

Regularized Linear Models

CS4780/5780 – Machine Learning
Fall 2019

Nika Haghtalab & Thorsten Joachims
Cornell University

Reading: UML 13.1, 9.2, 9.3

Discriminative ERM Learning

- Modeling Step:
 - Select classification rules H to consider (hypothesis space, features)
- Training Principle:
 - Given training sample $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$
 - Find h from H with lowest training error
→ Empirical Risk Minimization
 - Argument: low training error leads to low prediction error, if overfitting is controlled.
→ generalization error bounds
- Examples: SVM, decision trees, Perceptron

Generative vs. Conditional vs. ERM

- Empirical Risk Minimization
 - Find $h = \operatorname{argmin}_{h \in H} \text{Err}_S(h)$ s.t. overfitting control
 - Pro: directly estimate decision rule
 - Con: need to commit to loss, input, and output before training
- Discriminative Conditional Model
 - Find $P(Y|X)$, then derive $h(x)$ via Bayes rule
 - Pro: not yet committed to loss during training
 - Con: need to commit to input and output before training; learning conditional distribution is harder than learning decision rule
- Generative Model
 - Find $P(X,Y)$, then derive $h(x)$ via Bayes rule
 - Pro: not yet committed to loss, input, or output during training; often computationally easy (under strong assumptions)
 - Con: Needs to model dependencies in X

Bayes Decision Rule

- Assumption:
 - Learning task $P(X,Y)=P(Y|X)P(X)$ is known
- Question:
 - Given instance x , how should it be classified to minimize prediction error?
- Bayes Decision Rule (for zero/one loss):

$$h_{\text{bayes}}(\vec{x}) = \operatorname{argmax}_{y \in Y} [P(Y = y|X = \vec{x})]$$
- Bayes Decision Rule (general)

$$h_{\text{bayes}}(\vec{x}) = \operatorname{argmin}_{y \in Y} \left[\sum_{y'} \Delta(y', y) P(Y = y'|X = \vec{x}) \right]$$

Bayes Risk

- Given knowledge of $P(X,Y)$, the true error of the best possible h is

$$\text{Err}_P(h_{\text{bayes}}) = E_{x \sim P(X)} \left[\min_{y \in Y} (1 - P(Y = y|X = x)) \right]$$

for the 0/1 loss.

Logistic Regression

- Data:
 - $S = ((x_1, y_1) \dots (x_n, y_n))$, $x \in \mathbb{R}^N$ and $y \in \{-1, +1\}$
- Model:
 - $P(y|x, w) = \text{Ber}(y|\text{sigm}(w \cdot x))$
- Training objective:

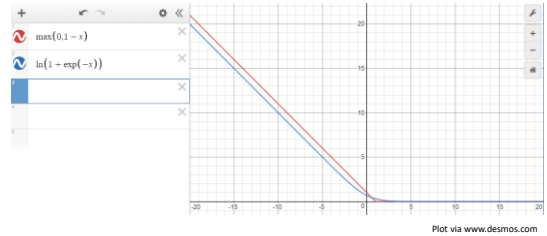
$$\hat{w} = \operatorname{argmin}_w \sum_{i=1}^n \log(1 + \exp(-y_i w \cdot x_i))$$
- Algorithm:
 - Stochastic gradient descent, Newton, etc.

Regularized Logistic Regression

- **Data:**
 - $S = ((x_1, y_1) \dots (x_n, y_n))$, $x \in \mathfrak{R}^N$ and $y \in \{-1, +1\}$
- **Model:**
 - $P(y|x, w) = \text{Ber}(y|\text{sigm}(w \cdot x))$, $P(w) = N(w|0, \Sigma)$
- **Training objective:**

$$\hat{w} = \underset{w}{\text{argmin}} \frac{1}{2} w \cdot w + C \sum_{i=1}^n \log(1 + \exp(-y_i w \cdot x_i))$$
- **Algorithm:**
 - Stochastic gradient descent, Newton, etc.

Logistic vs. Hinge Loss



Ridge Regression

- **Data:**
 - $S = ((x_1, y_1) \dots (x_n, y_n))$, $x \in \mathfrak{R}^N$ and $y \in \mathfrak{R}$
- **Model:**
 - $P(y|x, w) = N(y|w \cdot x, E)$, $P(w) = N(w|0, \Sigma)$
- **Training objective:**

$$\hat{w} = \underset{w}{\text{argmin}} \frac{1}{2} w \cdot w + C \sum_{i=1}^n (w \cdot x_i - y_i)^2$$
- **Algorithm:**
 - $\hat{w} = (\text{diag}(C) + X^T X)^{-1} X^T y$

Generative vs. Conditional vs. ERM

- **Empirical Risk Minimization**
 - Find $h = \underset{h \in H}{\text{argmin}} \text{Err}_S(h)$ s.t. overfitting control
 - Pro: directly estimate decision rule
 - Con: need to commit to loss, input, and output before training
- **Discriminative Conditional Model**
 - Find $P(Y|X)$, then derive $h(x)$ via Bayes rule
 - Pro: not yet committed to loss during training
 - Con: need to commit to input and output before training; learning conditional distribution is harder than learning decision rule
- **Generative Model**
 - Find $P(X, Y)$, then derive $h(x)$ via Bayes rule
 - Pro: not yet committed to loss, input, or output during training; often computationally easy (under strong assumptions)
 - Con: Needs to model dependencies in X

Discriminative Training of Linear Rules

$$\min_{w, b} R(w) + C \frac{1}{n} \sum_{i=1}^n L(w \cdot x_i + b, y_i)$$

Regularizer
 Regularization Parameter
 Training Cost / Empirical Risk / Training error

- **Soft-Margin SVM**
 - $R(w) = \frac{1}{2} w \cdot w$
 - $L(\bar{y}, y_i) = \max(0, 1 - y_i \bar{y})$
- **Perceptron**
 - $R(w) = 0$
 - $L(\bar{y}, y_i) = \max(0, -y_i \bar{y})$
- **Linear Regression**
 - $R(w) = 0$
 - $L(\bar{y}, y_i) = (y_i - \bar{y})^2$
- **Ridge Regression**
 - $R(w) = \frac{1}{2} w \cdot w$
 - $L(\bar{y}, y_i) = (y_i - \bar{y})^2$
- **Lasso**
 - $R(w) = \frac{1}{2} \sum |w_i|$
 - $L(\bar{y}, y_i) = (y_i - \bar{y})^2$
- **Regularized Logistic Regression / Conditional Random Field**
 - $R(w) = \frac{1}{2} w \cdot w$
 - $L(\bar{y}, y_i) = \log(1 + e^{-y_i \bar{y}})$