Penalized Likelihood in Bioinformatics

Yu Ding

MD Anderson Cancer Center

April 1, 2025

Outline

1. Motivation and Basics
2. Review of Standard Penalties
3. Advanced Penalties: SCAD and MCP
4. Oracle Property Overview
5. Applications in Genomics
6. Detailed Formulations & Examples
7. Code Snippets for Comparison
8. Group Lasso
9. Applications in Bioinformatics
10. Final Summary

High-Dimensional Bioinformatics (1/2)

Key Challenges:

- $\mathbf{p} \gg \mathbf{n}$: Many features (genes, SNPs) but few samples.
- Classic MLE is prone to overfitting or even undefined (if $p > n$).
- We need methods that reduce variance, control complexity.

Examples:

- Microarray / RNA-Seq: tens of thousands of genes, 50–200 samples.
- GWAS: up to millions of SNPs, typically a few thousand samples.

High-Dimensional Bioinformatics (2/2)

Penalized Likelihood:

- Adds a penalty term $\lambda\Omega(\theta)$ to discourage large or numerous parameters.
- Useful for **feature selection** (L1-based) or improved **stability** (L2-based).

Why it helps:

- Reduces variance by shrinking coefficients.
- Identifies a smaller subset of relevant features (genes/SNPs).
- Facilitates interpretability in biological research.

MLE vs. Penalized MLE

Maximum Likelihood Estimation (MLE)

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \ell(\theta),$$

where $\ell(\theta)$ is the log-likelihood.

Penalized MLE

$$\hat{\theta} = \arg\max_{\theta} \Big\{ \ell(\theta) - \lambda \, \Omega(\theta) \Big\}.$$

Interpretation:

- The penalty term $\Omega(\theta)$ constrains the parameter space.
- λ controls the balance between data fit and penalty.

Common Penalties (Overview)

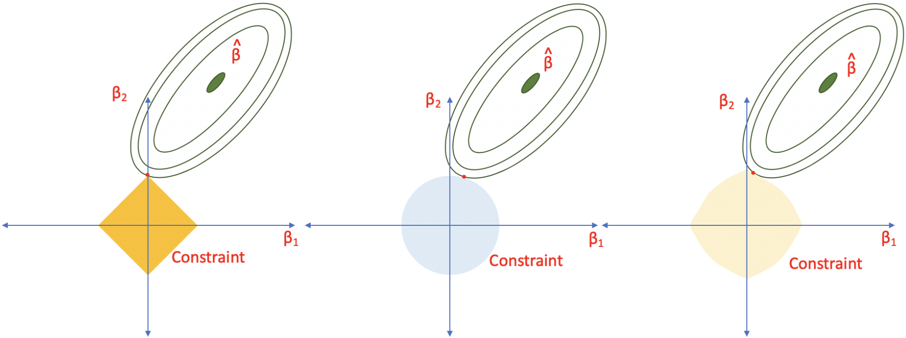
Forms of $\Omega(\theta)$:

- $\|\theta\|_2^2$ (Ridge)
- $\|\theta\|_1$ (Lasso)
- Combination: $\alpha\|\theta\|_1 + (1-\alpha)\|\theta\|_2^2$ (Elastic Net)
- Non-convex: SCAD, MCP

General Effects:

- L2 (Ridge): *continuous* shrinkage, no zero coefficients.
- L1 (Lasso): *sparsity*, some exact zeros.
- SCAD, MCP: *sparsity* + *less bias* on large coefficients.

Standard Penalties Overview



https://www.datasklr.com/extensions-of-ols-regression/regularization-and-shrinkage-ridge-lasso-and-elastic-net-regression

Ridge Regression (L2 Penalty)

Formulation

$$\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

Key Features:

- Coefficients never exactly zero.
- Great for multicollinearity.
- Typically used when we expect many variables to have moderate effects.

Bioinformatics Example:

- eQTL analysis with correlated SNPs in linkage disequilibrium.

