INF554. Machine Learning

Jesse Read

Regularization: A Summary and Further Notes

Outline

Outline

Motivation for Regularization

- Generalization (prevent overfitting!)
- When number of features $>$ number of examples
- Prevent numerical overflow
- Interpretation
- Dealing with collinearity / feature correlation.
- Efficiency (reduced complexity)

In machine learning we are interested in the expected loss,

$$J(\theta) = \mathbb{E}_{(x,y)\sim p}[\ell(Y, f(x))] = \int \ell(y, f(x))\, \mathrm{d}p(x, y)$$

where f defined by θ. Yet we don't have p. We usually minimize empirical risk:

$$\widehat{J}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f(x_i))$$

Kinds of Regularization

- Use a simple model
- Variable selection
- Ridge aka weight decay
- Lasso
- Principle-components regression
- Early Stopping
- Ensembles
- Cross validation
- Dropout (in neural networks)
- Dataset augmentation (add noise to inputs; classes)
- Adversarial training

In general:

$$\min_{\mathbf{w}} \{L(\mathbf{w}, \mathbf{X}, \mathbf{y}) + R(\mathbf{w})\}$$

Ridge:

$$\min_{\mathbf{w}} \left\{L(\mathbf{w}, \mathbf{X}, \mathbf{y}) + \lambda\|\mathbf{w}\|_2^2\right\}$$

Lasso:

$$\min_{\mathbf{w}} \{L(\mathbf{w}, \mathbf{X}, \mathbf{y}) + \lambda\|\mathbf{w}\|_1\}$$

Elastic Net:

$$\min_{\mathbf{w}} \left\{L(\mathbf{w}, \mathbf{X}, \mathbf{y}) + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{w}\|_2^2\right\}$$

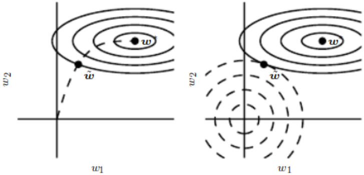
Under loss function L. Larger values of λ specify stronger regularization.

Various Remarks

- Never ever train on the test data!
- The scale of features matters when regularizing.
- Don't regularize the bias

- Don't overlook the potential of random features + regularization; e.g., "Extreme Learning Machines" and Reservoir Computing (Echo State Networks and Liquid State Machines); Black-box approach (e.g.,)
- Regularization in the Big Data era still important? Yes: Even more important!
  - Data often grows along columns as well as rows
  - Deep neural networks are powerful and complex.
- However: Some methods like CNNs are at risk of *under*-fitting! (Due to: parameter sharing, pooling layer, invariance to translation, . . .)

Early Stopping



1. Split training set into $\mathbf{X}_{\text{sub}}, \mathbf{y}_{\text{sub}}$ and $\mathbf{X}_{\text{val}}, \mathbf{y}_{\text{val}}$
2. Train on $\mathbf{X}_{\text{sub}}, \mathbf{y}_{\text{sub}}$ (update $\mathbf{w}$ over n iterations)
3. Stop when error rate $J(\mathbf{y}_{\text{val}}, h(\mathbf{X}_{\text{val}}; \mathbf{w}_n)$ stops decreasing

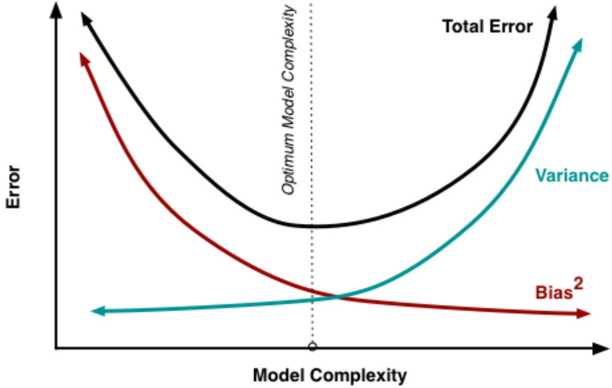
And when we stopped?

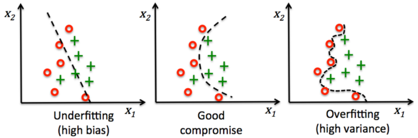
1. $\epsilon \leftarrow J(\mathbf{y}_{\text{sub}}, h(\mathbf{X}_{\text{sub}}, \mathbf{w}_n))$
2. while $J(\mathbf{w}, \mathbf{X}_{\text{val}}, \mathbf{y}_{\text{val}}) > \epsilon$:
   Train on $\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}$ some more

Bias-Variance Tradeoff

Where $\hat{f} := \hat{f}(\mathbf{x})$, $y = f(\mathbf{x}) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$:

