further define $V = |\hat{S}_{stable} \cap S_0^c|$ as the true number of false positives, $\hat{S}_\Lambda = \cup_{\lambda \in \Lambda} \hat{S}_\lambda$, $q_\Lambda = E[|\hat{S}_\Lambda(I)|]$ with I a random subsample.
→ If we assume: (1) exchangeability condition i.e. $\{\mathbb{1}_{[j \in \hat{S}_\lambda]} \mid j \in S_0^c\}$ is exchangeable $\forall \lambda \in \Lambda$, (2) $\hat{S}$ is not worse than random guessing, i.e. $E[|S_0 \cap \hat{S}_\lambda|]/E[|S_0^c \cap \hat{S}_\lambda|] \geq |S_0|/|S_0^c|$, then we can bound the expected number of false positives:

$$E[V] \leq \frac{1}{2\pi_{thr} - 1} \frac{q_\Lambda^2}{p} \quad (\pi_{thr} \in (\tfrac{1}{2}, 1))$$ (suppose we know q_Λ)

→ multivariate distribution of all stays the same after permutati → strong cond.

→ If eg. we always choose the top 10 variables, then we know $q_\Lambda \leq 10$

Specifically, we say that we tolerate some $E[V] \leq V_0$, then use $\pi_{thr} := \frac{1}{2} + q_\Lambda^2/(2pV_0)$ to choose our π_{thr}.
This formula works for any feature selection algorithm
→ Notice that as p grows, we get a better threshold of $E[V]$. Thus we can introduce additional random noise covariates (eg. i.i.d. $\mathcal{U}(0,1)$) to receive a better thresold with $p_{enlarged} > p$. But this comes at the cost of loss in power. (→ worse in detecting true positives)

→ thus the conditions are quite strong. (not always necessary; depending on method)

Data splitting procedure: To get good p-values for estimated parameters, we can do the so-called single data splitting procedure:
First, split the data into two parts I_1 and I_2 of equal size $\lfloor n/2 \rfloor$.
Then use Lasso (eg.) to select variables based on I_1: $\hat{S}(I_1)$.
Using these variables, we perform a (low-dimensional) statistical inference on I_2 base on data $(X_{I_2}^{(\hat{S}(I_1))}, y_{I_2})$. For example do inference using the t-test for single coefficients β_j^0, i.e. assign p-value 1 to $H_{0,j}: \beta_j^0 = 0$ if $j \notin \hat{S}(I_1)$.
→ this is a valid strategy due to independence of I_1 and I_2.
Note: the received p-value $P_{raw,j}$ is valid for testing H_0, if $S_0 \subseteq \hat{S}(I_1)$ ("screening property") holds.
If the screening property does not hold then $P_{raw,j}$ is still valid for $H_{0,j}(M): \beta_j^{(M)} = 0$ where $M = \hat{S}(I_1)$ is a selected sub-model, and $\beta(M) = ((X^{(M)})^T X^{(M)})^{-1} (X^{(M)})^T Y$.
→ it turns out that the received p-value depends a lot on the random split. (this can be solved by aggregating/averaging over multiple splits).

→ The issue of multiple testing is that we can at most have $|\hat{S}(I_1)|$ false positives, since $\tilde{P}_j = \begin{cases} P_{raw,j} \text{ based on } y_{I_2}, X_{I_2}^{(\hat{S}(I_1))} & \text{if } j \in \hat{S}(I_1) \\ 1 & \text{if } j \notin \hat{S}(I_1) \end{cases}$
We correct this issue with a factor of $|\hat{S}(I_1)|$ (in Bonferroni correction we use p), to receive corrected p-value:
$\tilde{P}_{corrected,j} = \min(\tilde{P}_j \cdot |\hat{S}(I_1)|, 1), \quad j = 1, \dots, p.$
(is fine when p small) → much more powerful than Bonferroni
Then using the decision rule: reject $H_{0,j} \Leftrightarrow \tilde{P}_{correction,j} \leq \alpha$ we ensure FWER $\leq \alpha$.