Lasso Regression (L1 Penalty, 1/2)

Formulation

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \Big\{ \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \Big\}.$$

Effect:

- Encourages exact zeros in coefficients (feature selection).
- Tends to keep a smaller subset of variables in the model.

Pro:

- Interpretability: Only a few features remain non-zero.

Lasso Regression (2/2)

Cons:

- **Correlated predictors**: Lasso may arbitrarily pick one among them, ignoring the rest.
- **Bias on large signals**: Tends to shrink big coefficients excessively.

Bioinformatics Example:

- Differential expression classification (case vs. control) with thousands of genes.
- Lasso yields a manageable gene subset, easy to validate in the lab.

Implementation:

```
glmnet(X, y, alpha = 1)
```

Elastic Net (Combination of L1 and L2)

Penalty

$$\Omega(\beta) = \alpha\|\beta\|_1 + (1-\alpha)\|\beta\|_2^2.$$

Interpretation:

- $\alpha = 1 \Rightarrow$ Lasso.
- $\alpha = 0 \Rightarrow$ Ridge.
- $0 < \alpha < 1 \Rightarrow$ Mix of L1 & L2.

Why use it?

- Stabilizes coefficient paths in correlated scenarios more than Lasso alone.
- Still yields some sparsity, but not as aggressive as pure Lasso.

Tuning Parameter λ (CV and Criteria)

Key Role of λ:

- Large λ: heavier penalty, more shrinkage, fewer (or smaller) coefficients.
- Small λ: closer to MLE, potentially overfit if $p \gg n$.

Selection Methods:

- **k-fold Cross-Validation**: Most common. Evaluate predictive error.
- **Information Criteria**: AIC, BIC in some contexts.
- **Empirical Bayes or fully Bayesian methods** (less common in typical workflows).

Implementation Example (R):

`cv.glmnet(X, y, alpha = 1)` $\Rightarrow$ picks λ_min.

Motivation for SCAD/MCP

- Lasso's shortcoming: over-shrinkage of large coefficients.
- SCAD and MCP reduce that bias, while still achieving sparsity.
- Both are **non-convex**, requiring specialized optimization (e.g., LLA).

In Bioinformatics:

- Large signals may exist (e.g., strongly differentially expressed genes, major eQTLs).
- SCAD/MCP can better preserve these signals, potentially leading to higher accuracy.

SCAD Penalty (1/2)

Definition (Fan & Li, 2001)

$$P_\lambda^{\text{SCAD}}(\theta) = \begin{cases} \lambda|\theta|, & |\theta| \le \lambda, \\ -\frac{\theta^2 - 2a\lambda|\theta| + \lambda^2}{2(a-1)}, & \lambda < |\theta| \le a\lambda, \\ \frac{(a+1)\lambda^2}{2}, & |\theta| > a\lambda, \end{cases}$$

for $a > 2$ (often $a = 3.7$).

Pieces:

- L1-like near zero: ensures sparsity.
- Flatter penalty for large $|\theta|$: reduces bias.

SCAD Penalty (2/2)

Key Properties:

- Near-zero region: ensures small coefficients vanish.
- Large coefficients: remain relatively intact (less shrinkage).
- **Oracle property**: can identify correct zero vs. non-zero with high probability.

Implementation Example (R):

```
ncvreg(X, y, family = "gaussian", penalty = "SCAD").
```

MCP Penalty (1/2)

Minimax Concave Penalty (Zhang, 2010)

$$P_\lambda^{\mathsf{MCP}}(\theta) = \begin{cases} \lambda|\theta| - \frac{\theta^2}{2\gamma}, & |\theta| \leq \gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2, & |\theta| > \gamma\lambda, \end{cases}$$

for $\gamma > 0$.

Mechanism:

- Initially behaves like L1 near zero.
- Penalty tapers for larger $|\theta|$.

MCP Penalty (2/2)

Key Points:

- Combines sparsity with lower bias on large coefficients.
- **Oracle property** under certain assumptions.
- Need to tune both λ and γ.