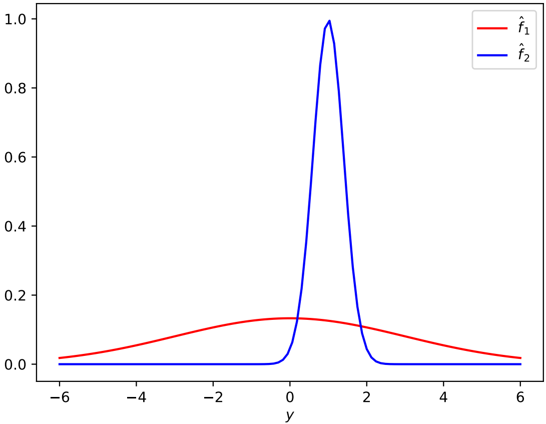
$$
\begin{aligned}
\mathbb{E}[\text{MSE}] &= \mathbb{E}[(y - \hat{f})^2] \\
&= \mathbb{E}[y^2 - 2y\hat{f} + \hat{f}^2] \\
&= \mathbb{E}[y^2] - \mathbb{E}[2y\hat{f}] + \mathbb{E}[\hat{f}^2] \\
&= \mathbb{V}[y] + \mathbb{V}[\hat{f}] + (\mathbb{E}[y])^2 + (\mathbb{E}[\hat{f}])^2 - 2f\mathbb{E}[\hat{f}] \\
&= \sigma^2 + \mathbb{V}[\hat{f}] + f^2 + (\mathbb{E}[\hat{f}])^2 - 2f\mathbb{E}[\hat{f}] \\
&= \sigma^2 + \mathbb{V}[\hat{f}] + (f - \mathbb{E}[\hat{f}])^2 \\
&= \underbrace{\sigma^2}_{\text{irreducible error}} + \underbrace{\mathbb{V}[\hat{f}]}_{\text{variance}} + \underbrace{(\mathbb{E}[f - \hat{f}])^2}_{\text{bias}^2}
\end{aligned}
$$


Optimum Model Complexity
Total Error
Variance
Bias^2
Error
Model Complexity



- Sources of bias: under-fitting: poor modelling of decision boundaries, incorrect assumptions, and models.
- Sources of variance: small local subsets of data (e.g., in decision trees near the leaves), too-local classifiers (kNN, large decision trees), high randomization (e.g., NNs), and unstable algorithms (e.g., decision trees).
- We can reduce bias to 0 (e.g., least squares on linear problem),
- such an unbiased estimator can correspond to intuition, but we should not reduce bias at any cost!

- A straight line on complex data has much bias, little variance
- Flexible methods: less bias, but more variance.
- OLS is an unbiased estimator, Ridge regression is biased.



An estimator with some bias can perform better than an unbiased estimator.

Bagging as a Regularizer

Motivation: decision trees (for example) have high variance $\mathrm{Var}[\hat{f}]$. (On any given dataset $\mathcal{D}_1 \sim p$, our model $\hat{f}_1$ is likely to be quite different from $\hat{f}_2$ built from another dataset $\mathcal{D}_2 \sim p$). How to reduce this variance?

$$\mathrm{Var}(\text{Using Bagging}) = \frac{1}{M}\mathrm{Var}[\text{Using Single Model}] \quad \text{(best case; if uncorrelated)}$$

Example: two models, two estimates, $\hat{y}^{(1)} = 0.5$, $\hat{y}^{(1)} = 1.5$ (suppose that $y = 0$).

$$\mathbb{E}[\text{MSE of bagging}] \leq \mathbb{E}[\text{MSE of model}] \quad \bullet\ ?$$
$$(1^2) \leq \frac{1}{2}\left(0.5^2 + 1.5^2\right) \quad \bullet\ !$$
$$1 \leq 1.25$$

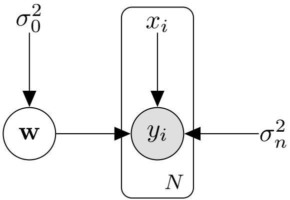
How to get more models? We were only given one dataset. Use that one to make more datasets

$$\mathcal{D}_m \sim \mathcal{D}$$

Bayesian View of Regularization

- Bayesian view: **w** are random variables, with a distribution
- We place a particular prior distribution, $p(\mathbf{w})$
- ...then calculate the posterior $p(\mathbf{w}|\mathbf{y})$
- With the likelihood, $p(\mathbf{y}|\mathbf{w})$, we can use Bayes' Rule:

$$p(\mathbf{w}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{y})}$$



- MAP (Maximum a Posteriori):

$$\underset{\mathbf{w}}{\operatorname{argmax}} \log p(\mathbf{w}|\mathbf{y}) = \underset{\mathbf{w}}{\operatorname{argmax}} \left\{ \underbrace{\log p(\mathbf{y}|\mathbf{w})}_{\text{Regularization}} + \underbrace{\log p(\mathbf{w})}_{\text{MLE loss}} \right\}$$

- N.B. Connection to Ridge regression!