Regularization Note

Lizi Chen

1 Background

Regularization is a method to resolve data over-fitting issue when **'training'** a proper statistical model for a given dataset.

1.1 Basics: (To look for the best fitting/estimation function!)

Considering **Empirical Risk Minimization**[1]:

- The hypothesis space of functions $\mathcal{F}$ (means all the possible fitting functions.)
- The complexity measure function $\Omega : \mathcal{F} \to [0, \infty)$.

1.1.1 Loss, True Risk, Empirical Risk Minimization

In the supervised learning set-up, given a set of data X, Y and a model, the **loss** is defined as the difference between he predicted value $\hat{Y}$ and he real value Y. The definition of **true risk** computes the average loss over all possibilities; however, all possibilities can never be known.

Therefore, we select a training set S from all the real data $\mathbf{R}$ which has unknown distribution $\mathcal{D}$ and labeled by some target function f. The goal in this set-up is to find a function (predictor, regressor, classifier) that minimizes the error with respect to the unknown $\mathcal{D}$ and f. The error here is called 'empirical risk'. It's *empirical* (not *true*), because we are using a dataset that's a subset of the whole real population. Thus, the process of minimizing the empirical risk is called 'Empirical Risk Minimization' (ERM).

1.1.2 L1-norm and L2-norm loss function

While L1-norm and L2-norm can be used for regularization, it can be used as loss function to calculate the difference between target value (Y) and predicted/estimated value ($\hat{Y}$).
L1-norm, Least Absolute Deviations (LAD):

$$L_{l1} = \sum_{i=1}^{N} |y_i - h_\theta(x_i)|$$

L2-norm, Least Squares:

$$L_{l2} = \sum_{i=1}^{N} (y_i - h_\theta(x_i))^2$$

L1 is robust; resistant to outliers, and produce sparse output(s). L2 is less robust, but stable, and always produce one solution. L1, due to its tendency to produces sparse coefficients, has built-in feature selection function can be leveraged.

[1] Read *Understanding Machine Learning: From Theory to Algorithms*, by Shai Ben-David and Shai Shalev-Shwartz

1.1.3 Constrained ERM (Empirical Risk Minimization), Ivanov Regularization:

We consider all functions in $\mathcal{F}$ with complexity **at most** r, where r refers to the increasing complexities as $r = 0, 1.2, 2.6, 5.4, ...$ We can say that the hypothesis space of fitting functions are nested as:

$$\mathcal{F}_0 \in \mathcal{F}_{1.2} \in \mathcal{F}_{2.6} \in \mathcal{F}_{5.4} \in ... \in \mathcal{F}$$

For a fixed $r \geq 0$, we want to minimize the loss:

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i), \quad s.t.\ \Omega(f) \leq r$$

1.1.4 Penalized ERM, Tikhonov Regularization

For a fixed $\lambda \geq 0$, we want to minimize:

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

(Tikhonov regularization is convenient, because it is unconstrained minimization.)

1.1.5 Example:

l_1 regularization: Consider linear models $\mathcal{F} = \{f : \mathbf{R}^d \to \mathbf{R} | f(x) = w^T x \ \text{ for } w \in \mathbf{R}^d\}$, the Linear Least Square Regression is ERM for l over $\mathcal{F}$:

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

PROBLEM!!

overfit when d is large compared to n

Note: $d \gg n$ is very common in NLP problems, i.e., 1M features vs 10K documents.

Add Proof? todo

1.2 What's overfit?

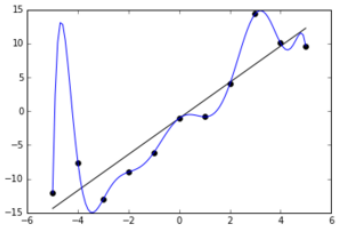


Figure 1: Over-fitting

模型非常完美的拟合上现有训练数据，导致该模型方程是一个过分复杂的模型。设 Fig.1 原本数据为 Linear, $(f(x) = b_0 + b_1 \cdot x)$, 但 overfitting 模型讲其过分拟合为方程 $f_{overfit}(x) = b_0 + b_1 \cdot x + b_2 \cdot x^2 + ... + b_n \cdot x^n$ 的复杂非线性方程, $f_{overfit}$ 在 training dataset 给予最低的误差，却无法保证最好的整体预测能力。

Overfitting can be detected when we reserve a portion of the training data as test dataset. (i.e., 80% training data, 20% testing data.) If the precision and recall for training dataset is relatively greater than the testing dataset. It's a red flag. Also, In Occam's Razor test, we always choose the simpler model from more complicated models that have comparable performance.

