Dis Fri

Friday, October 19, 2018 12:06 PM

Train vs. Test vs. Dev

- train - to train the model
- test - to test accuracy of the model
- dev - to tune hyperparameters

Hyperparameter vs. Parameter vs. Coefficient

- Hyperparameter - a value you have to set and fine tune
  - eg step size in gradient descent
- Parameter - values you're training
- Coefficient - weights that you multiply by

Cross-Validation

- splitting train / test / dev in different ways
  - could be used for variety of test sets in training model
  - or for variety of dev sets in tuning hyperparameters

Data Preprocessing

- want to adjust the data to make training smooth
- eg standardize to 0 mean and 1 stdev

$$x_{nd} \leftarrow \frac{x_{nd} - \overline{x_d}}{s_d}$$

Curse of Dimensionality

- high sparsity in the data set in a lot of dimensions
- so distance-based algorithms may not function as well in high dimensions

Perceptron Algorithm

- linear classification algorithm by finding a separating hyperplane
- model is learnt by making mistakes
- the algorithm converges in finite steps
  - Mistake Bound Theorem - if there exists a set of weights that are consistent with the data then the perceptron algorithm will converge. If the training data is not linearly separable, then the learning algorithm will eventually repeat in infinite loop
  - suppose there exists a unit vector $u \in R^n$ s.t. for some $\gamma > 0$ we have $y_i(u^T x_i) \geq \gamma$
  - Then the perceptron algorithm will make at most $\left(\frac{\boldsymbol{R}}{\boldsymbol{\gamma}}\right)^{\mathbf{2}}$
- key unit in perceptron network is the linear threshold unit - we take the $\mathrm{sgn}\left(\sum w_i x_i\right)$ to determine a binary classification

Algorithm:

1. Initialize $w \leftarrow 0 \in R^n$
2. For (x, y) in D:
   a. $\widehat{\boldsymbol{y}} = \mathrm{sgn}\left(\boldsymbol{w}^T \boldsymbol{x}\right)$
   b. $if\ \widehat{\boldsymbol{y}} \neq \boldsymbol{y}, \boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{yx}$
3. return **w**

Pf

See lecture slides

(Demo of perceptron algorithm in python)