$\ell_1$ and $\ell_2$ Regularization

David S. Rosenberg

New York University

January 30, 2018
Tikhonov and Ivanov Regularization
Hypothesis Spaces

- We’ve spoken vaguely about “bigger” and “smaller” hypothesis spaces
- In practice, convenient to work with a nested sequence of spaces:

  \[ F_1 \subset F_2 \subset F_n \cdots \subset F \]

Polynomial Functions

- \( F = \{ \text{all polynomial functions} \} \)
- \( F_d = \{ \text{all polynomials of degree} \leq d \} \)
Complexity Measures for Decision Functions

- Number of variables / features
- Depth of a decision tree
- Degree of polynomial
- How about for **linear** decision functions, i.e. \( x \mapsto w^T x = w_1 x_1 + \cdots + w_d x_d \)?
  - \( \ell_0 \) complexity: number of non-zero coefficients \( \sum_{i=1}^{d} 1(w_i \neq 0) \).
  - \( \ell_1 \) “lasso” complexity: \( \sum_{i=1}^{d} |w_i| \), for coefficients \( w_1, \ldots, w_d \)
  - \( \ell_2 \) “ridge” complexity: \( \sum_{i=1}^{d} w_i^2 \) for coefficients \( w_1, \ldots, w_d \)
Nested Hypothesis Spaces from Complexity Measure

- Hypothesis space: $\mathcal{F}$
- Complexity measure $\Omega : \mathcal{F} \rightarrow [0, \infty)$
- Consider all functions in $\mathcal{F}$ with complexity at most $r$:
  \[
  \mathcal{F}_r = \{ f \in \mathcal{F} \mid \Omega(f) \leq r \}
  \]
- Increasing complexities: $r = 0, 1.2, 2.6, 5.4, \ldots$ gives nested spaces:
  \[
  \mathcal{F}_0 \subset \mathcal{F}_{1.2} \subset \mathcal{F}_{2.6} \subset \mathcal{F}_{5.4} \subset \cdots \subset \mathcal{F}
  \]
Constrained Empirical Risk Minimization

Constrained ERM (Ivanov regularization)

For complexity measure $\Omega : \mathcal{F} \rightarrow [0, \infty)$ and fixed $r \geq 0$,

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

such that $\Omega(f) \leq r$

- Choose $r$ using validation data or cross-validation.
- Each $r$ corresponds to a different hypothesis spaces. Could also write:

$$\min_{f \in \mathcal{F}_r} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$
Penalized ERM (Tikhonov regularization)

For complexity measure $\Omega : \mathcal{F} \rightarrow [0, \infty)$ and fixed $\lambda \geq 0$,

$$
\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda \Omega(f)
$$

- Choose $\lambda$ using validation data or cross-validation.
- (Ridge regression in homework is of this form.)
Ivanov vs Tikhonov Regularization

- Let $L: \mathcal{F} \to \mathbb{R}$ be any performance measure of $f$
  - e.g. $L(f)$ could be the empirical risk of $f$
- For many $L$ and $\Omega$, Ivanov and Tikhonov are “equivalent”.
- What does this mean?
  - Any solution $f^*$ you could get from Ivanov, can also get from Tikhonov.
  - Any solution $f^*$ you could get from Tikhonov, can also get from Ivanov.
- In practice, both approaches are effective.
- Tikhonov convenient because it’s *unconstrained* minimization.

Can get conditions for equivalence from Lagrangian duality theory – details in homework.
Ivanov vs Tikhonov Regularization (Details)

Ivanov and Tikhonov regularization are equivalent if:

1. For any choice of $r > 0$, any Ivanov solution

$$f_r^* \in \arg \min_{f \in \mathcal{F}} L(f) \text{ s.t. } \Omega(f) \leq r$$

is also a Tikhonov solution for some $\lambda > 0$. That is, $\exists \lambda > 0$ such that

$$f_r^* \in \arg \min_{f \in \mathcal{F}} L(f) + \lambda \Omega(f).$$

2. Conversely, for any choice of $\lambda > 0$, any Tikhonov solution:

$$f_\lambda^* \in \arg \min_{f \in \mathcal{F}} L(f) + \lambda \Omega(f)$$

is also an Ivanov solution for some $r > 0$. That is, $\exists r > 0$ such that

$$f_\lambda^* \in \arg \min_{f \in \mathcal{F}} L(f) \text{ s.t. } \Omega(f) \leq r.$$
$\ell_1$ and $\ell_2$ Regularization
Linear Least Squares Regression

Consider linear models

$$\mathcal{F} = \{ f : \mathbb{R}^d \to \mathbb{R} \mid f(x) = w^T x \text{ for } w \in \mathbb{R}^d \}$$

- Loss: $\ell(\hat{y}, y) = (y - \hat{y})^2$
- Training data $\mathcal{D}_n = ((x_1, y_1), \ldots, (x_n, y_n))$
- Linear least squares regression is ERM for $\ell$ over $\mathcal{F}$:

$$\hat{w} = \arg\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \{ w^T x_i - y_i \}^2$$

- Can overfit when $d$ is large compared to $n$.
- e.g.: $d \gg n$ very common in Natural Language Processing problems (e.g. a 1M features for 10K documents).
Ridge Regression: Workhorse of Modern Data Science

Ridge Regression (Tikhonov Form)

The ridge regression solution for regularization parameter $\lambda \geq 0$ is

$$\hat{w} = \arg\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left( w^T x_i - y_i \right)^2 + \lambda \|w\|_2^2,$$

where $\|w\|_2^2 = w_1^2 + \cdots + w_d^2$ is the square of the $\ell_2$-norm.

Ridge Regression (Ivanov Form)

The ridge regression solution for complexity parameter $r \geq 0$ is

$$\hat{w} = \arg\min_{\|w\|_2^2 \leq r^2} \frac{1}{n} \sum_{i=1}^{n} \left( w^T x_i - y_i \right)^2.$$
How does $\ell_2$ regularization induce “regularity”?

- For $\hat{f}(x) = \hat{w}^T x$, $\hat{f}$ is **Lipschitz continuous** with Lipschitz constant $L = \|\hat{w}\|_2$.
- That is, when moving from $x$ to $x + h$, $\hat{f}$ changes no more than $L\|h\|$.
- So $\ell_2$ regularization controls the maximum rate of change of $\hat{f}$.
- Proof:

$$
\left| \hat{f}(x + h) - \hat{f}(x) \right| = |\hat{w}^T (x + h) - \hat{w}^T x| = |\hat{w}^T h|
\leq \|\hat{w}\|_2 \|h\|_2 \text{(Cauchy-Schwarz inequality)}
$$

- Since $\|\hat{w}\|_1 \geq \|\hat{w}\|_2$, an $\ell_1$ constraint will also give a Lipschitz bound.
Ridge Regression: Regularization Path

\[
\hat{w}_r = \arg \min_{\|w\|_2^2 \leq r^2} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2
\]

\[
\hat{w} = \hat{w}_\infty = \text{Unconstrained ERM}
\]

- For \( r = 0 \), \( \|\hat{w}_r\|_2 / \|\hat{w}\|_2 = 0 \).
- For \( r = \infty \), \( \|\hat{w}_r\|_2 / \|\hat{w}\|_2 = 1 \)

---

Modified from Hastie, Tibshirani, and Wainwright’s *Statistical Learning with Sparsity*, Fig 2.1. About predicting crime in 50 US cities.
Lasso Regression: Workhorse (2) of Modern Data Science

Lasso Regression (Tikhonov Form)

The lasso regression solution for regularization parameter $\lambda \geq 0$ is

$$\hat{w} = \arg\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2 + \lambda \| w \|_1,$$

where $\| w \|_1 = |w_1| + \cdots + |w_d|$ is the $\ell_1$-norm.