Practical Use:

- `ncvreg(..., penalty="MCP")` supports linear/logistic/Cox.
- For logistic or Cox, pay attention to local minima and warm starts.

Optimization Considerations

Coordinate Descent:

- Standard for convex penalties (Ridge/Lasso/EN).
- $\mathcal{O}(p \cdot n)$ per iteration is typical.

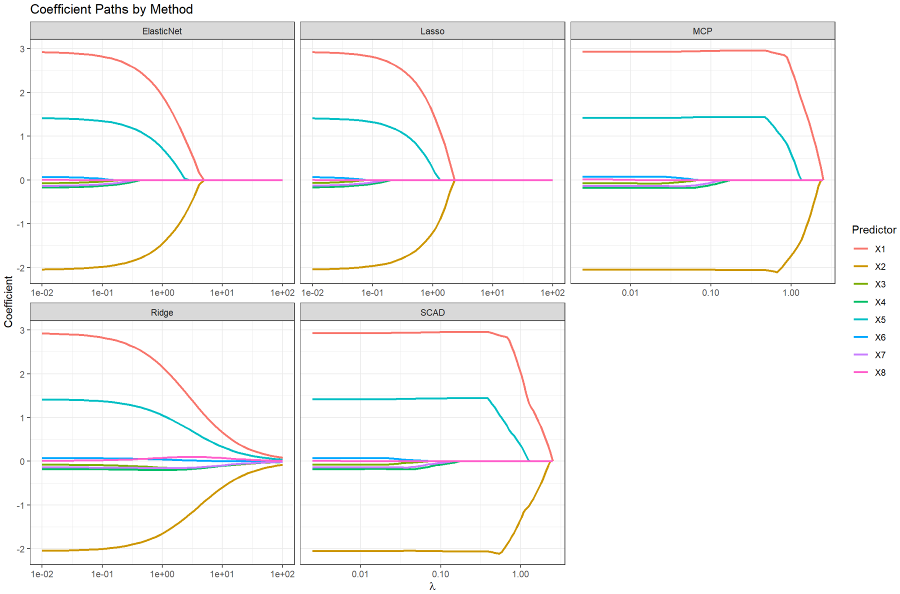
Local Linear Approx. (LLA) / Quadratic Approx. (LQA):

- SCAD/MCP: Non-convex $\Rightarrow$ approximate penalty locally, then do coordinate descent.
- *Warm starts* from Lasso solutions often help.

Pitfall:

- **Local minima** for non-convex: multiple solutions possible, so initialization matters.

Coefficient Paths by Method



Comparison of Penalties

Penalty	**Sparsity?**	**Bias on Large Coeffs?**	**Convex?**
Ridge (L2)	No	Low	Yes
Lasso (L1)	Yes	High	Yes
Elastic Net	Yes	High	Yes
SCAD	Yes	Lower	No
MCP	Yes	Lower	No

Takeaways:

- SCAD/MCP yield sparser solutions with less bias, but require non-convex optimization.
- Lasso/EN are simpler, widely available, handle large-scale data easily.

What is the Oracle Property? (1/2)

Definition:

- An estimator has **oracle property** if, asymptotically:
  1. It *correctly identifies* zero vs. non-zero coefficients.
  2. It *estimates non-zero coefficients* as well as if the true model were known beforehand.

Implication:

- No wasted effort on truly zero variables.
- Perfect or near-perfect estimation of large signals, with minimal bias.

What is the Oracle Property? (2/2)

Which Penalties Have It?

- **SCAD, MCP**: can exhibit near-oracle behavior under certain conditions (sample size, signal strength, etc.).
- **Lasso**: does *not* always have it; it can over-shrink big coefficients.

Conditions for Oracle:

- $\sqrt{n}$-consistency in coefficient estimates.
- Proper tuning of λ, a (SCAD), γ (MCP).
- Sufficient separation between zero and non-zero signals.

