

Advice for Applying Machine Learning

Sidharth Baskaran

June 2021

Improving algorithm

- If there are large errors in predictions
 - More training examples
 - Smaller set of features
 - Additional features
 - Add polynomial features (i.e. x_1x_2, x_1^2, x_2^2)
 - Decrease or increase λ
- Machine learning diagnostic
 - Test run to understand performance of algorithm

Evaluating a Hypothesis

- Low training error does not mean good $h_\theta(x)$
- Hard to plot hypothesis as $n \rightarrow$ large
- Split data into 2 parts \rightarrow test and training
- Training/testing procedure for linear regression
 - Learn parameter θ from training data (minimize training error $J(\theta)$)
 - Compute test error
 - * $J_{\text{test}}(\theta) = \frac{1}{2m_{\text{test}}} \sum_{i=1}^{m_{\text{test}}} (h_\theta(x_{\text{test}}^{(i)}) - y_{\text{test}}^{(i)})^2$
 - * Change appropriately for logistic regression
 - Misclassification error (0/1 misclassification error)
 - * Test error = $\frac{1}{m_{\text{test}}} \sum_{i=1}^{m_{\text{test}}} \text{err}(h_\theta(x_{\text{test}}^{(i)}), y_{\text{test}}^{(i)})$
 - Is proportion of misclassified test data

$$\text{err}(h_\theta(x), y) = \begin{cases} 1 & y = 0 \text{ if } h_\theta(x) \geq 0.5 \text{ or } y = 1 \text{ if } h_\theta(x) < 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Model Selection and training/validation/test sets

- Error of parameters as measured to fit to a set of data < than actual generalization error
- Let there be a list of hypotheses of varying polynomial degree d and $i \in \text{range}(1, d)$
 - Calculate $J_{\text{test}}(\theta^{(i)})$ in range
 - Lowest J_{test} means best model \rightarrow but not fair estimate of generalization \rightarrow optimistic choice
- Split data set into 3 parts
 - E.g. 60% training, 20% cross-validation (CV), 20% test
 - Determine $J_{\text{train}}, J_{\text{CV}}, J_{\text{test}}$
 - In list of polynomial models - find lowest J_{CV} from CV set
 - Estimate generalization error from polynomial indexed d with lowest CV error

Bias vs. Variance

- Increase degree d of polynomial \rightarrow decreases training error of polynomial
 - Test error very similar
- Cross-validation error initially decreases then increases as overfitting begins to occur
- **High bias** problem \rightarrow low d and a high error
 - High $J_{\text{train}}(\theta)$ and $J_{\text{CV}}(\theta) \approx J_{\text{train}}(\theta)$
- **High variance** problem \rightarrow high d and a high error
 - $J_{\text{train}}(\theta)$ is low
 - $J_{\text{CV}}(\theta) \gg J_{\text{train}}(\theta)$

Regularization and bias/variance

- High λ means penalized parameters θ so high bias \rightarrow underfit
- Intermediate $\lambda \rightarrow$ optimal
- Small $\lambda \rightarrow$ high variance and overfit
- Choosing regularization parameter λ
 - $J_{\text{train}}(\theta)$, $J_{\text{CV}}(\theta)$, $J_{\text{test}}(\theta)$ all do not have regularization term but $J(\theta)$ does
 - Try a range of λ , e.g. in multiples of 2 and calculate the corresponding $J_{\text{CV}}(\theta)$
 - * Pick choice with lowest CV error
 - Then apply to $J_{\text{test}}(\theta)$ to check for good generalization
- Can then plot $J_{\text{train}}(\theta)$ and $J_{\text{CV}}(\theta)$ as a function of λ
 - $J_{\text{train}}(\theta)$ will increase
 - $J_{\text{CV}}(\theta)$ will be upward parabolic

Learning Curves

- Plot either training or CV error
- A small training set size m means virtually no error
 - As m increases, avg. training error of hypotheses increases
- CV error is high (low generalization) \rightarrow small m
 - Tend to decrease with m
- More training data does not help case of high bias
- High variance problem
 - CV error will decrease with higher m
- As m increases, $J_{\text{CV}}(\theta) - J_{\text{train}}(\theta)$ and both curves approach each other

High bias

Value of m	$J_{\text{train}}(\theta)$	$J_{\text{CV}}(\theta)$
Low	Low	High
High	High	Low

High variance

Value of m	$J_{\text{train}}(\theta)$	$J_{\text{CV}}(\theta)$
Low	Low	High
High	Increases	Decreases

Debugging learning algorithm

- Choices
 - More training examples (m) \rightarrow fixes high variance
 - Smaller sets of features \rightarrow fixes high variance
 - Additional features \rightarrow fixed high bias
 - Adding polynomial features \rightarrow fixes high bias
 - Decreasing $\lambda \rightarrow$ fixes high bias
 - Increasing $\lambda \rightarrow$ fixed high variance
- Small neural network \rightarrow prone to underfitting due to less parameters, computationally cheaper
- Large neural network \rightarrow prone to overfitting and computationally expensive
 - Can address with λ

Machine Learning System Design

- Supervised learning $\rightarrow x =$ features of email, $y \in \{0, 1\}$, can choose 100 words indicative of spam/not for features
- Can encode an email into a feature vector
 - In practice \rightarrow take most frequently occurring n words in training set
- Could develop features to reduce errors, e.g. misspelling detection
 - Equal consideration of all options \rightarrow cannot tell which will work best

Error Analysis

- Start with **simple** algorithm to test on CV data
- Plot learning curves to decide if more data, features, etc.
- Error analysis \rightarrow manual examination of examples where errors occurred
 - Look for systematic error trend
 - Use evidence to guide decision-making not guesswork
- Numerical evaluation
 - Treating stem of word = to variants of word
 - Can use stemming software
 - Naturally \rightarrow CV error $J_{CV}(\theta)$ of algorithm with/without stemming and choose best options
 - * Do not $J_{train}(\theta)$ to allow for generalization

Error metrics for skewed classes

- Precision/recall
 - **Precision** \rightarrow Of all predictions $y = 1$, fraction that actually correspond to $y = 1$
 - * Precision = $\frac{\text{True pos.}}{\text{Predicted pos.}} = \frac{\text{True pos.}}{\text{True pos.} + \text{False pos.}}$
 - **Recall** \rightarrow Of all actual cases $y = 1$, what fraction was correctly detected by algorithm
 - * Recall = $\frac{\text{True pos.}}{\text{Actual pos.}} = \frac{\text{True pos.}}{\text{True pos.} + \text{False neg.}}$
- Tradeoff of precision/recall
 - If trying to have high confidence \rightarrow high precision, low recall
 - If trying to minimize false negatives \rightarrow high recall, low precision
- In general \rightarrow predict 1 if $h_{\theta}(x) \geq$ threshold
- F_1 score \rightarrow comparing precision/recall numbers
 - Can calculate average $\frac{P+R}{2}$ but susceptible to high or low recall/precision weighted
 - F_1 or F -score better $\rightarrow 2 \frac{PR}{P+R}$
 - * Gives more weight to $\min(P, R)$

Data for Machine Learning

- Large data rationale
 - Assume feature $x \in \mathbb{R}^{n+1}$ has enough information to predict y accurately
 - Can ask if given input x , can human expert confidently predict y
 - A learning algorithm with many parameters or NN with many hidden layers
 - * $J_{\text{train}}(\theta)$ very small
 - Very large training set \rightarrow unlikely to overfit
 - * $J_{\text{train}}(\theta) \approx J_{\text{test}}(\theta)$