Lasso Regression (Ivanov Form)

The lasso regression solution for complexity parameter $r \geq 0$ is

$$\hat{w} = \arg\min_{\| w \|_1 \leq r} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2.$$
Lasso Regression: Regularization Path

\[ \hat{w}_r = \arg \min_{\|w\|_1 \leq r} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2 \]
\[ \hat{w} = \hat{w}_\infty = \text{Unconstrained ERM} \]

- For \( r = 0 \), \( \|\hat{w}_r\|_1 / \|\hat{w}\|_1 = 0 \).
- For \( r = \infty \), \( \|\hat{w}_r\|_1 / \|\hat{w}\|_1 = 1 \)

Modified from Hastie, Tibshirani, and Wainwright’s *Statistical Learning with Sparsity*, Fig 2.1. About predicting crime in 50 US cities.
Ridge vs. Lasso: Regularization Paths

Modified from Hastie, Tibshirani, and Wainwright’s *Statistical Learning with Sparsity*, Fig 2.1. About predicting crime in 50 US cities.
Coefficient are 0 $\implies$ don’t need those features. What’s the gain?

- Time/expense to compute/buy features
- Memory to store features (e.g. real-time deployment)
- Identifies the important features
- Better prediction? sometimes
- As a feature-selection step for training a slower non-linear model
Ivanov and Tikhonov Equivalent?

- For ridge regression and lasso regression (and much more)
  - the Ivanov and Tikhonov formulations are equivalent
  - [Optional homework problem, upcoming.]
- We will use whichever form is most convenient.
Why does Lasso regression give sparse solutions?
Illustrate affine prediction functions in parameter space.
The $\ell_1$ and $\ell_2$ Norm Constraints

- For visualization, restrict to 2-dimensional input space
- $\mathcal{F} = \{ f(x) = w_1 x_1 + w_2 x_2 \}$ (linear hypothesis space)
- Represent $\mathcal{F}$ by $\{(w_1, w_2) \in \mathbb{R}^2\}$.

\begin{align*}
\ell_2 \text{ contour: } & \quad w_1^2 + w_2^2 = r \\
\ell_1 \text{ contour: } & \quad |w_1| + |w_2| = r
\end{align*}

Where are the “sparse” solutions?
The Famous Picture for $\ell_1$ Regularization

- $f^*_r = \arg\min_{w \in \mathbb{R}^2} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2$ subject to $|w_1| + |w_2| \leq r$

- Blue region: Area satisfying complexity constraint: $|w_1| + |w_2| \leq r$
- Red lines: contours of $\hat{R}_n(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2$. 

KPM Fig. 13.3
The Empirical Risk for Square Loss

- Denote the empirical risk of $f(x) = w^T x$ by

$$
\hat{R}_n(w) = \frac{1}{n} \|Xw - y\|^2,
$$

where $X$ is the design matrix.

- $\hat{R}_n$ is minimized by $\hat{w} = (X^T X)^{-1} X^T y$, the OLS solution.

- What does $\hat{R}_n$ look like around $\hat{w}$?
By “completing the square”, we can show for any \( w \in \mathbb{R}^d \):

\[
\hat{R}_n(w) = \frac{1}{n} (w - \hat{w})^T X^T X (w - \hat{w}) + \hat{R}_n(\hat{w})
\]

Set of \( w \) with \( \hat{R}_n(w) \) exceeding \( \hat{R}_n(\hat{w}) \) by \( c > 0 \) is

\[
\left\{ w \mid \hat{R}_n(w) = c + \hat{R}_n(\hat{w}) \right\} = \left\{ w \mid (w - \hat{w})^T X^T X (w - \hat{w}) = nc \right\},
\]

which is an ellipsoid centered at \( \hat{w} \).

We’ll derive this in homework.
The Famous Picture for $\ell_2$ Regularization

- $f_r^* = \arg\min_{w \in \mathbb{R}^2} \sum_{i=1}^{n} (w^T x_i - y_i)^2$ subject to $w_1^2 + w_2^2 \leq r$

- Blue region: Area satisfying complexity constraint: $w_1^2 + w_2^2 \leq r$
- Red lines: contours of $\hat{R}_n(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2$.

