Recitation 3

Lasso, Ridge, and Elastic Net: A Deeper Dive

DS-GA 1003 Machine Learning

Spring 2021

Feburary 17, 2021

Concept Check

- Explain why feature normalization is important if you are using L1 or L2 regularization.

Agenda

- Repeated Features
- Linearly Dependent Features
- Correlated Features
- The Case Against Sparsity
- Elastic Net
- Coding Exercise

Repeated Features

A Very Simple Model

- Suppose we have one feature $x_1 \in \mathrm{R}$.
- Response variable $y \in \mathrm{R}$.

A Very Simple Model

- Suppose we have one feature $x_1 \in \mathrm{R}$.
- Response variable $y \in \mathrm{R}$.
- Got some data and ran least squares linear regression.
- The ERM is

$$\hat{f}(x_1) = 4x_1.$$

A Very Simple Model

- Suppose we have one feature $x_1 \in \mathrm{R}$.
- Response variable $y \in \mathrm{R}$.
- Got some data and ran least squares linear regression.
- The ERM is
$$\hat{f}(x_1) = 4x_1.$$
- What happens if we get a new feature x_2,
  - but we always have $x_2 = x_1$?

Duplicate Features

- New feature x_2 gives no new information.

Duplicate Features

- New feature x_2 gives no new information.
- ERM is still

$$\hat{f}(x_1, x_2) = 4x_1.$$

- Now there are some more ERMs:

$$\begin{aligned}
\hat{f}(x_1, x_2) &= 2x_1 + 2x_2 \\
\hat{f}(x_1, x_2) &= x_1 + 3x_2 \\
\hat{f}(x_1, x_2) &= 4x_2
\end{aligned}$$

Duplicate Features

- New feature x_2 gives no new information.
- ERM is still

$$\hat{f}(x_1, x_2) = 4x_1.$$

- Now there are some more ERMs:

$$\begin{aligned}\hat{f}(x_1, x_2) &= 2x_1 + 2x_2 \\ \hat{f}(x_1, x_2) &= x_1 + 3x_2 \\ \hat{f}(x_1, x_2) &= 4x_2\end{aligned}$$

- What if we introduce ℓ_1 or ℓ_2 regularization?

Duplicate Features: ℓ_1 and ℓ_2 norms

- $\hat{f}(x_1, x_2) = w_1 x_1 + w_2 x_2$ is an ERM iff $w_1 + w_2 = 4$.

Duplicate Features: ℓ_1 and ℓ_2 norms

- $\hat{f}(x_1, x_2) = w_1 x_1 + w_2 x_2$ is an ERM iff $w_1 + w_2 = 4$.
- Consider the ℓ_1 and ℓ_2 norms of various solutions:

w_1	w_2	$\|w\|_1$	$\|w\|_2^2$
4	0	**4**	16
2	2	**4**	**8**
1	3	**4**	10
-1	5	6	26

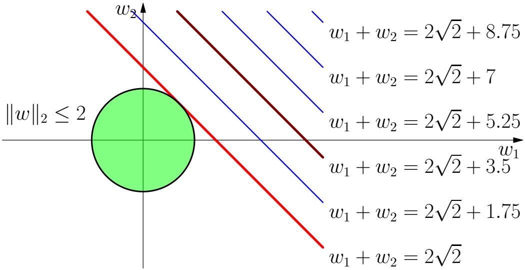
Duplicate Features: ℓ_1 and ℓ_2 norms

- $\hat{f}(x_1, x_2) = w_1 x_1 + w_2 x_2$ is an ERM iff $w_1 + w_2 = 4$.
- Consider the ℓ_1 and ℓ_2 norms of various solutions:

w_1	w_2	$\|w\|_1$	$\|w\|_2^2$
4	0	**4**	16
2	2	**4**	**8**
1	3	**4**	10
-1	5	6	26

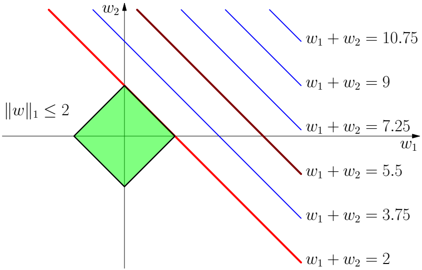
- $\|w\|_1$ doesn't discriminate, as long as all have same sign
- $\|w\|_2^2$ minimized when weight is spread equally
- Picture proof: Level sets of loss are lines of the form $w_1 + w_2 = 4$...

