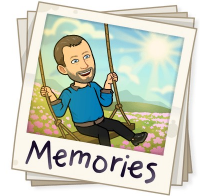


Machine Learning

Textbook: Machine Learning Refined, by Watt, Borhani, and Katsaggelos, 2nd edition:
MLRefined.com

Previous Lecture

- ◆ Centering changes the origin; standardization also rescales each feature by a chosen measure of spread.
- ◆ For an unregularized linear model, affine feature scaling can be absorbed into the coefficients without changing the representable predictions.
- ◆ Scaling can dramatically improve gradient-descent conditioning even when it does not change the underlying unregularized linear prediction problem.
- ◆ For distance-based methods such as k -means and PCA, scaling changes the geometry and can change the fitted model.
- ◆ Unordered categorical features require a representation such as one-hot encoding; with an intercept, redundant indicator columns create rank deficiency.
- ◆ All fitted preprocessing quantities must be estimated from training data only and then reused unchanged on validation and test data.



§9.7 - Regularization and Feature Selection

Last time:

feature scaling changed
the coordinate system of the model.

Now:

suppose a model has many coefficients
or several predictors tell us nearly the same thing.

A model may fit the training data well
while using unnecessarily large
or unstable coefficients.

Today:

add penalties to the objective
to control the fitted parameters.

This is **regularization**.

Ordinary Least Squares

For a linear model:

$$\hat{\mathbf{y}} = X\mathbf{w}.$$

Ordinary least squares minimizes

$$g_{\text{LS}}(\mathbf{w}) = \frac{1}{P} \|X\mathbf{w} - \mathbf{y}\|^2.$$

Only training fit appears in the objective.

If many different parameter vectors
fit nearly equally well,

least squares has no explicit preference
for a smaller or simpler coefficient vector.

Regularization adds such a preference.

Penalized Optimization

General form:

$$g_{\lambda}(\mathbf{w}) = \text{training loss} + \lambda (\text{penalty}).$$

Here

$$\lambda \geq 0$$

controls the strength of regularization.

If

$$\lambda = 0:$$

recover the unregularized model.

Larger

$$\lambda$$

places more weight on the penalty.

Thus fitting becomes a tradeoff:

fit the observed data

versus

keep the parameter vector controlled.

Usually Do Not Penalize the Intercept

Write

$$\mathbf{w} = \begin{bmatrix} w_0 \\ \mathbf{v} \end{bmatrix},$$

where

$$w_0$$

is the intercept

and

$$\mathbf{v}$$

contains feature coefficients.

A common convention:

penalize

$$\mathbf{v}$$

but not

$$w_0.$$

Why?

The intercept determines the baseline level.

Penalizing it can make the result depend unnecessarily on where the response and predictors are centered.

Most formulas below focus on the penalized coefficient vector $\mathbf{v}$.

L_2 Regularization, Ridge

The ridge objective is

$$g_{\text{ridge}} = \frac{1}{P} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{v}\|_2^2.$$

Since

$$\|\mathbf{v}\|_2^2 = \sum_{j=1}^N v_j^2.$$

large coefficient magnitudes receive a quadratic penalty.

Ridge regression usually **shrinks** coefficients toward zero.

For finite λ , it does not generally force them exactly to zero.

A One-Dimensional Ridge Problem

Consider

$$g(w) = (w - a)^2 + \lambda w^2.$$

Without regularization:

$$w^* = a.$$

Differentiate:

$$\begin{aligned} g'(w) &= 2(w - a) + 2\lambda w. \end{aligned}$$

Set equal to zero:

$$(1 + \lambda)w = a.$$

Therefore

$$w_{\text{ridge}}^* = \frac{a}{1 + \lambda}.$$

Ridge multiplies the unregularized solution by the shrinkage factor

$$\frac{1}{1+\lambda}.$$

Ridge Shrinkage Example

Let

$$a = 3.$$

Then

λ	0	0.5	1	2	9
w^*	3	2	1.5	1	0.3

As

$$\lambda$$

increases:

$$|w^*|$$

decreases.