KPM Fig. 13.3
Why are Lasso Solutions Often Sparse?

- Suppose design matrix $X$ is orthogonal, so $X^T X = I$, and contours are circles.
- Then OLS solution in green or red regions implies $\ell_1$ constrained solution will be at corner.

---

Fig from Mairal et al.'s Sparse Modeling for Image and Vision Processing Fig 1.6
The \((\ell_q)^q\) Constraint

- Generalize to \(\ell_q : (\|w\|_q)^q = |w_1|^q + |w_2|^q\).
- Note: \(\|w\|_q\) is a norm if \(q \geq 1\), but not for \(q \in (0, 1)\).
- \(\mathcal{F} = \{f(x) = w_1 x_1 + w_2 x_2\}\).
- Contours of \(\|w\|_q^q = |w_1|^q + |w_2|^q\):

\[
\begin{array}{c}
q = 4 \\
q = 2 \\
q = 1 \\
q = 0.5 \\
q = 0.1
\end{array}
\]
\( \ell_q \) Even Sparser

Suppose design matrix \( X \) is orthogonal, so \( X^T X = I \), and contours are circles.

Then OLS solution in green or red regions implies \( \ell_q \) constrained solution will be at corner -ball constraint is not convex, so more difficult to optimize.

Fig from Mairal et al.’s Sparse Modeling for Image and Vision Processing Fig 1.9
From Quora: “Why is L1 regularization supposed to lead to sparsity than L2? [sic]”

Does this picture have any interpretation that makes sense? (Aren’t those lines supposed to be ellipses?)

Yes... we can revisit.
Finding the Lasso Solution: Lasso as Quadratic Program
How to find the Lasso solution?

- How to solve the Lasso?

\[
\min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_1
\]

- \( \|w\|_1 = |w_1| + |w_2| \) is not differentiable!
Consider any number $a \in \mathbb{R}$.

Let the **positive part** of $a$ be

$$a^+ = a1(a \geq 0).$$

Let the **negative part** of $a$ be

$$a^- = -a1(a \leq 0).$$

Do you see why $a^+ \geq 0$ and $a^- \geq 0$?

How do you write $a$ in terms of $a^+$ and $a^-$?

How do you write $|a|$ in terms of $a^+$ and $a^-$?
How to find the Lasso solution?

- The Lasso problem
  \[
  \min_{w \in \mathbb{R}^d} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_1
  \]

- Replace each \(w_i\) by \(w_i^+ - w_i^-\).
- Write \(w^+ = (w_1^+, \ldots, w_d^+)\) and \(w^- = (w_1^-, \ldots, w_d^-)\).
The Lasso as a Quadratic Program

We will show: substituting $w = w^+ - w^-$ and $|w| = w^+ + w^-$ gives an equivalent problem:

$$\begin{align*}
\min_{w^+, w^-} \quad & \sum_{i=1}^n \left( (w^+ - w^-)^T x_i - y_i \right)^2 + \lambda 1^T (w^+ + w^-) \\
\text{subject to} \quad & w_i^+ \geq 0 \text{ for all } i, \\
& w_i^- \geq 0 \text{ for all } i,
\end{align*}$$

- Objective is differentiable (in fact, convex and quadratic)
- $2d$ variables vs $d$ variables and $2d$ constraints vs no constraints
- A “quadratic program”: a convex quadratic objective with linear constraints.
  - Could plug this into a generic QP solver.
Possible point of confusion

Equivalent to lasso problem:

$$\min_{w^+, w^-} \sum_{i=1}^{n} \left( (w^+ - w^-)^T x_i - y_i \right)^2 + \lambda 1^T (w^+ + w^-)$$

subject to $w_i^+ \geq 0$ for all $i$, $w_i^- \geq 0$ for all $i$,

- When we plug this optimization problem into a QP solver,
  - it just sees $2d$ variables and $2d$ constraints.
  - Doesn’t know we want $w_i^+$ and $w_i^-$ to be positive and negative parts of $w_i$.

