Cross-Validation

Test Errors

How you can tell that a hypothesis overfits?

- plotting - not always good

We can split all the data into 2 subsets

- training set $\approx$ ≈ **70%** of data, m m - number of examples in the training set
- testing set $\approx$ ≈ **30%** of data, m_{test} mtest - number of examples in the testing set

it's better to choose examples for training/testing sets randomly

Error Metrics

	Prediction	**Classificatio**
Example Model	Linear Regression	Logistic Regress
Test Error	$J_{test}(\theta) = \frac{1}{m_{test}} \sum error(h_\theta(x^{(i)}_{test}), y^{(i)}_{test})$ Jtest(θ)=1mtest∑error(hθ(xtest(i)),yte	
$error(h_\theta(x), y)$ error(hθ(x),y)	Average Square Error $error(h_\theta(x), y) = \frac{1}{2}(h_\theta(x) - y)^2$ error(hθ(x),y)=12(hθ(x)−y)2	Misclassification E $error(h_\theta(x), y) = \begin{cases} 0 \text{ if classification is correct} \\ 1 \text{ otherwise} \end{cases}$ error(

Cross-Validation

Generally cross-validation is used to find the best value of some parameter

- we still have training and test sets
- but additionally we have a cross-validation set to test the performance of our model depending on the parameter

Motivation: Model Selection Problem

We have a following problem of Model Selection

> Suppose we are about to create a model and not sure what degree of polynomial to choose

So the problem

- So we want to try 10 models
- let d d denote the degree of polynomial
- $d = 1 : h_\theta = \theta_0 + \theta_1 x$ d=1:hθ=θ0+θ1x
- $d = 2 : h_\theta = \theta_0 + \theta_1 x + \theta_2 x^2$ d=2:hθ=θ0+θ1x+θ2x2
- ...
- $d = 10 : h_\theta = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_{10} x^{10}$ d=10:hθ=θ0+θ1x+θ2x2+...+θ10x10

Results

- Each d d will give us a vector (or matrix) $\theta^{(d)}$ θ(d)
- Now we can try all d d models and see which gives the best (lowest) $J_{test}(\theta^{(i)})$ Jtest(θ(i))
- Let's say we decided to choose 5th model

How well this model generalize?

- the test error is $J_{test}(\theta^{(5)})$ Jtest(θ(5))

- but $J_{test}(\theta^{(5)})$ Jtest(θ(5)) is a very optimistic estimate of the generalization error
  - (because the lowest/best error was picked up - so the test error might be biased towards that)
- i.e. we the extra parameter d was fit to test, and it's not fair to estimate our hypothesis on the test set, because we already used that set to get the best d

To address that problem

- instead of splitting the data set into 2 categories, we split into 3 sets:
- training set ($\approx$ **≈ 60%**)
  - $x^{(i)}, y^{(i)}$ x(i),y(i), total m m examples
- cross-validation set (or cv, $\approx$ **≈ 20%**)
  - $x_{cv}^{(i)}, y_{cv}^{(i)}$ xcv(i),ycv(i), total m_{cv} mcv examples
- test set ($\approx$ **≈ 20%**)
  - $x_{test}^{(i)}, y_{test}^{(i)}$ xtest(i),ytest(i), total m_{test} mtest examples

Now we can define

- *Training error*

$$J(\theta) = J_{train}(\theta) = \frac{1}{2m}\sum cost(x^{(i)}, y^{(i)})$$ J(θ)=Jtrain(θ)=12m∑cost(x(i),y(i))

- *Cross-Validation error*

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum cost(x_{cv}^{(i)}, y_{cv}^{(i)})$$ Jcv(θ)=12mcv∑cost(xcv(i),ycv(i))

- *Test Error*

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum cost(x_{test}^{(i)}, y_{test}^{(i)})$$ Jtest(θ)=12mtest∑cost(xtest(i),ytest(i))

So for Model Selection to fit d, we

- obtain $\theta^{(1)}, \ldots, \theta^{(d)}$ θ(1),...,θ(d) and select best (lowest) $J_{cv}(\theta^{(i)})$ Jcv(θ(i))
- estimate generalization error for the test set $J_{test}(\theta^{(i)})$ Jtest(θ(i))

Tuning Learning Parameter

General algorithm

- Split data set into
  - Learning set
  - Validation set
  - Test set
- use validation set for tuning control parameters
- use test set only for final evaluation

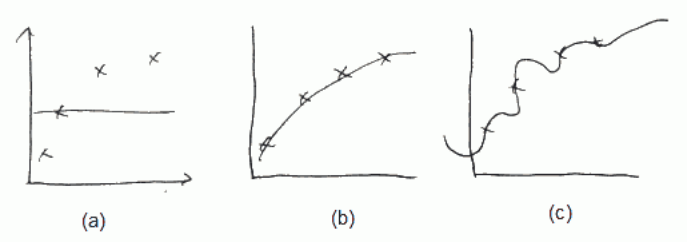
Let γ γ be the control parameter

- for every possible value of γ γ
  - build a classification model C C using the learning set
  - use the validation set to estimate the expected error rate of C C
- select the optimal value of the control parameter
  - that is the value of γ γ s.t. the error rate is minimal