But for

$$a \neq 0$$

and finite

$$\lambda:$$

$$w^* \neq 0.$$

This continuous shrinkage

is characteristic of ridge regression.

Ridge Normal Equations

First suppose every coefficient is penalized.

Objective:

$$g(\mathbf{w}) = \frac{1}{P} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2.$$

Gradient:

$$\begin{aligned} \nabla g \\ = \frac{2}{P} X^T(X\mathbf{w} - \mathbf{y}) + 2\lambda \mathbf{w}. \end{aligned}$$

Set equal to zero:

$$X^T X \mathbf{w} - X^T \mathbf{y} + P \lambda \mathbf{w} = \mathbf{0}.$$

Therefore

$$(X^T X + P \lambda I) \mathbf{w} = X^T \mathbf{y}.$$

Leave the Intercept Unpenalized

If the first coefficient is an intercept,

define

$$R = \text{diag}(0, 1, \dots, 1).$$

Then the penalty is $\mathbf{w}^T R \mathbf{w}$.

Ridge normal equations become $(X^T X + P \lambda R) \mathbf{w} = X^T \mathbf{y}$.

Again:
do not numerically form an explicit matrix inverse.

Solve the linear system.

Ridge Can Stabilize a Nearly Singular Problem

Suppose $X^T X$
has eigenpair $(\eta, \mathbf{q})$.

Then $(X^T X + P \lambda I) \mathbf{q}$
 $= (\eta + P \lambda) \mathbf{q}$.

Thus every eigenvalue is shifted upward:
 $\eta \rightarrow \eta + P \lambda$.

Very small curvature directions
become less singular.

This is one reason ridge regression
can stabilize models with
highly correlated predictors.

A Two-Coefficient Ridge Example

Take $X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$,

and $\mathbf{y} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$.

Here $P = 3$.

Compute

$$X^T X = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

and

$$X^T \mathbf{y} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}.$$

Ordinary least squares gives

$$\mathbf{w}_{\text{LS}} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Add Ridge Regularization

Choose

$$\lambda = \frac{1}{3}.$$

Then

$$P\lambda = 1.$$

Solve

$$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \mathbf{w} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}.$$

The solution is

$$\begin{aligned} \mathbf{w}_{\text{ridge}} &= \begin{bmatrix} 7/8 \\ 11/8 \end{bmatrix} \\ &= \begin{bmatrix} 0.875 \\ 1.375 \end{bmatrix}. \end{aligned}$$

Both coefficient magnitudes
are smaller than the ordinary least-squares coefficients.

The Ridge Solution Fits Training Data Less Exactly

Ordinary least squares predicts

(1,2,3)
exactly.

Training SSE:

$$0.$$

Ridge predicts

$$\begin{aligned} &\left(\frac{7}{8}, \frac{11}{8}, \frac{18}{8}\right) \\ &= (0.875, 1.375, 2.25). \end{aligned}$$

Residuals:

$$-0.125, \quad -0.625, \quad -0.75.$$

Ridge SSE:

$$\frac{1}{64} + \frac{25}{64} + \frac{36}{64}$$
$$= \frac{62}{64} = 0.96875.$$

Regularization deliberately accepts
some training error
in exchange for smaller coefficients.

Why Would Worse Training Fit Help?

The goal is not
to minimize training error at any cost.

The goal is
useful behavior on future observations.

Large coefficients can make predictions
sensitive to small changes in inputs
and to sampling noise.

Regularization can reduce this sensitivity.

This introduces the central tradeoff:

training fit
versus
model complexity / stability.

We will study this tradeoff explicitly
when we discuss generalization.

L_1 Regularization, Lasso

The lasso objective replaces squared coefficient size
with absolute coefficient size:

$$g_{\text{lasso}} = \frac{1}{P} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{v}\|_1.$$

Where

$$\|\mathbf{v}\|_1 = \sum_{j=1}^N |v_j|.$$

Like ridge:
lasso shrinks coefficients.

Unlike ridge:
lasso can produce
 $v_j = 0$
exactly.

