Learning Reflection 2

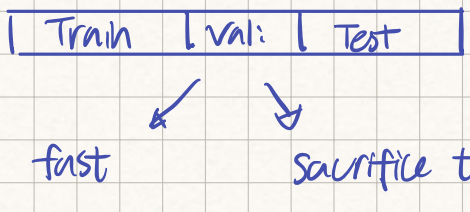
2022/4/4 (Mon.)

Cross validation

choosing complexity → need lower train error, lower test error.

2 ways to pick the model complexity without ruining test set:

1. validation set
2. Cross validation



Cross validation:

use small validation sets w/o losing too much data.

slow, usually use K=5 to 10.

```
train, validation, test = split_data(dataset)
for each model complexity p:
    model = train_model(model_p, train)
    val_err = error(model, validation)
    Keep track of p with smallest val_err
return best p + error(model, test).
```

```
chunk1, ..., chunkK, test = split_data(dataset)
for each model complexity p:
    for i in [1, K]:
        model = train_model(model_p, chunks-i)
        val_err = error(model, chunk=i)
    avg_val_err = average val_err over chunks.
    keep track of p with smallest avg_val_err
return model trained on train with best
    p + error(model, test)
```

Coefficients and overfitting

Overfitting - large estimated parameters $\hat{w}$, low bias, high variance.

Underfitting - small parameters $\hat{w}$, high bias, low variance.

※ Overfitting is not limited to polynomial regression of large degree.
It can also happen if you use a large # of features.

※ prevent overfitting. Slide 17.

Regularization.

quality metric (minimized loss): $\hat{w} = \min_w L(w)$

$\hat{w} = \min_w \underbrace{L(w)}_{\text{measure of fit}} + \eta \underbrace{R(w)}_{\text{measure the magnitude of coefficients.}}$

Say, $w = [w_0, w_1, \ldots, w_D]$,

Sum: $R(w) = \sum_{j=0}^{D} w_j$ Doesn't work, $w = [10000, -10000]$ $R(w) = 0$

Sum of abs.: $R(w) = \sum_{j=0}^{D} |w_j| \triangleq \|w\|_1$ L1 Norm Lasso

sum of sqrs: $R(w) = \sum_{j=0}^{D} w_j^2 \triangleq \|w\|_2^2$ L2 Norm Ridge.

Ridge Regression (L2 Norm)

minimize $\hat{w} = \min_w RSS(w) + \eta \|w\|_2^2$

η is a tuning parameter that changes how much the model cares about the regularization term.

$\eta = 0$, $\hat{w} = \min_w RSS(w) = \hat{w}_{OLS}$ (OLS) ordinary least squares.

$\eta = \infty$, $\hat{w} = \vec{0}$

If any $w_j \neq 0$, $RSS(w) + \eta \|w\|_2^2 = \infty$

If all $w_j = 0$, $RSS(w) + \eta \|w\|_2^2 = RSS(w) < \infty$

η in between

$0 \leq \|w\|_2^2 \leq \|\hat{w}_{OLS}\|_2^2$

Complex model.

$\eta = 0$ → model have many parameters, low bias, high variance

$\eta = \infty$ → model have few parameters, low variance, high bias.

Simple model.

Code to train a Ridge model.

```
from sklearn.linear_model import Ridge.
model = Ridge(alpha=1.5)
```

Intercept.

Two ways dealing with intercept issues.

1. Change the measure of overfitting to not including the intercept.

$$\min_{w_0, w_{rest}} RSS(w_0, w_{rest}) + \eta \|w_{rest}\|_2^2$$

2. Center the y values so they have mean 0.

Scaling features.

Scale of coefficients.

$$x = 100\ m^2 = \frac{100}{(1000)^2}\ km^2$$

This makes the input value to be smaller, and the $\hat{w}_i$ will be larger.

This accidentaly penalize features for having large coefficients due to having small value inputs.

Fix this by normalizing the feature $\tilde{h}_j(x_i) = \frac{h_j(x_i) - \mu_j(x_1, \ldots, x_N)}{\sigma_j(x_1, \ldots, x_N)}$

where the mean of feature j: $\mu_j(x_1, \ldots, x_N) = \frac{1}{N}\sum_{i=1}^{N} h_j(x_i)$

standard deviation of feature j: $\sigma_j(x_1, \ldots, x_N) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(h_j(x_i) - \mu_j(x_1, \ldots, x_N))^2}$

Must Scale the Test data & all future data using the means & std. of training set.

2022, 4.6(三)

☆ Overfitting: { polynomial regression of large degree
large number of features.

☆ How to choose the best value of λ?

After we train each model with a certain λ_i in order to find

$\hat{w}_i = \text{argmin}_w \text{MSE}(w) + \lambda_i \|w\|_2^2$

→ Pick the λ_i that has the smallest $\text{MSE}(\hat{w}_i)$ on the validation set.

We want to choose the λ that will do best on future data.

= Need to minimize validation error.

* for λ in λs:

Train a model using Gradient Descent

$$\hat{w}_{ridge}(\lambda) = \min_w \text{MSE}_{train}(w) + \lambda \|w_{1:D}\|_2^2$$

Compute validation Error.

validation error = $\text{MSE}_{val}(\hat{w}_{ridge}(\lambda))$

Track λ with smallest validation error.

Return λ^* + estimated future error $\text{RSS}_{test}(\hat{w}_{ridge}(\lambda))$

☆ Feature selection

① Complexity, adjust to appropriate

② Interpretability

③ Efficiency

If $\hat{w}$ is sparse, only need to look at the non-zero coefficients.

$$y = \sum_{\hat{w}_j \neq 0} \hat{w}_j h_j(x)$$

✡ How many linear regression models in total do we have to evaluate for a dataset with d features in order to find the global optimum?

$\boxed{2^d}$

✡ Greedy Algo.

- Forward stepwise
- Backward stepwise
- Combining.

Runtime: $O(K^2)$

min_val = ∞

for i ← 1 ··· K:

Find feature f_i not in S_{i-1}, that when we combined w/ S_{i-1}, minimizes the validation loss the most.

$S_i \leftarrow S_{i-1} \cup \{f_i\}$

if val_loss(S_i) > min_val:

break.

✡ Sparsity

for regression, means many of the learned coefficients are 0.

✡ Hyper-parameter

a parameter you specify for the model that influences which param. (e.g. coefficients) are learned by ML. alg.

✡ LASSO

adds bias to the Least Squares solution (avoid variance → overfitting)

If want to remove bias from LASSO soln.:

1. Run LASSO to select features
2. Run regular OLS on dataset with only those features.

LASSO select highly correlated features arbitrarily amongst them.

→ ElasticNet: $\hat{w}_{elastic} = \min_w RSS(w) + \eta_1 \|w\|_1 + \eta_2 \|w\|_2^2$

Summary:

1. For feature selection,
   adjusting feature size to pick the best model having lowest $MSE_{val}(\hat{w})$.
2. Three options choosing feature numbers:
   a validation set; cross validation; Bayesian Information
   other metrics for penalizing complexity like BIC. Criterion.
3. L1 penalty causes more sparse than L2 penalty.
4. Lasso → feature selection.
5. L1 (LASSO)
   more sparsity
   feature selection.

   L2 (RIDGE)
   Makes weights small (not 0)
   More sensitive to outliers (due to square)
   usual works better in practice

Uncertainties:

1. If Elastic Net allows us to combine L1 & L2 together to do the regularization, why not using elastic net instead of using L1 or L2 separately?