1.3 Bias vs. Variance Tradeoff

Bias occurs when an algorithm has limited flexibility to learn the true signal from a dataset. **Variance** refers to an algorithm's sensitivity to specific sets of training data.

Figure 2: Over-fitting

High bias, low variance → model is consistent, but inaccurate on average.
High variance, low bias → model is inconsistent, but accurate on average.

SOLUTION: **Regularization!**

2 Regularization

2.1 What is Regularization?

In order to suppress the tendency of over-fitting; or more concretely to say, to suppress the $b_2, b_3, ...b_n$ weight parameters for x^i (, where $i \geq 2$), we introduce **Regularization**, to reduce model complexity.

To simplify: Regularization is nothing but adding a penalty term to the objective function and control the model complexity using that penalty term.

2.2 Ridge Regression:

Regression with l_2 regularization is the same as Ridge Regression. Ridge Regression should probably be called Tikhonov Regularization, since Tikhonov has the earliest claim on this method.

For the Linear Least Square Regression model (from previous example) in the Penalized ERM (Tikhonov) form, we have:

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

, where $||w||^2 = w_z^2 + .. + w_d^2$ is the square of the l_2 norm.

Similarly, for the Constrained ERM (Ivanov) form, we have:

$$\hat{w} = \underset{||w||_2^2 \leq r^2}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \{w^T x_i - y_i\}, \text{ where the complexity parameter } r \geq 0. \tag{2}$$

Here, the w is suppressed.

Add Lagrangian duality, KKT intro, etc.

Ref. SVM doc

Ref 1 : David Rosenberg: Lagrangian Duality in Ten minutes?
Ref 2 : David Rosenberg: Lagrangian Duality and Convex Optimization
Ref 3 : Khan Academy: Applications of Multi-variable Derivatives
Ref 4 : 马同学: 如何理解拉格朗日乘子法?

We use KKT method to find solution:

$$\arg\min_{w} \max_{\lambda,\ \lambda\geq 0} L(w,\lambda) = \arg\min_{w} \max_{\lambda,\ \lambda\geq 0} \frac{1}{n}\sum_{i=1}^{n}\{w^T x_i - y_i\} + \lambda(w^T S w - \mathbf{R}) \tag{3}$$

, where $S = diag([0,1,\cdots,1]^T) \in \mathbf{R}^{(d+1)\times(d+1)}$

Listing 1: "Pseudo-code solve Lagrangian form of constrained regression formula."

```
input:
   w^(0) ∈ R^d is an initial guess.
   λ^(0) = 0.
   δ > 0, increment amount for λ.
repeat:
      Solve w^(t+1) = arg min_w L(w; λ^(t)) using some iterative algorithm;
      starting at w^(t)
      if w^(t+1)^T w^(t+1) > R then:
            λ^(t+1) = λ^(t) + δ in order to increase L(λ; w^(w+1);
      end
until w^(t+1)^T w^(t+1) ≤ R
```

How does Regularization work?

2.3 How does λ and r induce regularity? - Lipschitz Continuous (函数上任意两点连线的斜率有界。)

Formally, for $\widehat{\mathbf{f}}(x) = \hat{w}^T x$, if $\widehat{\mathbf{f}}$ is **Lipschitz continuous** with Lipschitz constant $L = ||\hat{w}||_2$. The function $\widehat{\mathbf{f}}$ never change with rate larger than L. When moving from x to $x+h$, $\widehat{\mathbf{f}}$ changes no more than $L||h||$:

$$\begin{aligned} |\hat{f}(x+h) - \hat{f}(x)| &= |\hat{w}^T(x+h) - \hat{w}^T x| \\ &= |\hat{w}^T h| \\ &\leq ||\hat{w}||_2\ ||h||_2 \quad \text{(Cauchy-Schwarz Inequality)} \end{aligned} \tag{4}$$