That creates a direct connection
to feature selection.

A One-Dimensional Lasso Problem

Consider

$$g(w) = (w - a)^2 + \lambda|w|.$$

The absolute value creates three cases:

$w > 0;$

$w < 0;$

$w = 0.$

Away from zero
the derivative is ordinary.

At zero
the objective is not differentiable.

This corner
is exactly what allows
a coefficient to stop at zero.

Case 1: $w > 0$

If

$w > 0:$
 $|w| = w.$

Then

$$\begin{aligned} g'(w) \\ = 2(w - a) + \lambda. \end{aligned}$$

Set equal to zero:

$$w = a - \frac{\lambda}{2}.$$

This solution is valid only when
 $a - \frac{\lambda}{2} > 0.$

Thus:

$$a > \frac{\lambda}{2} \quad \Rightarrow \quad w^* = a - \frac{\lambda}{2}.$$

Case 2: $w < 0$

If

$w < 0:$
 $|w| = -w.$

Then

$$g'(w)$$

$$= 2(w - a) - \lambda.$$

Set equal to zero:

$$w = a + \frac{\lambda}{2}.$$

This is negative only when

$$a < -\frac{\lambda}{2}.$$

Therefore:

$$a < -\frac{\lambda}{2} \implies w^* = a + \frac{\lambda}{2}.$$

Case 3: the Solution Is Exactly Zero

If

$$|a| \leq \frac{\lambda}{2},$$

neither nonzero stationary point
lies in its assumed region.

Then the minimizer is the corner:

$$w^* = 0.$$

Combine all three cases:

$$w^* = \text{sign}(a) \max\left(|a| - \frac{\lambda}{2}, 0\right).$$

This is the

soft-thresholding operator.

Ridge Versus Lasso in One Dimension

For

$$(w - a)^2$$

plus a penalty:

Penalty	Minimizer
λw^2	$\frac{a}{1+\lambda}$
$\lambda w $	$\text{sign}(a) \max\left(a - \frac{\lambda}{2}, 0\right)$

Ridge:

smooth shrinkage.

Lasso:

shrinkage plus thresholding.

If the unregularized signal is small enough,

lasso removes it completely.

A Numerical Lasso Example

Let
 $a = 3.$

Then

λ	0	2	4	6	8
w^*	3	2	1	0	0

At
 $\lambda = 6:$
 $\lambda/2 = 3.$

The coefficient reaches
0.

Increasing the penalty further
keeps it at zero.

This is sparsity.

Subgradients Explain the Zero

For
 $|w|:$
 $\partial|w| = \begin{cases} \{1\}, & w > 0, \\ \{-1\}, & w < 0. \end{cases}$
 $w = 0, 1],$

At
 $w = 0,$

the lasso stationarity condition is
 $0 \in -2a + \lambda[-1, 1].$

This is possible exactly when
 $|2a| \leq \lambda,$

or
 $|a| \leq \lambda/2.$

The interval of subgradients at the corner
allows exact zeros.

Penalty Form Versus Constraint Form

Ridge may also be written conceptually as:

minimize training loss
subject to
 $\|\mathbf{v}\|_2^2 \leq t.$

Lasso may be viewed as:

minimize training loss
subject to
 $\|\mathbf{v}\|_1 \leq t.$

For suitable relationships between
 t
and
 $\lambda,$

the constrained and penalized forms
describe the same tradeoff.

This gives a geometric interpretation.

Why L_1 Geometry Encourages Zeros

In two dimensions:

$\|\mathbf{v}\|_2 \leq t$
is a disk.

Its boundary is smooth.

But
 $\|\mathbf{v}\|_1 \leq t$
is a diamond:
 $|v_1|+|v_2| \leq t.$

The diamond has corners
on the coordinate axes.

As elliptical loss contours expand
until they first touch the feasible set,

a corner is comparatively likely
to be the contact point.

At a corner:
at least one coefficient is zero.

This geometric picture matches the soft-threshold formula.