Genomic Data Scenarios

Typical Problems:

- **Differential expression**: thousands of genes, small sets of highly relevant ones.
- **eQTL/GWAS**: linking SNPs to phenotypes or expression, with correlation structures.
- **Survival analysis**: time-to-event data (Cox models).
- **Multi-omics**: integrate different data layers (expression, methylation, proteomics).

Common Themes:

- Overfitting is easy if not penalized.
- Biological interpretability demands smaller, more stable feature sets.

Differential Expression (1/2)

Standard Approach:

- Many separate t-tests or fold-change criteria on each gene.
- Multiple-testing corrections (FDR).

Penalized Model Approach:

- Fit a multi-gene model using Lasso/SCAD logistic regression (case vs. control).
- Sheds light on *joint* effects among genes, handles correlation.
- Potentially smaller, more robust gene set.

Differential Expression (2/2)

Why SCAD/MCP might help:

- If a subset of genes are truly highly differentially expressed, Lasso might over-shrink them.
- SCAD/MCP keep large coefficients closer to their true values ⇒ might approach oracle selection.

Implementation Sketch:

```
ncvreg(X, y, family="binomial", penalty="SCAD")
```

with cross-validation.

Outcome:

- A set of top genes with strong classification power.
- Possibly fewer false negatives if big signals exist.

eQTL Mapping

Goal:

- Identify which SNPs influence gene expression levels.

Challenge:

- Usually $p \gg n$, correlation (LD).
- Single-locus tests ignore multi-SNP effects and correlations.

Penalized Model:

- Linear regression: $y_i = x_i^T \beta + \varepsilon_i$.
- **Ridge/EN** for correlated moderate signals.
- **SCAD/MCP** if a few strong SNPs dominate.

GWAS (Genome-Wide Association Study)

Traditional Approach:

- Millions of univariate SNP tests, correct for multiple comparisons (Bonferroni, FDR).

Penalized Approach:

- Fit a *multi-locus* model with Lasso/EN/SCAD.
- Handles correlation among SNPs, potentially revealing polygenic signals.

Pros & Cons:

- **Pro**: Simultaneous SNP selection, capturing correlated signals.
- **Con**: Large-scale optimization with $p \approx 10^6$ is non-trivial, especially for SCAD/MCP (non-convex).

Survival Analysis (Cox Models)

Cox Partial Likelihood

$$\ell_{\text{Cox}}(\beta) = \sum_{i=1}^{n} \delta_i \Big[x_i^T \beta - \log \Big(\sum_{j \in R_i} e^{x_j^T \beta} \Big) \Big].$$

High-Dimensional Extension:

- Penalty on β: Lasso, EN, SCAD, or MCP.
- Identifies prognostic biomarkers (genes, SNPs) associated with hazard.

SCAD/MCP Benefit:

- Strong hazard signals remain less biased.
- Oracle property if n and signals are adequate.

Multi-Omics Integration

Concept:

- Combine gene expression, DNA methylation, proteomics, metabolomics, etc.
- Summaries from each data type form a large feature matrix.

Penalties:

- Group Lasso / Group SCAD if you want to keep entire blocks of omics or pathways.
- SCAD/MCP can reduce bias in each block if only a few are truly influential.

Implementation:

- Possibly multi-task learning or multi-response regression if multiple omic readouts are outcomes.

Formulation Recap: Linear Regression Case

Linear Model Setup:

$$y_i = x_i^T \beta + \varepsilon_i, \quad i = 1, \ldots, n.$$

Penalized Objective:

$$\min_{\beta} \Big\{ \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} P(\beta_j) \Big\},$$

where $P(\beta_j)$ is one of:

- $|\beta_j|$ (L1)
- β_j^2 (L2)
- SCAD or MCP form

Interpretation:

- Minimizes squared error + penalty cost.
- L1/L2 are convex, SCAD/MCP are non-convex.

