INFO251 – Applied Machine Learning

Lab 7

Suraj R. Nair

Announcements

- PS4 posted, due Monday March 12. Start early!
- Quiz 1 solutions in lecture tomorrow

Today's Topics

1. Putting it all together: training an ML algorithm from scratch
2. Common loss functions
3. Practice
   - Bivariate OLS, squared error loss
   - Multivariate OLS, squared error loss
   - Multivariate OLS, squared error loss with Ridge regularization
4. Cross validation for optimal regularization parameter

Training ML Algorithms

Table 1: The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Domingos, 2016

Training ML Algorithms (Linear Models)

1. Define a model
2. Define a loss function
   - Add regularization to the loss function [**find optimal parameter**]
3. Optimization
   - Gradient Descent [**batch size, step size / learning rate, stopping rule**]

1. Define a Model

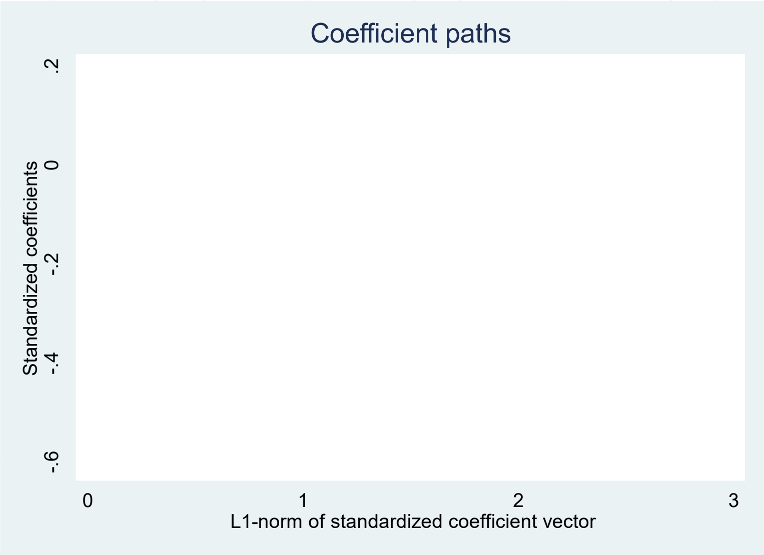
- For today: **Linear regression models**, of the form $y = \theta x$
  - Multivariate models: $y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$
  - Nonlinearities: $y = \theta_k x^k$
  - Interaction terms: $y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2$

2. Define a Loss Function

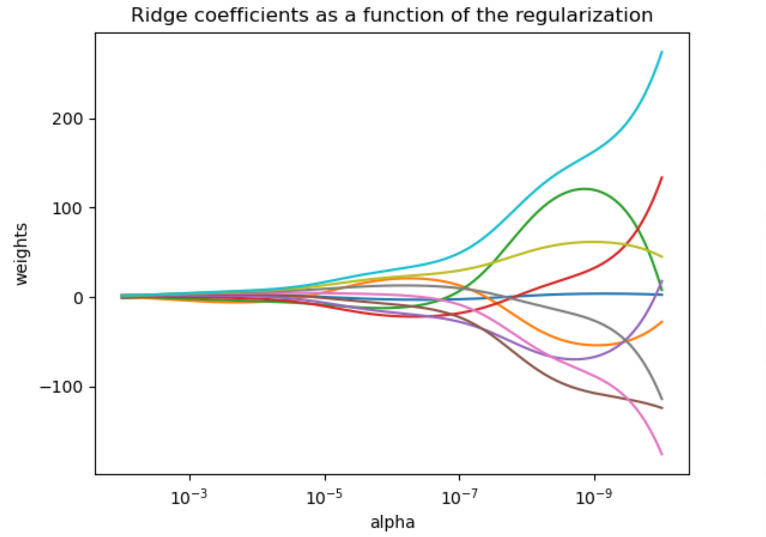
- Common loss functions for **regression**:
  - Squared error loss: $J(y, \hat{y}) = (y - \hat{y})^2$
  - Absolute error loss: $J(y, \hat{y}) = |y - \hat{y}|$
- Common loss functions for **binary classification**:
  - Logistic loss: $J(y, \hat{p}) = -(y \log(\hat{p}) + (1 - y)\log(1 - \hat{p}))$
  - Hinge loss: $J(y, \hat{p}) = \max(0, 1 - \hat{p}y)$
- Common loss functions for **multivariate classification**:
  - Cross-entropy loss: $J(y, \hat{p}) = \sum_{c=1}^{M} y_c \log(\hat{p}_c))$

3. Optionally Add Regularization to the Loss

- **LASSO**: $J(\theta) \mathrel{+}= \|\theta\|_1 = \sum_{j=1}^{k} |\theta_k|$



Ridge: $J(\theta) \mathrel{+}= \|\theta\|_2 = \sum_{j=1}^{k} \theta_k^2$



4. Gradient Descent

- Input:
  - parameters (θ)
  - gradient of the loss function with respect to the parameters ($d\theta$),
  - learning rate (α)
- Update parameters: $\theta_{new} = \theta - \alpha \times d\theta$
- Output: updated parameters (θ_{new})

Model: Linear regression (univariate)

Cost: Squared error

1. Define the model
2. Define the loss function
3. ~~Optionally add regularization to the loss function~~
4. Calculate partial derivatives
5. Write pseudocode

Model: Linear regression (multivariate)

Cost: Squared error

1. Define the model
2. Define the loss function
3. ~~Optionally add regularization to the loss function~~
4. Calculate partial derivatives
5. Write pseudocode

Model: Linear regression (multivariate)

Cost: Squared error + Ridge regularization

1. Define the model
2. Define the loss function
3. Optionally add regularization to the loss function
4. Calculate partial derivatives
5. Write pseudocode