Regularization Parameter λ λ

Suppose now we're fitting a model with high-order polynomial

- $h_\theta(x) = \theta_0 + \theta_1 x + \ldots + \theta_4 x^4$ hθ(x)=θ0+θ1x+...+θ4x4
- to prevent overfitting we use regularization
- $J(\theta) = \frac{1}{m}\sum cost(h_\theta(x^{(i)}), y^{(i)}) + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$ J(θ)=1m∑cost(hθ(x(i)),y(i))+λ2m∑j=1mθj2



- (a) If λλ if large (say $\lambda = 10000$ λ=10000), all θθ are penalized and $\theta_1 \approx \theta_2 \approx \ldots \approx 0$ θ1≈θ2≈...≈0, $h_\theta(x) \approx \theta_0$ hθ(x)≈θ0
- (b) if λλ is intermediate, we fit well
- (c) if λλ is small (close to 0) we fit too well, i.e. we overfit

How can we chose good λλ? Let's define

- $J_{train}(\theta) = \frac{1}{m} \sum \text{cost}(h_\theta(x^{(i)}), y^{(i)})$ Jtrain(θ)=1m∑cost(hθ(x(i)),y(i)) (same as $J(\theta)$ J(θ), but without regularization)
- $J_{cv}(\theta)$ Jcv(θ) and $J_{test}(\theta)$ Jtest(θ) - same, but on cross-validation and test datasets respectively

Now we

- Choose a range of possible values for λλ (say 0, 0.01, 0.02, 0.04, ..., 10.24) - that gives us 12 models to checks
- For each $\lambda^{(i)}$ λ(i),
  - calculate $\theta^{(i)}$ θ(i),
  - calculate $J_{cv}(\theta^{(i)})$ Jcv(θ(i)),
  - and take $\lambda^{(i)}$ λ(i) with lowest $J_{cv}(\theta^{(i)})$ Jcv(θ(i))

After that we report the test error $J_{test}(\theta^{(i)})$ Jtest(θ(i))

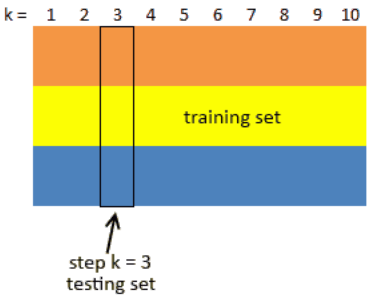
K-Fold Cross-Validation

K K-Fold Cross-Validation If we want to reduce variability in the data

- we can perform multiple rounds of cross-validation using different partitions
- and then average the results over all the rounds

We're given a dataset S S sampled from the population D D

- we partition S S into K K equal disjoint subsets $(T_1, \ldots, T_K)$ (T1,...,TK)(typically 5-10 subsets)
- then perform K K steps, and at step k k do:
  - use $R_k = S - T_k$ Rk=S−Tk as the training set
  - build classifier C_k Ck using R_k Rk
  - use T_k Tk as the test test, compute error $\delta_k = \text{error}(C_k, T_k)$ δk=error(Ck,Tk)
- let $\delta^* = \frac{1}{K} \sum_{k=1}^{K} \delta_k$ δ∗=1K∑k=1Kδk
  - this is the expected error rate



- note that there's only two subsets at each iteration
  - and they are used to estimate the true error

Tuning Learning Parameter

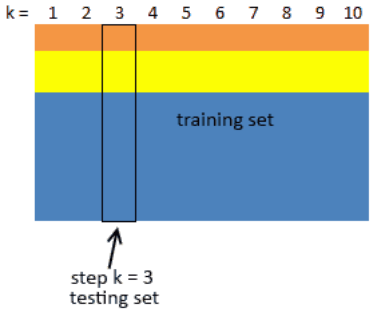
Choosing best value for parameter γ γ with K K-Fold Cross-Validation

- you have to put some data aside before training your classifier
  - i.e. split your data into learning set and test set

- for every possible value of γ γ
  - use K K-fold Cross-Validation to estimate the expected error rate
  - use the learning set as the set S S (and do K K-fold Cross-Validation only on it)
- select the optimal value of γ γ
- and then use the test set for the final evaluation

Stratified K-Fold Cross-Validation

What if we want to preserve the class distribution over K K runs?

- then for each T_i Ti pick up the same proportion of labels as in the original dataset
- 

- it's very important if the test distribution is not uniform

See Also

- Overfitting
- Model Selection

Sources

- Machine Learning (coursera)
- Data Mining (UFRT)
- http://en.wikipedia.org/wiki/Cross-validation_(statistics)

Categories: Machine Learning | Model Performance Evaluation | Classifiers