- Turns out – they will come out that way as a result of the optimization!

- But to eliminate confusion, let’s start by calling them $a_i$ and $b_i$ and prove our claim...
The Lasso as a Quadratic Program

Lasso problem is trivially equivalent to the following:

$$\min_w \min_{a,b} \sum_{i=1}^n \left( (a-b)^T x_i - y_i \right)^2 + \lambda 1^T (a+b)$$

subject to  
\begin{align*} 
    a_i &\geq 0 \text{ for all } i \quad b_i \geq 0 \text{ for all } i, \\
    a-b &= w \\
    a+b &= |w|
\end{align*}

- **Claim**: Don’t need constraint $a+b = |w|$.
- $a' \leftarrow a - \min(a,b)$ and $b' \leftarrow b - \min(a,b)$ at least as good
- So if $a$ and $b$ are minimizers, at least one is 0.
- Since $a-b = w$, we must have $a = w^+$ and $b = w^-$. So also $a+b = |w|$. 

The Lasso as a Quadratic Program

\[
\min_w \min_{a,b} \sum_{i=1}^n \left( (a - b)^T x_i - y_i \right)^2 + \lambda 1^T (a + b)
\]
subject to \( a_i \geq 0 \) for all \( i \), \( b_i \geq 0 \) for all \( i \),
\( a - b = w \)

- Claim: Can remove \( \min_w \) and the constraint \( a - b = w \).

- One way to see this is by switching the order of minimization...
The Lasso as a Quadratic Program

\[
\min_{a,b} \min_w \sum_{i=1}^{n} \left( (a - b)^T x_i - y_i \right)^2 + \lambda 1^T (a + b)
\]
subject to \( a_i \geq 0 \) for all \( i \), \( b_i \geq 0 \) for all \( i \),
\( a - b = w \)

- For any \( a \geq 0, b \geq 0 \), there’s always a single \( w \) that satisfies the constraints.
- So the inner minimum is always attained at \( w = a - b \).
- Since \( w \) doesn’t show up in the objective function,
  - nothing changes if we drop \( \min_w \) and the constraint.
So lasso optimization problem is equivalent to

\[
\min_{a,b} \sum_{i=1}^{n} \left( (a - b)^T x_i - y_i \right)^2 + \lambda 1^T (a + b)
\]

subject to \( a_i \geq 0 \) for all \( i \), \( b_i \geq 0 \) for all \( i \),

where at the end we take \( w^* = a^* - b^* \) (and we’ve shown above that \( a^* \) and \( b^* \) are positive and negative parts of \( w^* \), respectively.)

Has constraints – how do we optimize?
Projected SGD

\[
\min_{w^+, w^- \in \mathbb{R}^d} \sum_{i=1}^n \left( (w^+ - w^-)^T x_i - y_i \right)^2 + \lambda_1^T (w^+ + w^-)
\]

subject to \( w^+_i \geq 0 \) for all \( i \)

\( w^-_i \geq 0 \) for all \( i \)

- Just like SGD, but after each step
  - Project \( w^+ \) and \( w^- \) into the constraint set.
  - In other words, if any component of \( w^+ \) or \( w^- \) becomes negative, set it back to 0.
Finding the Lasso Solution: Coordinate Descent (Shooting Method)
Coordinate Descent Method

- **Goal:** Minimize $L(w) = L(w_1, \ldots, w_d)$ over $w = (w_1, \ldots, w_d) \in \mathbb{R}^d$.
  - In gradient descent or SGD, each step potentially changes all entries of $w$.
  - In each step of **coordinate descent**, we adjust only a single $w_i$.

In each step, solve

$$w_i^{new} = \arg \min_{w_i} L(w_1, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots, w_d)$$

Solving this argmin may itself be an iterative process.

Coordinate descent is great when
  - it’s easy or easier to minimize w.r.t. one coordinate at a time
Coordinate Descent Method

Goal: Minimize \( L(w) = L(w_1, \ldots, w_d) \) over \( w = (w_1, \ldots, w_d) \in \mathbb{R}^d \).

