Machine Learning Basics: Building a Machine Learning Algorithm

Sargur N. Srihari
srihari@cedar.buffalo.edu

Topics

1. Learning Algorithms
2. Capacity, Overfitting and Underfitting
3. Hyperparameters and Validation Sets
4. Estimators, Bias and Variance
5. Maximum Likelihood Estimation
6. Bayesian Statistics
7. Supervised Learning Algorithms
8. Unsupervised Learning Algorithms
9. Stochastic Gradient Descent
10. Building a Machine Learning Algorithm
11. Challenges Motivating Deep Learning

Recipe for Machine Learning

- All Machine Learning algorithms are instances of a recipe:
  1. Specification of a dataset
     - In Linear regression $\boldsymbol{X}$ and y
  2. A cost function $\boxed{J(\boldsymbol{w},\boldsymbol{b}) = -E_{x,y\sim\hat{p}_{data}} \log p_{\text{model}}(y \mid \boldsymbol{x})}$
  3. An optimization procedure
     - In linear regression: normal equations
  4. A model $\boxed{p_{\text{model}}(y \mid \boldsymbol{x}) = N(y; \boldsymbol{x}^T\boldsymbol{w} + \boldsymbol{b}, 1)}$

- Example of building a linear regression algorithm is shown next

Ex: Linear Regression Algorithm

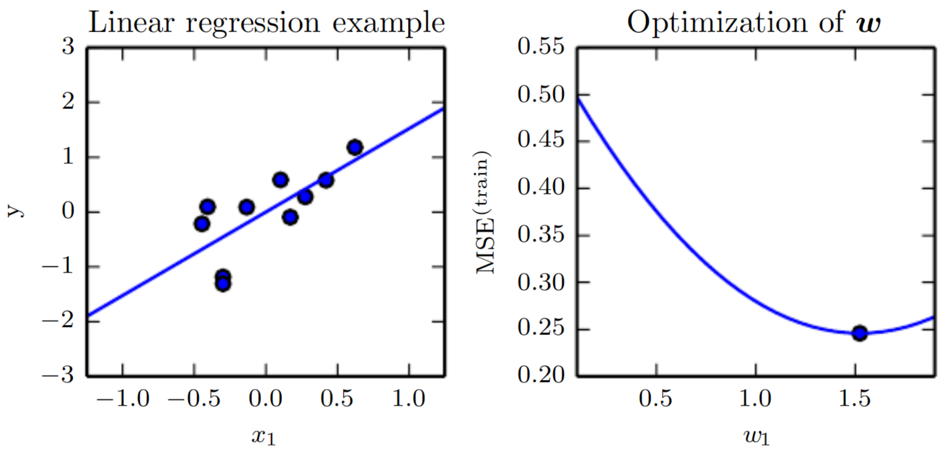
1. Data set : X and y
2. Cost function:

$$\boxed{J(\boldsymbol{w},\boldsymbol{b}) = -E_{x,y\sim\hat{p}_{data}} \log p_{\text{model}}(y \mid \boldsymbol{x})}$$

3. Model specification:

$$\boxed{p_{\text{model}}(y \mid \boldsymbol{x}) = N(y; \boldsymbol{x}^T\boldsymbol{w} + \boldsymbol{b}, 1)}$$

4. Optimization algorithm: solving for where the cost is zero using the normal equations



- We can replace any of these components mostly independently from the others and obtain a variety of algorithms

Recipe for Cost Function

1. Cost function typically has a term that causes learning to perform statistical estimation
   – Most common cost: negative log-likelihood
      - Minimizing the cost maximizes the likelihood
2. Cost function may include additional terms
   – E.g., we can add weight decay to get

$$\boxed{J(\boldsymbol{w},\boldsymbol{b}) = \lambda \left\|\boldsymbol{w}\right\|_2^2 - E_{x,y\sim \hat{p}_{data}} \log p_{\text{model}}(y \mid \boldsymbol{x})}$$

      - which still allows closed-form optimization

- If we change model to be nonlinear most cost functions cannot be optimized in closed-form
   – Requires numerical optimization: gradient descent

Recipe for unsupervised learning

- Same recipe for both supervised and unsupervised learning
- Data set contains only $\boldsymbol{X}$
- Cost and model needed
  - Ex: we can obtain the first PCA vector by specifying loss

$$\boxed{J(\boldsymbol{w}) = E_{\boldsymbol{x}\sim\hat{p}_{data}} \,||\, \boldsymbol{x} - r(\boldsymbol{x};\boldsymbol{w}) \,||_2^2}$$

    - While model is defined to have $\boldsymbol{w}$ with norm one and reconstructed function $\mathrm{r}(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\boldsymbol{w}$

Recipe explains all ML algorithms

- Most machine learning algorithms make use of this recipe
- Some models such as decision trees and k-means require special case optimizers
  - Because their cost functions have flat regions, gradient-based optimization is inappropriate
- Recipe helps to see different algorithms as part of a taxonomy of methods for doing related tasks

Intractable Cost

- Sometimes the cost function cannot be evaluated due to computational reasons
- In these cases we can still minimize it using iterative numerical optimization
  - As long as we have some way of approximating the gradient