Equal Features, ℓ_2 Constraint



- Suppose the line $w_1 + w_2 = 2\sqrt{2} + 3.5$ corresponds to the empirical risk minimizers.
- Empirical risk increase as we move away from these parameter settings
- Intersection of $w_1 + w_2 = 2\sqrt{2}$ and the norm ball $\|w\|_2 \leq 2$ is ridge solution.
- Note that $w_1 = w_2$ at the solution

Equal Features, ℓ_1 Constraint



- Suppose the line $w_1 + w_2 = 5.5$ corresponds to the empirical risk minimizers.
- Intersection of $w_1 + w_2 = 2$ and the norm ball $\|w\|_1 \leq 2$ is lasso solution.
- Note that the solution set is $\{(w_1, w_2) : w_1 + w_2 = 2, w_1, w_2 \geq 0\}$.

Linearly Dependent Features

Linearly Related Features

- Linear prediction functions: $f(x) = w_1 x_2 + w_2 x_2$
- Same setup, now suppose $x_2 = 2x_1$.

Linearly Related Features

- Linear prediction functions: $f(x) = w_1 x_2 + w_2 x_2$
- Same setup, now suppose $x_2 = 2x_1$.
- Then all functions with $w_1 + 2w_2 = k$ are the same.
  - give same predictions and have same empirical risk

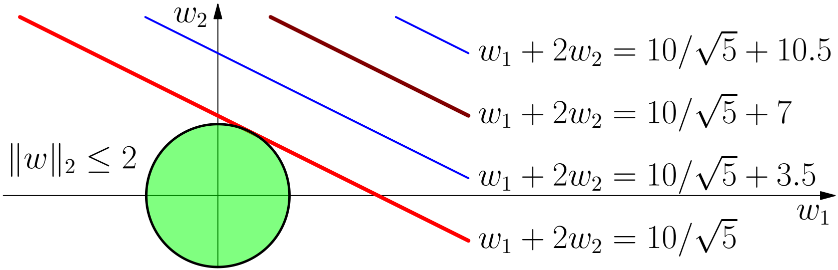
Linearly Related Features

- Linear prediction functions: $f(x) = w_1 x_2 + w_2 x_2$
- Same setup, now suppose $x_2 = 2x_1$.
- Then all functions with $w_1 + 2w_2 = k$ are the same.
  - give same predictions and have same empirical risk
- What function will we select if we do ERM with ℓ_1 or ℓ_2 constraint?

Linearly Related Features

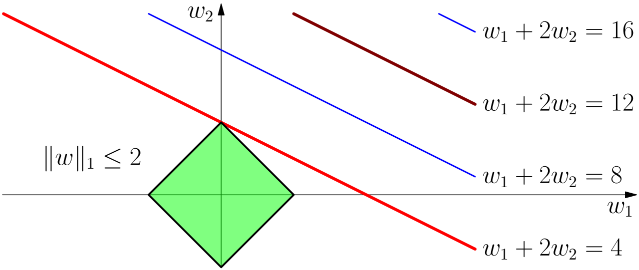
- Linear prediction functions: $f(x) = w_1 x_2 + w_2 x_2$
- Same setup, now suppose $x_2 = 2x_1$.
- Then all functions with $w_1 + 2w_2 = k$ are the same.
  - give same predictions and have same empirical risk
- What function will we select if we do ERM with ℓ_1 or ℓ_2 constraint?
- Compare a solution that just uses w_1 to a solution that just uses w_2...

Linearly Related Features, ℓ_2 Constraint



- $w_1 + 2w_2 = 10/\sqrt{5} + 7$ corresponds to the empirical risk minimizers.
- Intersection of $w_1 + 2w_2 = 10\sqrt{5}$ and the norm ball $\|w\|_2 \leq 2$ is ridge solution.
- At solution, $w_2 = 2w_1$.

Linearly Related Features, ℓ_1 Constraint



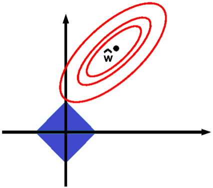
- Intersection of $w_1 + 2w_2 = 4$ and the norm ball $\|w\|_1 \leq 2$ is lasso solution.
- Solution is now a corner of the ℓ_1 ball, corresponding to a sparse solution.