- **Initialize** \( w^{(0)} = 0 \)
- **while** not converged:
  - Choose a coordinate \( j \in \{1, \ldots, d\} \)
  - \( w_j^{\text{new}} \leftarrow \arg\min_{w_j} L(w_1^{(t)}, \ldots, w_{j-1}^{(t)}, w_j, w_{j+1}^{(t)}, \ldots, w_d^{(t)}) \)
  - \( w_j^{(t+1)} \leftarrow w_j^{\text{new}} \) and \( w^{(t+1)} \leftarrow w^{(t)} \)
  - \( t \leftarrow t + 1 \)

- Random coordinate choice \( \Rightarrow \text{stochastic coordinate descent} \)
- Cyclic coordinate choice \( \Rightarrow \text{cyclic coordinate descent} \)

In general, we will adjust each coordinate several times.
Why mention coordinate descent for Lasso?

In Lasso, the coordinate minimization has a **closed form solution**!
Coordinate Descent Method for Lasso

Closed Form Coordinate Minimization for Lasso

\[ \hat{w}_j = \arg \min_{w_j \in \mathbb{R}} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda |w|_1 \]

Then

\[ \hat{w}_j = \begin{cases} 
(c_j + \lambda) / a_j & \text{if } c_j < -\lambda \\
0 & \text{if } c_j \in [-\lambda, \lambda] \\
(c_j - \lambda) / a_j & \text{if } c_j > \lambda 
\end{cases} \]

\[ a_j = 2 \sum_{i=1}^{n} x_{i,j}^2 \]

\[ c_j = 2 \sum_{i=1}^{n} x_{i,j} (y_i - w_{-j}^T x_{i,-j}) \]

where \( w_{-j} \) is \( w \) without component \( j \) and similarly for \( x_{i,-j} \).
Coordinate Descent: When does it work?

- Suppose we’re minimizing \( f : \mathbb{R}^d \rightarrow \mathbb{R} \).
- Sufficient conditions:
  1. \( f \) is continuously differentiable and
  2. \( f \) is strictly convex in each coordinate

- But lasso objective

\[
\sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_1
\]

is not differentiable...

- Luckily there are weaker conditions...
Coordinate Descent: The Separability Condition

Theorem

\(^a\) If the objective \( f \) has the following structure

\[
f (w_1, \ldots, w_d) = g (w_1, \ldots, w_d) + \sum_{j=1}^{d} h_j (x_j),
\]

where

- \( g : \mathbb{R}^d \rightarrow \mathbb{R} \) is differentiable and convex, and
- each \( h_j : \mathbb{R} \rightarrow \mathbb{R} \) is convex (but not necessarily differentiable)

then the coordinate descent algorithm converges to the global minimum.

\(^a\) Tseng 2001: “Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization”
Coordinate Descent Method – Variation

- Suppose there’s no closed form? (e.g. logistic regression)
- Do we really need to fully solve each inner minimization problem?
- A single projected gradient step is enough for $\ell_1$ regularization!
  - Shalev-Shwartz & Tewari’s “Stochastic Methods...” (2011)
Stochastic Coordinate Descent for Lasso – Variation

- Let \( \tilde{w} = (w^+, w^-) \in \mathbb{R}^{2d} \) and

\[
L(\tilde{w}) = \sum_{i=1}^{n} \left( (w^+ - w^-)^T x_i - y_i \right)^2 + \lambda (w^+ + w^-)
\]

Stochastic Coordinate Descent for Lasso - Variation

Goal: Minimize \( L(\tilde{w}) \) s.t. \( w_i^+, w_i^- \geq 0 \) for all \( i \).

- Initialize \( \tilde{w}^{(0)} = 0 \)
  - while not converged:
    - Randomly choose a coordinate \( j \in \{1, \ldots, 2d\} \)
    - \( \tilde{w}_j \leftarrow \tilde{w}_j + \max \{ -\tilde{w}_j, -\nabla_j L(\tilde{w}) \} \)