Feature Selection

Suppose
 $v_j = 0.$

Then feature j
does not contribute to the linear prediction:
 $v_j x_j = 0.$

So exact zero coefficients
effectively remove features from the model.

This motivates using lasso
as a form of
embedded feature selection.

The selection occurs
while fitting the predictive model,
not in a separate preprocessing step.

Subset Selection Is a Discrete Alternative

One could instead ask:
which subset of the
 N
features should be included?

There are
 2^N
possible subsets.

For
 $N = 30$:
 $2^{30} = 1,073,741,824$.

Exhaustive search quickly becomes expensive.

Lasso replaces this combinatorial search
with a convex penalty
that can create sparse coefficient vectors.

It is not identical to best-subset selection,
but it provides a computationally attractive alternative.

Feature Scaling Is Essential When Coefficients Are Penalized

Suppose prediction uses
 $w x$.

Replace
 x
by
 $z = 100x$.

To preserve the same predictions:
 $\tilde{w} = \frac{w}{100}$.

But the ridge penalty changes from
 w^2

to

$$\tilde{w}^2 = \frac{w^2}{10000}.$$

The same prediction function
receives a very different numerical penalty.

Therefore:
regularization is not invariant to arbitrary feature scaling .

Why Standardization Is Common Before Ridge or Lasso

If predictors are placed on comparable scales:
a unit coefficient magnitude
has a more comparable interpretation across features.

Then
 $\sum_j v_j^2$
or
 $\sum_j |v_j|$
penalizes coefficients more evenly.

Without scaling:
features expressed in large numerical units
may require tiny coefficients
and therefore receive little penalty.

Features expressed in small numerical units
may require large coefficients
and receive much larger penalty.

This would make regularization depend
on arbitrary measurement units.

Correlated Predictors: Ridge and Lasso Behave Differently

Suppose two features
contain nearly the same information.

Ridge often:

keeps both;

shares coefficient weight between them;

stabilizes their combined effect.

Lasso may:

choose one;

set another to zero.

But which correlated feature is selected
can be unstable across samples.

Therefore:
a nonzero lasso coefficient
should not automatically be interpreted
as uniquely scientifically important.

Regularization Is Not Causal Feature Discovery

A selected feature can be:
correlated with a true driver;

a proxy for another variable;

favored because of scaling or sampling variation.

A coefficient equal to zero
does not prove a feature has no real-world relationship.

A nonzero coefficient
does not prove a causal effect.

Regularization solves
a predictive optimization problem.

Scientific interpretation requires
additional assumptions and evidence.

What Does λ Control?

Small
 λ :

weak penalty;

closer fit to training data;

larger coefficient freedom.

Large
 λ :

stronger shrinkage;

simpler / more constrained coefficient vector;

potentially larger training error.

Extremely large

λ :
coefficients may be driven close to zero
or exactly zero under lasso.

Which value should we choose?

Not from training loss alone.

Why Training Loss Cannot Choose λ

As regularization increases,
training fit is constrained more strongly.

Therefore the unpenalized training loss
generally cannot improve
by imposing a stronger constraint.

If we chose
 λ
only to minimize training error:
we would usually prefer
 $\lambda = 0$.

The purpose of regularization is
improved behavior beyond the training sample.

So
 λ
must be treated as a
hyperparameter
evaluated using held-out data.

Cross-validation will provide a systematic method later.

Ridge Remains a Convex Problem

Least-squares loss is convex.

The ridge penalty
 $\lambda \|\mathbf{v}\|_2^2$
is convex.

Therefore their sum is convex.

For
 $\lambda > 0$
with all non-intercept directions penalized,

the penalized coefficient block receives
additional positive curvature.

Thus ridge is usually easier numerically

than an unstable unregularized least-squares problem.

There are no spurious local minima.

Lasso Is Convex but Nonsmooth

Least-squares loss is convex.

The function

$$|w|$$

is convex.

Therefore

$$\|\mathbf{v}\|_1$$

is convex.

So lasso is also a convex optimization problem.

But:

$$|w|$$

is not differentiable at

$$w = 0.$$

Thus ordinary smooth-gradient formulas
are not enough by themselves.