Linearly Dependent Features: Take Away

- For identical features
  - ℓ_1 regularization spreads weight arbitrarily (all weights same sign)
  - ℓ_2 regularization spreads weight evenly
- Linearly related features
  - ℓ_1 regularization chooses variable with larger scale, 0 weight to others
  - ℓ_2 prefers variables with larger scale – spreads weight proportional to scale

Empirical Risk for Square Loss and Linear Predictors

- Recall our discussion of linear predictors $f(x) = w^T x$ and square loss.
- Sets of w giving same empirical risk (i.e. level sets) formed ellipsoids around the ERM.



- With x_1 and x_2 linearly related, $X^T X$ has a 0 eigenvalue.
- So the level set $\left\{ w \mid (w - \hat{w})^T X^T X (w - \hat{w}) = nc \right\}$ is no longer an ellipsoid.
- It's a degenerate ellipsoid – that's why level sets were pairs of lines in this case

KPM Fig. 13.3

Correlated Features

Correlated Features – Same Scale

- Suppose x_1 and x_2 are highly correlated and the same scale.
- This is quite typical in real data, after normalizing data.

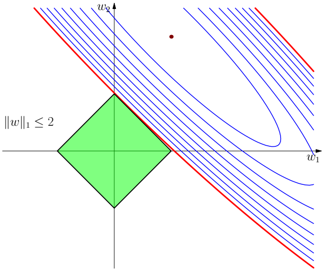
Correlated Features – Same Scale

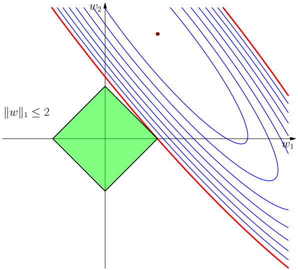
- Suppose x_1 and x_2 are highly correlated and the same scale.
- This is quite typical in real data, after normalizing data.
- Nothing degenerate here, so level sets are ellipsoids.

Correlated Features – Same Scale

- Suppose x_1 and x_2 are highly correlated and the same scale.
- This is quite typical in real data, after normalizing data.
- Nothing degenerate here, so level sets are ellipsoids.
- But, the higher the correlation, the closer to degenerate we get.
- That is, ellipsoids keep stretching out, getting closer to two parallel lines.

Correlated Features, ℓ_1 Regularization





- Intersection could be anywhere on the top right edge.
- Minor perturbations (in data) can drastically change intersection point – very unstable solution.
- Makes division of weight among highly correlated features (of same scale) seem arbitrary.
  - If $x_1 \approx 2x_2$, ellipse changes orientation and we hit a corner. (Which one?)

A Case Against Sparsity

- Suppose there's some unknown value $\theta \in \mathbb{R}$.

A Case Against Sparsity

- Suppose there's some unknown value $\theta \in \mathbb{R}$.
- We get 3 noisy observations of θ:

$$x_1, x_2, x_3 \sim \mathcal{N}(\theta, 1) \quad \text{(i.i.d)}$$

A Case Against Sparsity

- Suppose there's some unknown value $\theta \in \mathbb{R}$.
- We get 3 noisy observations of θ:

$$x_1, x_2, x_3 \sim \mathcal{N}(\theta, 1) \ \text{(i.i.d)}$$

- What's a good estimator $\hat{\theta}$ for θ?

A Case Against Sparsity

- Suppose there's some unknown value $\theta \in \mathbb{R}$.
- We get 3 noisy observations of θ:

$$x_1, x_2, x_3 \sim \mathcal{N}(\theta, 1) \ \text{(i.i.d)}$$

- What's a good estimator $\hat{\theta}$ for θ?
- Would you prefer $\hat{\theta} = x_1$ or $\hat{\theta} = \frac{1}{3}(x_1 + x_2 + x_3)$?

Estimator Performance Analysis

- $\mathrm{Exp}\left[x_1\right] = \theta$ and $\mathrm{Exp}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \theta$. So both unbiased.

Estimator Performance Analysis

- $\mathrm{Exp}\left[x_1\right] = \theta$ and $\mathrm{Exp}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \theta$. So both unbiased.
- $\mathrm{Var}\left[x_1\right] =$

Estimator Performance Analysis

- $\mathrm{Exp}\left[x_1\right] = \theta$ and $\mathrm{Exp}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \theta$. So both unbiased.
- $\mathrm{Var}\left[x_1\right] = 1$.