2.3.1 Proof of Cauchy-Schwarz Inequality:

Vector form:

$$||x|| \cdot ||y|| \geq |(x,y)|$$

2D Form:

$$(a^2+b^2)(c^2+d^2) \geq (ac+bd)^2 \ \equiv\ ac+bd \leq \sqrt{(a^2+b^2)(c^2+d^2)}$$

General 2D Form:

$$\sum_{i=1}^{n} a_i^2 \sum_{i=1}^{n} b_i^2 \geq \Big(\sum_{i=1}^{n} a_i b_i\Big)^2$$

2.4 LASSO:

(Least Absolute Shrinkage and Selection Operator). Regression with l_1 regularization is the same as LASSO.

LASSO Regression, Tikhonov Form:

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

, where$||w||_1 = |w_1| + \cdots + |w_d|$ is the l_1-norm. $\lambda \geq 0$ is the regularization parameter.

LASSO Regression, Ivanov Form:

$$\hat{w} = \arg\min_{||w||_1 \leq r} \frac{1}{n} \sum_{i=1}^{n} \{w^T x_i - y_i\}^2$$

$r \geq 0$ is the complexity parameter.

2.4.1 Group LASSO

Modify the objective function so that the coefficients of the model are split into sets. For example, to split the coefficients into m sets:

$$\min_w \frac{1}{2} \underbrace{\left|\left| y - \sum_{l=1}^{m} X^{(l)} w^{(l)} \right|\right|_2^2}_{\text{OLS}} + \lambda \underbrace{\sum_{l=1}^{m} \sqrt{p_l} ||w^{(l)}||_2}_{\text{Sum of L2 norms}}$$

, where λ controls the overall penalty and $\sqrt{p_l}$ is the weighted penalty for each set of coefficients.

Example: There are 10 coefficients: $w_1, w_2, \cdots, w_{10}$, into 2 sets: $\{w_1, w_2, .., w_5\}$, and $\{w_6, w_7, .., w_{10}\}$. The group lasso penalty (Sum of coefficient L2 norm) is:

$$\lambda \cdot \left(\sqrt{p_1} \sqrt{\sum_{i=1}^{5} w_i^2} + \sqrt{p_2} \sqrt{\sum_{i=6}^{10} w_i^2} \right)$$

2.4.2 Sparse Group Lasso, Simon et al 2013

Introduces α to control over L1 norm and sum of L2 norm:

$$\frac{1}{2n} \left|\left| y - \sum_{l=1}^{m} X^{(l)} w^{(l)} \right|\right|_2^2 + (1-\alpha) \cdot \lambda \sum_{l=1}^{m} \sqrt{p_l} ||w_{(l)}||_2 + \alpha \cdot \lambda ||w||_1$$

While Group Lasso method 'eliminate' coefficient to 0 by groups, Sparse Group Lasso method can also 'eliminate' single coefficient in a coefficient group.

2.5 Elastic Nets

Elastic Nets have proved to be (in theory and in practice) better than L1/LASSO. Elastic Nets combine L1 and L2 regularization at the 'only' cost of introducing another hyper-parameter to tune. (See details at: Regularization and variable selection via the elastic net, Hui Zou, Trevor Hastie, 2005, http://web.stanford.edu/~hastie/Papers/B67.2%20%282005%29%20301-320%20Zou%20&%20Hastie.pdf)

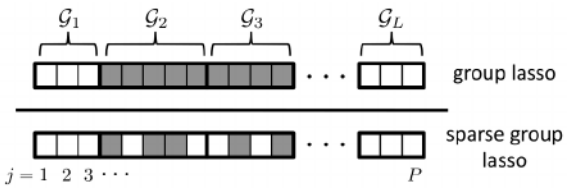
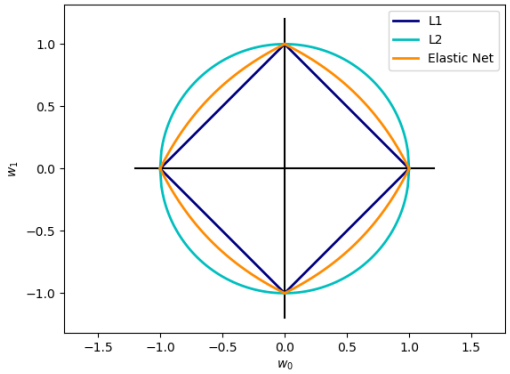


Figure 3: Group Lasso vs. Sparse Group Lasso.



