Logistic Regression

Cornelia Caragea

Department of Computer Science and Engineering
University of North Texas

Acknowledgments: Piyush Rai, Andrew Ng

June 22, 2016

Linear Classification

- **Goal:** Assign input vector $\mathbf{x}$ to one of the K discrete classes $\mathcal{C}_k$.
- Generally, the input space is divided into decision regions, whose boundaries are called *decision boundaries.*
- For linear models, decision boundaries are linear functions of the input vector $\mathbf{x}$.
- Data sets whose classes can be separated *exactly* by linear decision boundaries are said to be linearly separable.

Linear Classification

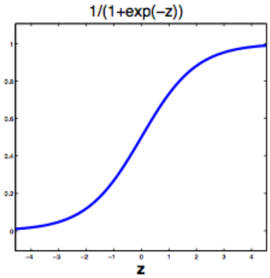
- **Goal:** Assign input vector $\mathbf{x}$ to one of the K discrete classes $\mathcal{C}_k$.
- Generally, the input space is divided into decision regions, whose boundaries are called *decision boundaries.*
- For linear models, decision boundaries are linear functions of the input vector $\mathbf{x}$.
- Data sets whose classes can be separated *exactly* by linear decision boundaries are said to be linearly separable.
- Examples of binary classification ($y \in \{0, 1\}$):
  - Email: spam / not spam?
  - Tumor: malignant / benign?

Linear Classification

- In regression problems, y is a real number, $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ (in the simplest case), where $h_{\mathbf{w}}(\mathbf{x})$ can be any real-valued number.
- In classification problems, we wish to predict discrete class labels, or more generally posterior probabilities that lie in the range $(0, 1)$, i.e., $0 \leq h_{\mathbf{w}}(\mathbf{x}) \leq 1$.
  - *Generalized linear models*: transform the linear function of $\mathbf{w}$ using a nonlinear function $\sigma(\cdot)$: $h_{\mathbf{w}}(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$.

Logistic Regression for Binary Classification

- Generalized linear model for classification where $\sigma(\cdot)$ is the logistic sigmoid function, i.e., $\sigma(z) = \frac{1}{1+e^{-z}}$



- Properties of σ:
  - Symmetry: $\sigma(-z) = 1 - \sigma(z)$
  - Inverse: $z = ln(\sigma/1 - \sigma)$ (aka logit function)
  - Derivative: $d\sigma/dz = \sigma(1 - \sigma)$

Logistic Regression for Binary Classification

Transform the linear function of $\mathbf{w}$ using $\sigma(\cdot)$

- Hypothesis Representation for Logistic Regression:

$$h_{\mathbf{w}}(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}},$$

  where $\mathbf{x}$ is a feature vector

- Hypothesis Output Interpretation:
  - $h_{\mathbf{w}}(\mathbf{x}) = P(y = 1|\mathbf{x}, \mathbf{w})$ - the confidence in the predicted label
  - $P(y = 0|\mathbf{x}, \mathbf{w}) = 1 - P(y = 1|\mathbf{x}, \mathbf{w})$
- Logistic regression seen as probabilistic discriminative model
  - Directly models conditional probabilities $P(y|\mathbf{x})$

Decision Boundary

- How does the decision boundary look like for Logistic Regression?
  - Suppose predict $y = 1$ if $h_{\mathbf{w}}(\mathbf{x}) \geq 0.5 \Leftrightarrow \mathbf{w}^T\mathbf{x} \geq 0$
  - Predict $y = 0$ if $h_{\mathbf{w}}(\mathbf{x}) < 0.5 \Leftrightarrow \mathbf{w}^T\mathbf{x} < 0$
- Decision boundary: $\mathbf{w}^T\mathbf{x} = 0$.
  - Hence, the decision boundary is therefore linear $\Rightarrow$ Logistic Regression is a linear classifier (note: it is possible to kernelize and make it nonlinear)

Cost Function

- Training set: $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$
- Hypothesis representation:

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

- How to choose parameters **w**?

Cost Function

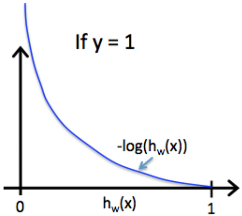
- Training set: $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$
- Hypothesis representation:

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

- How to choose parameters $\mathbf{w}$?
- Previously, for linear regression, $E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)})^2$
- For logistic regression, $E(\mathbf{w}) = \sum_{i=1}^{N} Cost(h_{\mathbf{w}}(\mathbf{x}^{(i)}), y^{(i)})$

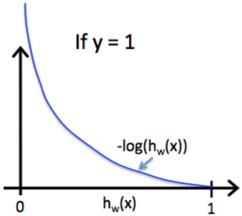
Cost Function

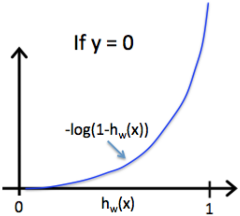
- $Cost(h_{\mathbf{w}}(\mathbf{x}), y) = -\log(h_{\mathbf{w}}(\mathbf{x}))$ if $y = 1$
- $Cost(h_{\mathbf{w}}(\mathbf{x}), y) = -\log(1 - h_{\mathbf{w}}(\mathbf{x}))$ if $y = 0$
- If $y = 1$
  - if $h_{\mathbf{w}}(\mathbf{x}) = 1,\ Cost = 0$
  - If $h_{\mathbf{w}}(\mathbf{x}) \to 0,\ Cost \to \infty$
  - Captures intuition that if $h_{\mathbf{w}}(\mathbf{x}) = 0$, but $y = 1$, we will penalize the learning algorithm by a very large cost.



