

# Machine Learning Basics: Building a Machine Learning Algorithm

Sargur N. Srihari  
[srihari@cedar.buffalo.edu](mailto:srihari@cedar.buffalo.edu)

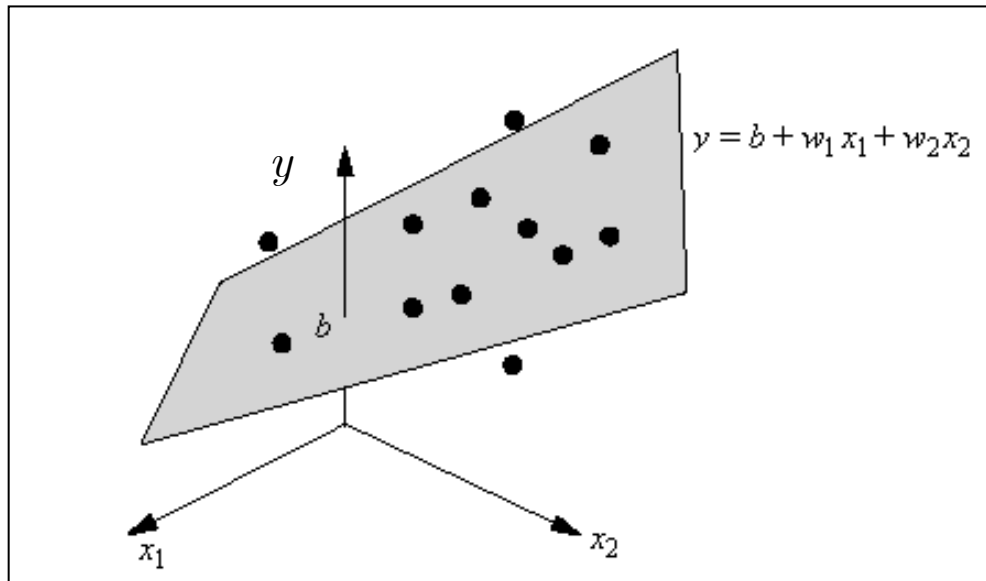
# Topics

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 is an instance of a recipe:
  1. Specification of a dataset
  2. A cost function
  3. An optimization procedure
  4. A model
- Example of building an ML model for linear regression is shown next

# Ex: Linear Regression Dataset



$x_1$	$x_2$	$t$
1	2	2
2	5	1
2	3	2
2	2	2
3	4	1
3	5	3
4	6	2
5	5	3
5	6	4
5	7	3
6	8	4
7	6	2
8	4	4
8	9	3
9	8	4

# Ex: Linear Regression Algorithm

1. Data set :  $X$  and  $y$

2. Cost function:

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

3. Model specification:

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

4. Optimization algorithm: solving for where the cost is minimal

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

# Linear Regression 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

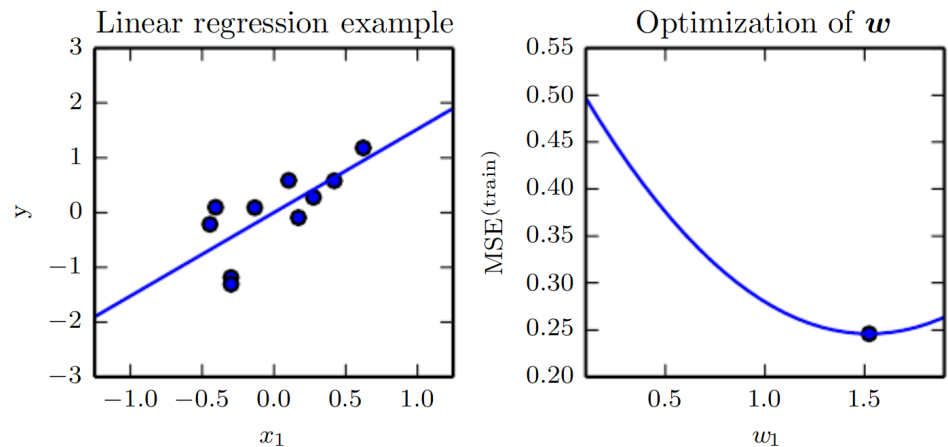
$$J(\mathbf{w}, \mathbf{b}) = \lambda \|\mathbf{w}\|_2^2 - E_{x, y \sim \hat{p}_{data}} \log p_{\text{model}}(y | \mathbf{x})$$

- which still allows closed-form optimization

# Optimization Procedure

- Cost function optimized in closed form for

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



- 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  $\mathbf{X}$
- Cost and model needed
  - Ex: we can obtain the first PCA vector by specifying loss

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

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



# Recipe explains all ML algorithms

- Most ML 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