Formulation Recap: Logistic Regression Case

Logistic Model Setup:

$$P(y_i = 1) = \frac{1}{1 + e^{-x_i^T \beta}}.$$

Penalized Objective:

$$\min_{\beta} \left\{ -\sum_{i=1}^{n} \left[y_i x_i^T \beta - \log(1 + e^{x_i^T \beta}) \right] + \lambda \sum_{j=1}^{p} P(\beta_j) \right\}.$$

Examples:

- Classify disease vs. control with thousands of gene features.
- Lasso/SCAD/MCP logistic for feature selection + classification.

Formulation Recap: Cox Survival Case

Cox Partial Likelihood:

$$\ell_{\text{Cox}}(\beta) = \sum_{i=1}^{n} \delta_i \Big[x_i^T \beta - \log \big(\sum_{j \in R_i} e^{x_j^T \beta} \big) \Big].$$

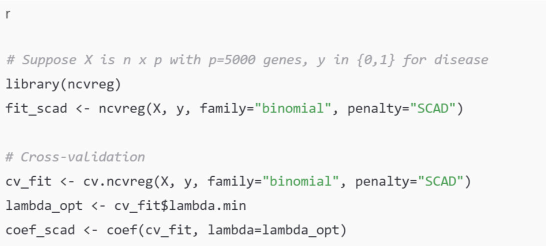
Penalized Objective:

$$\min_{\beta} \Big\{ -\ell_{\text{Cox}}(\beta) + \lambda \sum_{j=1}^{p} P(\beta_j) \Big\}.$$

Why penalize?

- High-dimensional gene sets in survival studies ($p \gg n$).
- Identify genes with strong hazard ratios, reduce others to zero.

Example: SCAD for Differential Expression (Logistic)

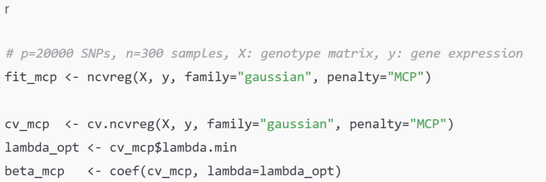


```
r

# Suppose X is n x p with p=5000 genes, y in {0,1} for disease
library(ncvreg)
fit_scad <- ncvreg(X, y, family="binomial", penalty="SCAD")

# Cross-validation
cv_fit <- cv.ncvreg(X, y, family="binomial", penalty="SCAD")
lambda_opt <- cv_fit$lambda.min
coef_scad <- coef(cv_fit, lambda=lambda_opt)
```

Why SCAD?

- If a small subset of genes have large log-odds effects, SCAD preserves them better than Lasso might.
- Potentially meets oracle property if sample size is adequate.

Example: MCP for eQTL (Linear)



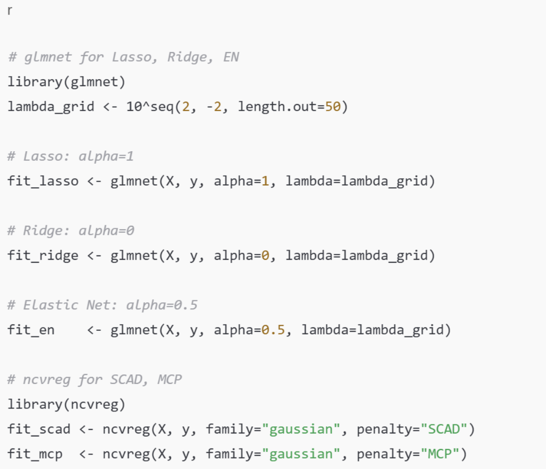
r

```r
# p=20000 SNPs, n=300 samples, X: genotype matrix, y: gene expression
fit_mcp <- ncvreg(X, y, family="gaussian", penalty="MCP")

cv_mcp  <- cv.ncvreg(X, y, family="gaussian", penalty="MCP")
lambda_opt <- cv_mcp$lambda.min
beta_mcp   <- coef(cv_mcp, lambda=lambda_opt)
```

Why MCP?

- If a small number of SNPs have large effect sizes, MCP can zero out the rest while keeping big signals close to true values.
- Lower bias => near-oracle selection if conditions hold.