Estimator Performance Analysis

- $\mathrm{Exp}\left[x_1\right] = \theta$ and $\mathrm{Exp}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \theta$. So both unbiased.
- $\mathrm{Var}\left[x_1\right] = 1$.
- $\mathrm{Var}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] =$

Estimator Performance Analysis

- $\text{Exp}\left[x_1\right] = \theta$ and $\text{Exp}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \theta$. So both unbiased.
- $\text{Var}\left[x_1\right] = 1$.
- $\text{Var}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \frac{1}{9}\left(1 + 1 + 1\right) = \frac{1}{3}$.

Estimator Performance Analysis

- $\mathrm{Exp}\left[x_1\right] = \theta$ and $\mathrm{Exp}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \theta$. So both unbiased.
- $\mathrm{Var}\left[x_1\right] = 1$.
- $\mathrm{Var}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \frac{1}{9}\left(1 + 1 + 1\right) = \frac{1}{3}$.
- Average has a smaller variance — the independent errors cancel each other out.

Estimator Performance Analysis

- $\mathrm{Exp}\left[x_1\right] = \theta$ and $\mathrm{Exp}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \theta$. So both unbiased.
- $\mathrm{Var}\left[x_1\right] = 1$.
- $\mathrm{Var}\left[\frac{1}{3}\left(x_1 + x_2 + x_3\right)\right] = \frac{1}{9}\left(1 + 1 + 1\right) = \frac{1}{3}$.
- Average has a smaller variance — the independent errors cancel each other out.
- Similar thing happens in regression with correlated features:
  - e.g. If 3 features are correlated, we could keep just one of them.
  - But we can potentially do better by using all 3.

Example with highly correlated features

- Model in words:
  - y is some unknown linear combination of z_1 and z_2.
  - But we don't observe z_1 and z_2 directly.

Example with highly correlated features

- Model in words:
  - y is some unknown linear combination of z_1 and z_2.
  - But we don't observe z_1 and z_2 directly.
  - We get 3 noisy observations of z_1, call them x_1, x_2, x_3.
  - We get 3 noisy observations of z_2, call them x_4, x_5, x_6.

Example with highly correlated features

- Model in words:
  - y is some unknown linear combination of z_1 and z_2.
  - But we don't observe z_1 and z_2 directly.
  - We get 3 noisy observations of z_1, call them x_1, x_2, x_3.
  - We get 3 noisy observations of z_2, call them x_4, x_5, x_6.
- We want to predict y from our noisy observations.

Example with highly correlated features

- Model in words:
    - y is some unknown linear combination of z_1 and z_2.
    - But we don't observe z_1 and z_2 directly.
    - We get 3 noisy observations of z_1, call them x_1, x_2, x_3.
    - We get 3 noisy observations of z_2, call them x_4, x_5, x_6.
- We want to predict y from our noisy observations.
- That is, we want an estimator $\hat{y} = f(x_1, x_2, x_3, x_4, x_5, x_6)$ for estimating y.

Example from Section 4.2 in Hastie et al's *Statistical Learning with Sparsity*.

Example with highly correlated features

- Suppose (x, y) generated as follows:

$$\begin{aligned} z_1, z_2 &\sim \mathcal{N}(0,1) \text{ (independent)} \\ \epsilon_0, \epsilon_1, \ldots, \epsilon_6 &\sim \mathcal{N}(0,1) \text{ (independent)} \end{aligned}$$

Example with highly correlated features

- Suppose (x, y) generated as follows:

$$\begin{aligned} z_1, z_2 &\sim \mathcal{N}(0,1) \text{ (independent)} \\ \epsilon_0, \epsilon_1, \ldots, \epsilon_6 &\sim \mathcal{N}(0,1) \text{ (independent)} \\ y &= 3z_1 - 1.5z_2 + 2\epsilon_0 \end{aligned}$$

Example with highly correlated features

- Suppose (x, y) generated as follows:

$$
\begin{aligned}
z_1, z_2 &\sim \mathcal{N}(0,1) \text{ (independent)} \\
\epsilon_0, \epsilon_1, \ldots, \epsilon_6 &\sim \mathcal{N}(0,1) \text{ (independent)} \\
y &= 3z_1 - 1.5z_2 + 2\epsilon_0 \\
x_j &= \begin{cases} z_1 + \epsilon_j/5 & \text{for } j = 1,2,3 \\ z_2 + \epsilon_j/5 & \text{for } j = 4,5,6 \end{cases}
\end{aligned}
$$