P-value aggregation: As mentioned above, with multiple testing, the p-value remains very unstable during computation. Thus we want to find a way to aggregate these p-values into one.
For this, let us run sample splitting B-times. We obtain the p-values $P_{corr,j}^{[1]}, \dots, P_{corr,j}^{[B]}$. To aggregate these dependent p-values, define
$Q_j(\gamma) = \min\left\{ q_\gamma\left(\{ \tilde{P}_{corr,j}^{[b]} / \gamma \mid b = 1, \dots, B \} \right), 1 \right\} \quad \gamma \in (0,1).$
($\frac{1}{2}$ is often good)
with $q_\gamma(\cdot)$ the empirical γ-quantile function. → sample γ-quantile of the p-values and divide by γ.
→ Then FWER $\leq \gamma$. (for $Q_j(\gamma)$ values)

→ Another possible aggregation method is
$P_j = \min\left\{ (1 - \log(\gamma_{min})) \cdot \inf_{\gamma \in (\gamma_{min}, 1)} \{Q_j(\gamma), 1\} \right\} \quad j = 1, \dots, p.$
Then for any $\gamma_{min} \in (0,1)$, the P_j are p-values which control the FWER.
⇒ Note that these methods always work for any p-values → indep. of high/low dim.

→ Generally multi-sample splitting has better level and power than single sample splitting.

Undirected graphical models: We call (G, P) a graphical model, for $G = (V, E)$ a graph with vertices/nodes $V = \{1, \dots, p\}$, and $E \subseteq V \times V$ a set of edges. Further, with each node we associate a random variable $X = X^{(1)}, \dots, X^{(p)}$ with multidim. distr. P.
- pairwise Markov property: ("wrt G") $(j,k) \notin E \Rightarrow X^{(j)} \perp X^{(k)} \mid X^{(V \setminus \{j,k\})}$
  i.e. two variables are independent, given all other variables.
- global Markov property: $A, B, C \subseteq V$ disjoint, then A, B are separated by $C \Rightarrow X^{(A)} \perp X^{(B)} \mid X^{(C)}$ → separated: any path from A to B passes through C.

→ global Markov property ⇒ pairwise Markov property

↓ (undirected graphical models)

→ If P has a positive and continuous density wrt. Lebesgue measure (e.g. Gaussian). Then global Markov property $\Leftrightarrow$ pairwise Markov property.

→ Note that in general, while no edge between nodes imply independence (of some sort), we cannot say that existing edges imply dependency, right away!

↳ Example: If P is Gaussian, then an edge does imply conditional dependence, i.e. $(j,k) \in E \Leftrightarrow X^{(j)} \not\perp X^{(k)} \mid X^{(V \setminus \{j,k\})}$ → but NO equivalence in global property.

Note

→ Conditional independence graph (GIC): (G,P) satisfies pairwise Markov property.
→ Gaussian Graphical Model (GGM): GIC with P Gaussian. ($P \sim \mathcal{N}_p(0,\Sigma)$)
We have for GGM: $(j,k) \in E \Leftrightarrow (\Sigma^{-1})_{j,k} \neq 0 \Leftrightarrow X^{(j)} \perp X^{(k)} \mid X^{(V \setminus \{j,k\})}$.

Nodewise regression: To estimate the edges of a graphical model (given $X = X^{(1)}, \dots, X^{(p)}$) we can use Lasso. First note that $(j = 1, \dots, p)$

$X^{(j)} = \beta_k^{(j)} X^{(k)} + \sum_{r \neq j,k} \beta_r^{(j)} X^{(r)} + \varepsilon^{(j)}$ and $X^{(k)} = \beta_j^{(k)} X^{(j)} + \sum_{r \neq k,l} \beta_r^{(k)} X^{(r)} + \varepsilon^{k}$

For a GGM it holds: $(j,k) \in E \Leftrightarrow \beta_k^{(j)} \neq 0 \Leftrightarrow \beta_j^{(k)} \neq 0$.

This leads us to the nodewise regression algorithm:
1. Run Lasso for every $X^{(j)}$ versus all others $\{X^{(k)} \mid k \neq j\}$ $(j = 1, \dots, p)$
2. Receive estimated active set $\hat{S}^{(j)} = \{r \mid \hat{\beta}_r^{(j)} \neq 0\}$ $(j = 1, \dots, p)$
3. There are two options for estimating $\hat{E}$:
   or-rule: $(j,k) \in \hat{E} \Leftrightarrow j \in \hat{S}^{(k)}$ or $k \in \hat{S}^{(j)}$
   and-rule: $(j,k) \in \hat{E} \Leftrightarrow j \in \hat{S}^{(k)}$ and $k \in \hat{S}^{(j)}$

→ Running Lasso p-times is quite fast. In total: $\mathcal{O}(np^2 \min(n,p))$.
This method has "near-optimal" statistical properties.