Lasso vs. Ridge vs. EN vs. SCAD vs. MCP (Overview)

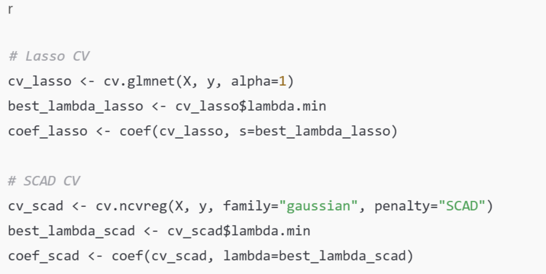


```
r

# glmnet for Lasso, Ridge, EN
library(glmnet)
lambda_grid <- 10^seq(2, -2, length.out=50)

# Lasso: alpha=1
fit_lasso <- glmnet(X, y, alpha=1, lambda=lambda_grid)

# Ridge: alpha=0
fit_ridge <- glmnet(X, y, alpha=0, lambda=lambda_grid)

# Elastic Net: alpha=0.5
fit_en    <- glmnet(X, y, alpha=0.5, lambda=lambda_grid)

# ncvreg for SCAD, MCP
library(ncvreg)
fit_scad <- ncvreg(X, y, family="gaussian", penalty="SCAD")
fit_mcp  <- ncvreg(X, y, family="gaussian", penalty="MCP")
```

Key difference: For SCAD/MCP, `ncvreg` picks its own λ sequence; for glmnet, you can specify or use defaults.

Cross-Validation and Coefficients

Cross-Validation Examples:



```
r


# Lasso CV
cv_lasso <- cv.glmnet(X, y, alpha=1)
best_lambda_lasso <- cv_lasso$lambda.min
coef_lasso <- coef(cv_lasso, s=best_lambda_lasso)

# SCAD CV
cv_scad <- cv.ncvreg(X, y, family="gaussian", penalty="SCAD")
best_lambda_scad <- cv_scad$lambda.min
coef_scad <- coef(cv_scad, lambda=best_lambda_scad)
```

Interpretation:

- Compare how many features remain for each method.
- Are large signals fully preserved or shrunk heavily?

Coefficient Path Plots (Conceptual)

- Plot each coefficient β_j vs. $\log(\lambda)$.
- **Ridge**: all coefficients shrink smoothly but never hit zero.
- **Lasso**: some lines cross zero at moderate λ.
- **SCAD/MCP**: large coefficients flatten at certain λ (reduced bias).

Benefit:

- Helps visualize which features remain robust to penalty changes.
- Large signals may remain stable across λ, small signals drop out early.

Group Lasso (1/3)

Motivation:

- Sometimes features come in predefined groups (e.g., genes in same pathway).
- We want to select or drop entire groups, not just individual features.

Penalty (Group Lasso):

$$\Omega(\beta) = \sum_{g=1}^{G} \sqrt{\|\beta_g\|_2^2},$$

where β_g is the subvector of coefficients in group g.

Group Lasso (2/3)

Interpretation:

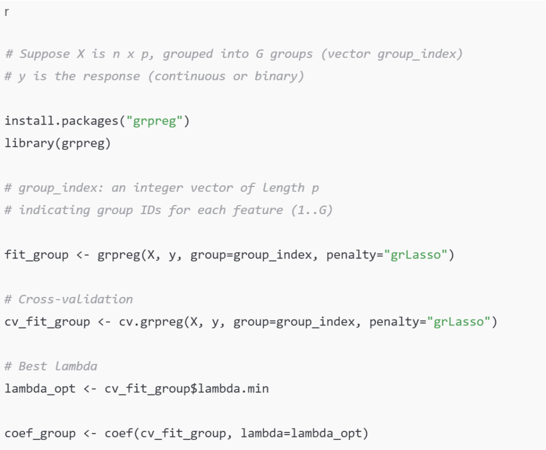
- Group-level sparsity: entire pathways or functional modules can be dropped if uninformative.
- Encourages correlated features in the same group to be kept or removed together.