Example with highly correlated features

- Suppose (x, y) generated as follows:

$$
\begin{aligned}
z_1, z_2 &\sim \mathcal{N}(0,1) \text{ (independent)} \\
\epsilon_0, \epsilon_1, \ldots, \epsilon_6 &\sim \mathcal{N}(0,1) \text{ (independent)} \\
y &= 3z_1 - 1.5z_2 + 2\epsilon_0 \\
x_j &= \begin{cases} z_1 + \epsilon_j/5 & \text{for } j = 1,2,3 \\ z_2 + \epsilon_j/5 & \text{for } j = 4,5,6 \end{cases}
\end{aligned}
$$

- Generated a sample of $((x_1, \ldots, x_6), y)$ pairs of size $n = 100$.

Example with highly correlated features

- Suppose (x, y) generated as follows:

$$
\begin{aligned}
z_1, z_2 &\sim \mathcal{N}(0,1) \text{ (independent)} \\
\epsilon_0, \epsilon_1, \ldots, \epsilon_6 &\sim \mathcal{N}(0,1) \text{ (independent)} \\
y &= 3z_1 - 1.5z_2 + 2\epsilon_0 \\
x_j &= \begin{cases} z_1 + \epsilon_j/5 & \text{for } j = 1,2,3 \\ z_2 + \epsilon_j/5 & \text{for } j = 4,5,6 \end{cases}
\end{aligned}
$$

- Generated a sample of $((x_1, \ldots, x_6), y)$ pairs of size $n = 100$.
- That is, we want an estimator $\hat{y} = f(x_1, x_2, x_3, x_4, x_5, x_6)$ that is good for estimating y.

Example with highly correlated features

- Suppose (x, y) generated as follows:

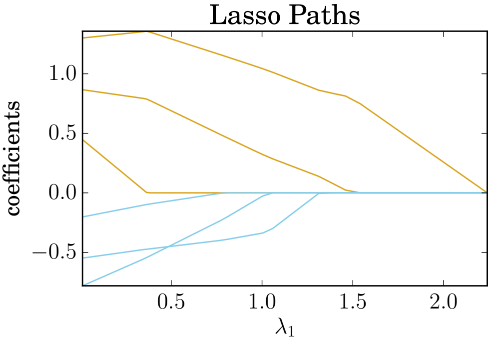
$$
\begin{aligned}
z_1, z_2 &\sim \mathcal{N}(0,1) \text{ (independent)} \\
\epsilon_0, \epsilon_1, \ldots, \epsilon_6 &\sim \mathcal{N}(0,1) \text{ (independent)} \\
y &= 3z_1 - 1.5z_2 + 2\epsilon_0 \\
x_j &= \begin{cases} z_1 + \epsilon_j/5 & \text{for } j = 1,2,3 \\ z_2 + \epsilon_j/5 & \text{for } j = 4,5,6 \end{cases}
\end{aligned}
$$

- Generated a sample of $((x_1, \ldots, x_6), y)$ pairs of size $n = 100$.
- That is, we want an estimator $\hat{y} = f(x_1, x_2, x_3, x_4, x_5, x_6)$ that is good for estimating y.
- **High feature correlation**: Correlations within the groups of x's is around 0.97.

Example from Section 4.2 in Hastie et al's *Statistical Learning with Sparsity*.

Example with highly correlated features

- Lasso regularization paths:



- Lines with the same color correspond to features with essentially the same information
- Distribution of weight among them seems almost arbitrary

Hedge Bets When Variables Highly Correlated

- When variables are highly correlated (and same scale – assume we've standardized features),
  - we want to give them roughly the same weight.

Hedge Bets When Variables Highly Correlated

- When variables are highly correlated (and same scale – assume we've standardized features),
  - we want to give them roughly the same weight.
- Why?
  - Let their errors cancel out

Hedge Bets When Variables Highly Correlated

- When variables are highly correlated (and same scale – assume we've standardized features),
  - we want to give them roughly the same weight.
- Why?
  - Let their errors cancel out
- How can we get the weight spread more evenly?