Graph resource: http://scikit-learn.org/stable/modules/sgd.html#mathematical-formulation

3 Discussion:

3.1 Regularization Path

From 2.2(Ridge Regression) and 2.4(LASSO), the Ivanov Form of each gives $\hat{w}$ for the best parameters ($\hat{w}_r$) as result of $argmin$, given r^2 or r is the constraint. Now assume $\hat{w} = \hat{w}_\infty$ is the Unconstrained ERM, meaning it's the best possible yet unknown parameters for the regression model, the following pictures shows the relations between $\hat{w}_r$ and $\hat{w}_\infty$:

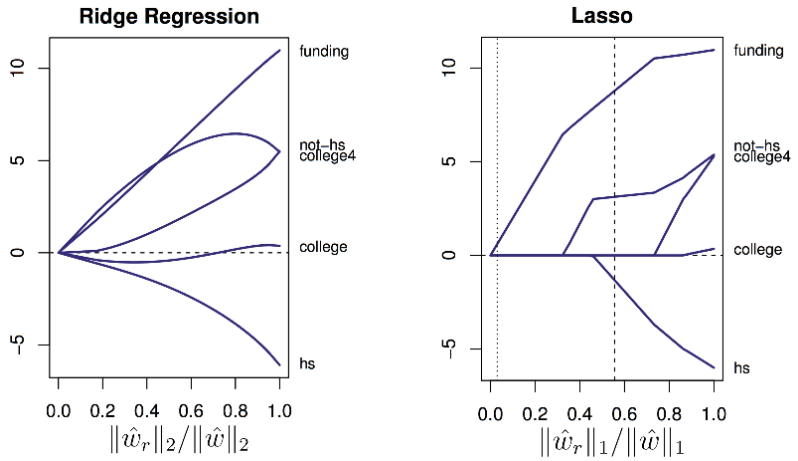


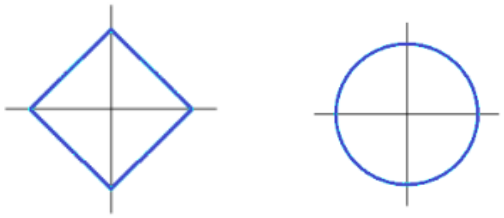
Figure 4: When $r = 0, ||\hat{w}_r||/||\hat{w}_\infty = 0$, when $r = \infty, ||\hat{w}_r||/||\hat{w}_\infty = 1$.

We can tell that LASSO tends to give sparse result as at certain point of r, more 0's appears in the model's coefficient. What's so good about 0's in model coefficient:

1. Efficient in computation and memory storage.
2. Easier to identify more important features.
3. Better prediction.
4. As a feature-selection step for training a slower non-linear model.

3.2 Sparsity: Contours of l_1 and l_2 Regularization for Visualization

For visualization purpose, restrict to 2D input space. $\mathcal{F} = \{f(x) = w_1x_1 + w_2x_2\}$.