Bioinformatics Example:

- Pathway-based gene sets, or multi-omics blocks.
- e.g., keep all methylation sites in a region if beneficial.

Group Lasso (3/3) - R Code with

Example using grpreg **package**:



```
r

# Suppose X is n x p, grouped into G groups (vector group_index)
# y is the response (continuous or binary)

install.packages("grpreg")
library(grpreg)

# group_index: an integer vector of length p
# indicating group IDs for each feature (1..G)

fit_group <- grpreg(X, y, group=group_index, penalty="grLasso")

# Cross-validation
cv_fit_group <- cv.grpreg(X, y, group=group_index, penalty="grLasso")

# Best lambda
lambda_opt <- cv_fit_group$lambda.min

coef_group <- coef(cv_fit_group, lambda=lambda_opt)
```

Outcome:

- Groups can be zeroed out entirely.
- Facilitates pathway-level interpretation in genomics.

Applications in Bioinformatics

Penalized likelihood for scRNA-seq data analysis

- **UMI count data**
  - For gene g in cell c, the UMI count is x_{gc}
- **What's the distribution of x_{gc}?**
  - Binomial distribution
  - $x_{gc} \sim NB(\mu_{gc}, \theta_g), \quad \ln \mu_{gc} \; = \; \beta_{g0} + \ln n_c$
  - θ_g is the gene-specific dispersion parameter
  - $n_c = \sum_g x_{gc}$ is the total sequencing depth
  - Variance of NB: $\mu_{gc} \; + \; \mu_{gc}^2/\theta_g$

Applications in Bioinformatics

Sparse logistic regression in cancer classification

- **Data: leukemia patient samples**
  - Acute lymphoblast leukemia (ALL), 49 samples
  - Acute myeloid leukemia (AML), 23 samples
  - Each sample has the profile of 7129 genes
  - Data available at `https://search.r-project.org/CRAN/refmans/propOverlap/html/leukaemia.html`
- **Aim:** leukemia subtype classification & gene selection

Applications in Bioinformatics

Sparse logistic regression in cancer classification

- Consider a general binary classification problem:

$$\{(y_i, x_i)\}_{i=1}^n, \quad y_i \in \{0,1\},\ x_i \in \mathbb{R}^p$$

- The (linear) logistic regression model assumes:

$$\Pr(y = 1 \mid x) \ = \ \frac{e^{x^T\beta}}{1 + e^{x^T\beta}}$$

Applications in Bioinformatics

Sparse logistic regression in cancer classification

- The logistic model is fitted by minimizing the *negative binomial log-likelihood* of the data:

$$\min_{\beta} \; -\ell(\beta) \; + \; \lambda \|\beta\|_1 \quad (12)$$

- $\ell(\beta) = \sum_{i=1}^{n} \Big[y_i \, (x_i^T \beta) \; - \; \ln\big(1 + e^{x_i^T \beta}\big) \Big]$
  - $\|\beta\|_1$ is the penalty term for sparsity
  - λ is the regularization parameter

Applications in Bioinformatics

Penalized likelihood for scRNA-seq data analysis

- **UMI count data**
  - For gene g in cell c, the UMI count is x_{gc}
- **What's the distribution of x_{gc}?**
  - Zero-inflated mixture distribution

$$\Pr(x_{gc} = x) \;=\; (1 - \pi_g)\, I(x = 0) \;+\; \pi_g\, I(x \neq 0)\, F\big(x \mid \mu_{gc}, \sigma_g^2\big)$$

Applications in Bioinformatics

Penalized likelihood for scRNA-seq data analysis

- **Penalization in scRNA-seq data analysis?**
  - Clustering / cell-cell subgroup detection
  - Gene selection
  - Other tasks

Mixed Models & Large-Scale HPC

- Mixed models handle repeated measures, random effects for batch.
- Penalized approach can zero out fixed effects among many genes.
- For extremely large p, HPC or big data solutions needed.