Graphical Lasso (GLasso): Assume our data $X_1, \dots, X_n$ i.i.d. $\sim \mathcal{N}(\mu, \Sigma)$.
Goal is to estimate $K = \Sigma^{-1}$ (precision matrix). GLasso is:

$$\hat{K}, \hat{\mu} = \operatorname{argmin}_{K \text{ p.d.}, \mu} \left(-\text{log-likelihood}(K, \mu, X_1, \dots, X_n) + \lambda \|K\|_1 \right)$$

↳ positive definite

Alternatively using the sample mean: $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n X_n$

GLasso decouples into: $\hat{K} = \operatorname{argmin}_{K \text{ p.d.}} \underbrace{\left(-\text{log-likelihood}(K, \hat{\mu}, X_1, \dots, X_n) + \lambda \|K\|_1 \right)}_{\propto -\log(\det(K)) + \operatorname{trace}(\hat{\Sigma}_{MLE} \cdot K)}$

where $\|K\|_1 = \sum_{j,k} |K_{j,k}|$ or $\sum_{j \neq k} |K_{j,k}|$

and $\hat{\Sigma}_{MLE} = \frac{1}{n} \sum_{i=1}^n (X_i - \hat{\mu})(X_i - \hat{\mu})^T$ → $\hat{K}$ is invertible, since it is pos. definite

→ thus $\hat{K}$ is usually sparse. But $\hat{K}^{-1} = \hat{\Sigma}$ is not necessarily sparse!
→ GLasso is computationally (much) slower than nodewise regression: $\mathcal{O}(np^3)$.
→ A hybrid approach: nodewise selection with estimated edges $\hat{E}$. Then GLasso restricted to $\hat{E}$ with $\lambda = 0$, i.e. unpenalized MLE restricted to $\hat{E}$.
This method is fast and accurate, and analogous to the Lasso-OLS hybrid in regression.

Nonparanormal graphical model: We look at other distributions, besides the Gaussian, where the previous conditional independence between two r.v.'s is encoded as zeros in a matrix (e.g. Σ^{-1}).

→ We say X has nonparanormal distribution if there exist functions f_j $(j=1,\dots,p)$ s.t. $Z = f(X) = (f_1(X^{(1)}),\dots,f_p(X^{(p)})) \sim \mathcal{N}_p(\mu,\Sigma)$ where w.l.o.g. $\mu=0$, $\Sigma_{jj}=1$ (by choosing f_j accordingly).
Then we have that $Z_j = f_j(X^{(j)}) \sim \mathcal{N}(0,1)$ and thus $f_j(\cdot) = \Phi^{-1}(F_j(\cdot))$ (with $F_j(u) = \mathbb{P}(X^{(j)} \le u)$) is monotone.
We call this the semiparametric Gaussian copula model.

→ If (G,P) is a nonparanormal graphical model (i.e. P nonparanormal), with f_j differentiable, $(\forall j=1,\dots,p)$. Then:

$$(j,k) \in E \iff X^{(j)} \not\perp X^{(k)} \mid X^{(V\setminus\{j,k\})} \iff \Sigma^{-1}_{j,k} \neq 0$$

(→ Covariance matrix after transformation.)

- but note that Σ here is not the covariance matrix of $X=(X^{(1)},\dots,X^{(p)})$, but of the unknown $f_1(X^{(1)}),\dots,f_p(X^{(p)})$
  → the "best" proposal to still get an estimate for Σ of X is rank based: Compute empirical rank correlation of $X^{(1)},\dots,X^{(p)}$ with bias correction from Kendall (1948). Then denote this empirical rank correlation matrix as $\hat{R}$ (this is invariant under monotone f_j's). Then put this into Glasso:

$$\hat{K} = \operatorname{argmin}_{K \text{ p.d.}} \left(-\log(\det(K)) + \operatorname{trace}(\hat{R}K) + \lambda\|K\|_1\right)$$

→ the rank-based version of Glasso has some robustness for estimating the conditional independence pattern of $X \sim P$. That is, if the distribution is nonparanormal, it still works well and properly.

Hidden confounding: All discussed methods and models still give "wrong" answers when we have hidden confounding. That is that there might be hidden (unobserved) factors which considerably influence data/outcomes in a way which we don't measure.

→ It turns out that in the population least squares principle, we have small bias if the confounding variable has dense effects. This is a blessing of high dimensionality!

→ If we have a parameter vector of the form β^0 + "bias", where β^0 is sparse, and "bias" is dense, then we should use high dimensional methods. (↳ all entries non-zero)