Elastic Net

Elastic Net

- The **elastic net** combines lasso and ridge penalties:

$$\hat{w} = \underset{w \in ^d}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2 + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2$$

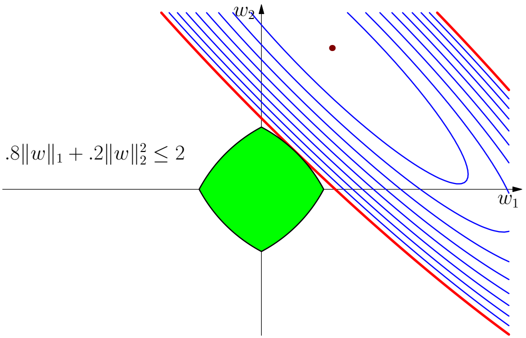
Elastic Net

- The **elastic net** combines lasso and ridge penalties:

$$\hat{w} = \underset{w \in ^d}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2 + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2$$

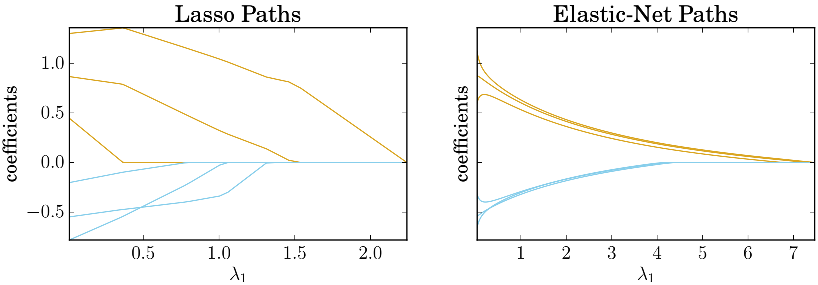
- We expect correlated random variables to have similar coefficients.

Highly Correlated Features, Elastic Net Constraint



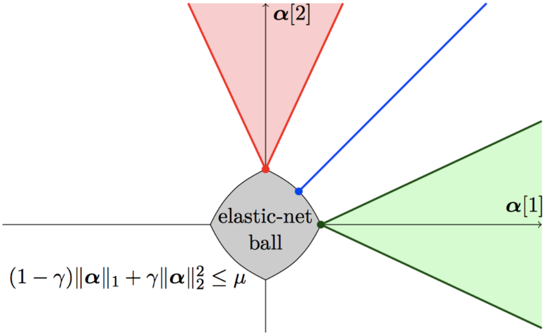
- Elastic net solution is closer to $w_2 = w_1$ line, despite high correlation.

Elastic Net Results on Model



- Lasso on left; Elastic net on right.
- Ratio of ℓ_2 to ℓ_1 regularization roughly $2:1$.

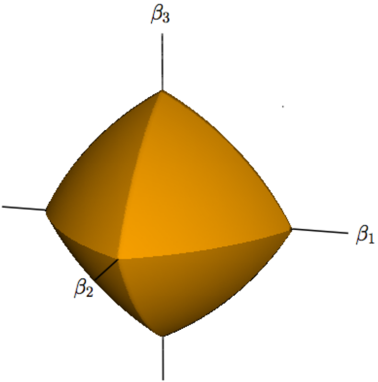
Elastic Net - "Sparse Regions"



- Suppose design matrix X is orthogonal, so $X^T X = I$, and contours are circles (and features uncorrelated)
- Then OLS solution in green or red regions implies elastic-net constrained solution will be at corner

Fig from Mairal et al.'s Sparse Modeling for Image and Vision Processing Fig 1.9

Elastic Net vs Lasso Norm Ball



From Figure 4.2 of Hastie et al's *Statistical Learning with Sparsity*.

$\ell_{1.2}$ vs Elastic Net

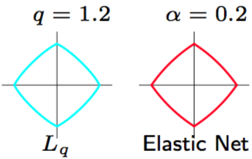


FIGURE 3.13. *Contours of constant value of* $\sum_j |\beta_j|^q$ *for* $q = 1.2$ *(left plot), and the elastic-net penalty* $\sum_j (\alpha\beta_j^2 + (1-\alpha)|\beta_j|)$ *for* $\alpha = 0.2$ *(right plot). Although visually very similar, the elastic-net has sharp (non-differentiable) corners, while the* $q = 1.2$ *penalty does not.*

From Hastie et al's *Elements of Statistical Learning*.

References

- DS-GA 1003 Machine Learning Spring 2019