Cost Function

- $Cost(h_{\mathbf{w}}(\mathbf{x}), y) = -\log(h_{\mathbf{w}}(\mathbf{x}))$ if $y = 1$
- $Cost(h_{\mathbf{w}}(\mathbf{x}), y) = -\log(1 - h_{\mathbf{w}}(\mathbf{x}))$ if $y = 0$
- If $y = 1$
  - if $h_{\mathbf{w}}(\mathbf{x}) = 1$, $Cost = 0$
  - If $h_{\mathbf{w}}(\mathbf{x}) \to 0$, $Cost \to \infty$
  - Captures intuition that if $h_{\mathbf{w}}(\mathbf{x}) = 0$, but $y = 1$, we will penalize the learning algorithm by a very large cost.





Cost Function

Cost Function for Logistic Regression:

$$
\begin{aligned}
E(\mathbf{w}) &= \sum_{i=1}^{N} Cost(h_{\mathbf{w}}(\mathbf{x}^{(i)}), y^{(i)}) \\
&= -\left[\sum_{i=1}^{N} y^{(i)} \log(h_{\mathbf{w}}(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\mathbf{w}}(\mathbf{x}^{(i)}))\right]
\end{aligned}
$$

To fit parameters **w**:

$$\min_{\mathbf{w}} E(\mathbf{w})$$

To make a prediction given a new **x**: Output

$$h_{\mathbf{w}}(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T \mathbf{x}}}$$

Gradient Descent

Cost Function for Logistic Regression:

$$E(\mathbf{w}) = -\left[\sum_{i=1}^{N} y^{(i)} \log(h_{\mathbf{w}}(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\mathbf{w}}(\mathbf{x}^{(i)}))\right]$$

Want:

$$\min_{\mathbf{w}} E(\mathbf{w})$$

Repeat until convergence {

$$w_j := w_j - \alpha \frac{\partial E(\mathbf{w})}{\partial w_j}$$

} (simultaneously update all w_j).

Gradient Descent

Cost Function for Logistic Regression:

$$E(\mathbf{w}) = -\left[\sum_{i=1}^{N} y^{(i)} \log(h_{\mathbf{w}}(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\mathbf{w}}(\mathbf{x}^{(i)}))\right]$$

Want:

$$\min_{\mathbf{w}} E(\mathbf{w})$$

Repeat until convergence {

$$w_j := w_j - \alpha \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}$$

} (simultaneously update all w_j).

The algorithm looks the same as for linear regression! Is it?

Gradient Descent

Cost Function for Logistic Regression:

$$E(\mathbf{w}) = -\left[\sum_{i=1}^{N} y^{(i)} \log(h_{\mathbf{w}}(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\mathbf{w}}(\mathbf{x}^{(i)}))\right]$$

Want:

$$\min_{\mathbf{w}} E(\mathbf{w})$$

Repeat until convergence {

$$w_j := w_j - \alpha \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}$$

} (simultaneously update all w_j).

The algorithm looks the same as for linear regression! Is it? No.

Multiclass Logistic Regression

Examples:

- Email foldering/tagging: Work, Friends, Family, Hobby
- Medical diagrams: Not ill, Cold, Flu
- Research articles by topics: Machine Learning, Data Mining, Algorithms

Multiclass logistic regression ($k > 2$):

- We maintain a separator weight vector $\mathbf{w}_k$ for each class k

Multiclass Logistic Regression

- Train a logistic regression classifier $h_{\mathbf{w}}^{(k)}(\mathbf{x})$ for each class k to predict the probability that class is k.
- On a new input $\mathbf{x}$, to make a prediction, pick the class k that maximizes

$$max_{\mathbf{k}} h_{\mathbf{w}}^{(k)}(\mathbf{x})$$

Nonlinear Basis Functions in Linear Models

- We use linear classification models
  - If non-linearity in input space, make nonlinear transformations of the inputs using a vector of basis functions $\phi(\mathbf{x})$.
  - Linear-separability in feature space does not imply linear-separability in input space

Logistic Regression for the Non-Linear Case

- Hypothesis Representation for Logistic Regression:

$$h_{\mathbf{w}}(\phi) = \sigma(\mathbf{w}^T\phi) = \frac{1}{1 + e^{-\mathbf{w}^T\phi}},$$

  where ϕ is an M-dimensional feature vector

- Hypothesis Output Interpretation:
  - $h_{\mathbf{w}}(\phi) = P(y = 1|\phi, \mathbf{w})$ - the confidence in the predicted label
  - $P(y = 0|\phi, \mathbf{w}) = 1 - P(y = 1|\phi, \mathbf{w})$

Summary

Logistic Regression Model

- Model Representation
- How to Choose a Hypothesis?
- Multiclass Logistic Regression
- Logistic Regression for the Non-Linear Case