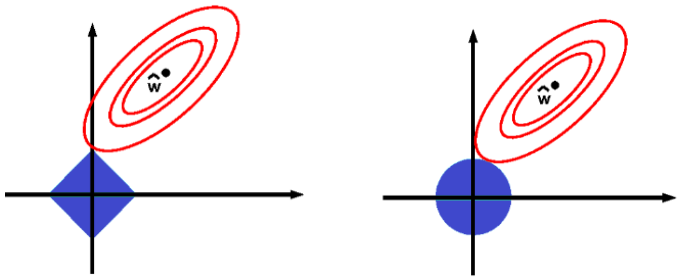
(a) l_1 contour: $|w_1| + |w_2| = r$

(b) l_2 contour: $w_1^2 + w_2^2 = r$

Figure 5: Represent $\mathcal{F}$ by $\{(w_1, w_2) \in \mathbf{R}^2\}$

The contours of RSS[2] objective function above shows why l_1 regularization yield sparse solution *(Kevin M. Chapter 13)*: The optimal solution occurs at the point where the lowest level set of the objective function intersects the constraint surface. For the OLS

[2]RSS: Residual Sum of Squares, $RSS(w) = \sum_{i=1}^{n}(y_i - w^T x_i)^2$.



(a) l_1 (b) l_2

Figure 6: Contour of RSS objective function. $f_r^* = \arg\min_{w \in \mathbf{R}^2} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2$
Blue Region: Area satisfying complexity constraint, r.
Red lines: Contours of RSS $\sum_{i=1}^{n} (w^T x_i - y_i)^2$ as it moves away from the minimum.

set-up given above, the RSS objective function is minimized by $\hat{w} = (X^T X)^{-1} X^T y$. It should be geometrically clear that as we relax the constraint r, we "grow" the l_1 diamond and l_1 diamond until it meets the objective; the corners of the diamond are more likely to intersect the ellipse than one of the sides, especially in high dimensions, because the corners "stick out" more. The corners correspond to sparse solutions, which lie on the coordinate axes. By contrast, when we "grow" the l_2 ball, it can intersect the objective at any point; there are no "corners", so there is no preference for sparsity.

Todo: Homework Question and 'Completing-the-square.pdf' todo

A more rigorous way to see l_1 sparsity needs introduction of subgradients/subderivative and subdifferential. todo

3.3 Proper Penalty (r or λ):

In LASSO, larger r results in more 0 coefficients. In Ridge Regression, larger r results in more smaller coefficients yet not 0's. Both approaches need proper r value. A larger r tends to have the model under-fitting, a smaller r tends to have the model over-fitting. We use AIC 赤池信息准则 or BIC 贝叶斯信息准则 to evaluate the model. explain

3.4 Regularization from a Bayesian Perspective

4 LASSO Solution

4.1 LASSO as Quadratic Program

4.2 Coordinate Descent (Shooting Method)

5 Regularization in Artificial Neural Networks

5.1 Weight Decay

Usually a weight update rule is:

$$w \leftarrow w - \eta \cdot \frac{\partial L}{\partial w} \tag{5}$$

However, when we have less and less input data for some weights, we want to make sure those weights become smaller:

$$w \leftarrow \lambda \cdot w - \eta \cdot \frac{\partial L}{\partial w} \text{ , where } \lambda \in (0, 1) \text{ , often } 0.99. \tag{6}$$

5.2 Drop-Out

An empirical trick that eliminates some percentage of the neurons in a layer; in order to train all submodels with a bagging-like criterion.

5.3 Max Pooling

6 Other solutions to overfitting

- Cross-validation.
- Collect more data for training.
- Remove 'Irrelevant' features.
- Early Stopping.
- Ensemble methods, i.e., Bagging, Boosting,

7 Python Code

8 Takeaways

L1 regularization
helps perform feature selection in sparse feature spaces; however, does not perform better than L2 in practice.

When to use l_1-regularized classifiers?
When to identify important features. For most cases, l_1 regularization does not give higher accuracy.

Why do we still have to worry about this basic models when there are many advanced approaches?
We always need to build baseline model to compare with fancy model, so that we know the fancier model is working a lot better. If the advanced model cannot compete with a baseline linear model, that means either we can just use a easy linear model or we have to figure out what's wrong with the advanced model, i.e., hyperparameter settings.

9 References:

- Kevin Murphy, Machine Learning: A Probabilistic Perspective
- David S. Rosenberg, ML1003 NYU.
- LIBLINEAR FAQ, https://www.csie.ntu.edu.tw/~cjlin/liblinear/FAQ.html#training_and_prediction
- What is the difference between L1 and L2 regularization? How does it solve the problem of overfitting? Which regularizer to use and when? Quora