Common solution methods include:

coordinate descent;

proximal-gradient methods;

specialized convex solvers.

The soft-thresholding operator

is central to several of these algorithms.

A Bridge to Elastic Net

Sometimes we combine the two penalties:

$$\lambda_1 \|\mathbf{v}\|_1 + \lambda_2 \|\mathbf{v}\|_2^2.$$

This is the basic idea of

elastic-net regularization.

It can combine:

lasso-like sparsity

with

ridge-like stabilization of correlated predictors.

We will not develop it further here.

The main conceptual distinction remains:

- L_2 :
smooth shrinkage;
- L_1 :
shrinkage plus possible exact zeros.

Ridge Regression in Python

For a design matrix without an intercept:

```
import numpy as np

X = np.array([[1., 0.],
              [0., 1.],
              [1., 1.]])

y = np.array([1., 2., 3.])

lam = 1/3

P = len(y)

A = X.T @ X + P*lam*np.eye(X.shape[1])

b = X.T @ y

w_ridge = np.linalg.solve(A, b)
```

Result:
[0.875 1.375]

As before:
solve the system;
do not compute an explicit inverse.

Leave the Intercept Unpenalized in Python

If the first column of

```
X
is the intercept:
R = np.eye(X.shape[1])
R[0, 0] = 0.0

A = X.T @ X + P*lam*R

w_ridge = np.linalg.solve(
    A,
    X.T @ y
)
```

The matrix
 R
selects which coefficients
receive the penalty.

This matches

$$(X^T X + P \lambda R) \mathbf{w} = X^T \mathbf{y}.$$

Soft Thresholding in Python

Implement

```
S_tau(a) = sign(a) max(|a|-tau, 0).
def soft_threshold(a, tau):
    return np.sign(a) * np.maximum(
        np.abs(a) - tau,
        0
    )
```

For our lasso example:

```
a = 3.0

for lam in [0, 2, 4, 6, 8]:
    w = soft_threshold(
        a,
        lam/2
    )

    print(lam, w)
```

Output:

```
0 3.0
2 2.0
4 1.0
6 0.0
8 0.0
```

This directly reproduces
the analytic solution.

Coefficient Paths

A useful diagnostic is to fit
a sequence of
 λ
values
and record the coefficients.

For ridge:
coefficients usually move smoothly toward zero.

For lasso:
coefficients can hit zero
and remain there.

A plot of
coefficient value versus regularization strength
is called a
regularization path
or

coefficient path.

It reveals how strongly each feature survives increasing regularization.

A Practical Regularization Checklist

Before using coefficient penalties:

decide which coefficients should be penalized;

place numeric predictors on defensible comparable scales;

keep preprocessing inside the training procedure;

distinguish ridge shrinkage from lasso sparsity;

do not interpret selected variables causally by default;

choose

λ

using validation / cross-validation,
not training error alone.

Regularization changes the learning objective.

Its consequences are mathematical,
computational,
and statistical.

What did we learn?

- ◆ Regularization adds a coefficient penalty to the training objective and trades training fit against parameter control.
- ◆ Ridge uses an L_2 penalty and shrinks coefficients smoothly; its normal equations add $P\lambda I$ to the least-squares curvature matrix.
- ◆ Lasso uses an L_1 penalty; its nonsmooth corner at zero allows coefficients to become exactly zero.
- ◆ In one dimension, lasso produces the soft-threshold rule $\text{sign}(a) \max(|a| - \lambda/2, 0)$.
- ◆ Exact zero lasso coefficients provide embedded feature selection, but selected features need not be uniquely important or causal.
- ◆ Feature scaling changes coefficient penalties, so scaling and regularization must be designed together.



Next

- Linear models remain linear in the original features.
- Regularization controls coefficient use, but it does not create curved relationships by itself.
- Next we enlarge the feature representation with polynomial, interaction, and other nonlinear transformations

while keeping the final model linear in its parameters.

- **1.28 - Nonlinear Feature Maps**