Bayesian Interpretation

- L2 $\leftrightarrow$ Gaussian prior, L1 $\leftrightarrow$ Laplace prior.
- SCAD/MCP $\leftrightarrow$ more complex priors that flatten for large $|\theta|$.
- Bayesian frameworks can provide posterior distributions on coefficients.

Comparison Recap: Standard vs. Advanced Penalties

- Lasso/EN: simpler, well-known, widely used, can handle large p quickly.
- SCAD/MCP: better for big signals, near-oracle selection, but costlier optimization.
- Group Lasso: entire groups of features selected or dropped.
- In practice: start with Lasso/EN, then refine with SCAD if you suspect truly large effects or group lasso if group structure is known.

Key Takeaways

- Penalized likelihood essential for $p \gg n$ in bioinformatics.
- Different penalties (L1, L2, EN, SCAD, MCP) suit different correlation/effect size scenarios.
- Group Lasso addresses group-level sparsity (pathways, modules).
- Oracle property: SCAD/MCP can identify large signals effectively but need careful tuning.

Checklist for Analysis

1. Filter data (optional but often helpful).
2. Choose penalty type (L1, L2, EN, SCAD, MCP, group lasso) based on goals.
3. Use cross-validation to pick λ (and γ / a if needed).
4. Assess out-of-sample performance or AUC.
5. Validate results biologically if feasible (lab tests or external cohorts).

Future Directions

- Deep learning + advanced penalties (SCAD, MCP).
- Structured penalties for multi-omics or graph-based knowledge.
- Faster approximate solvers for large p (10k–1M).

Limitations and Pitfalls

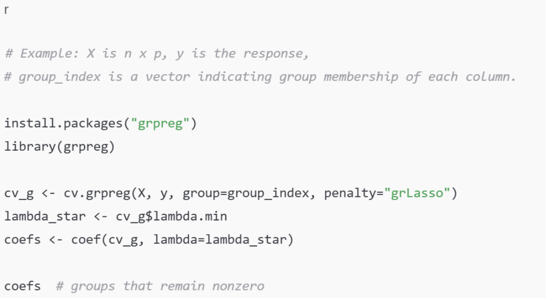
- Non-convex penalties (SCAD/MCP) risk local minima.
- Lasso can be unstable with correlated features.
- Large-scale data $=>$ HPC or approximate methods often needed.
- Oracle property not guaranteed if sample size is too small.

Recommended Resources

- **R Packages**: `glmnet`, `ncvreg`, `grpreg`, `biglasso`.
- **Key Papers**:
  - Tibshirani (1996): Lasso.
  - Fan & Li (2001): SCAD.
  - Zhang (2010): MCP.
  - Yuan & Lin (2006): Group Lasso.
- **Books**: *The Elements of Statistical Learning* (Hastie et al.), *Statistical Learning with Sparsity* (Hastie, Tibshirani, Wainwright).

Additional R Code Example: Group Lasso

Group Lasso with `grpreg`:



```r
r

# Example: X is n x p, y is the response,
# group_index is a vector indicating group membership of each column.

install.packages("grpreg")
library(grpreg)

cv_g <- cv.grpreg(X, y, group=group_index, penalty="grLasso")
lambda_star <- cv_g$lambda.min
coefs <- coef(cv_g, lambda=lambda_star)

coefs  # groups that remain nonzero
```

Interpretation:

- Non-zero group coefficients => entire group is kept.
- Zero group => entire group removed.

Practical Advice for Group Lasso

- Carefully define group boundaries (pathways, gene sets).
- If group sizes vary drastically, consider weighting or adjusting penalty.
- Cross-validate to pick λ.
- Evaluate if entire pathways are relevant or not.

Overall Summary

- Ridge, Lasso, EN: simpler, good baseline.
- SCAD, MCP: advanced, near-oracle selection if conditions are right.
- Group Lasso: entire group-level sparsity for structured data.
- All can be tuned via cross-validation in R packages (`glmnet`, `ncvreg`, `grpreg`).

Q & A

Any Questions?