University of Amsterdam
Informatics Institute

AMLAB
Amsterdam Machine Learning Lab

Machine Learning 1

Lecture 7.3 - Supervised Learning
Classification - Logistic Regression:
Stochastic Gradient Descent

Erik Bekkers

(Bishop 4.3.2)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Logistic Regression for Two Classes

- Given: Dataset $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^T$ with binary targets
  $\mathbf{t} = (t_1, ..., t_N)^T$ with $t_n \in \{\mathcal{C}_1, \mathcal{C}_2\} = \{1, 0\}$
- Conditional likelihood function:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} p(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n}$$

$$y_n = p(C_1|\phi_n) = \sigma(w^T\phi_n) \qquad \boldsymbol{\phi}_n = \boldsymbol{\phi}(\mathbf{x}_n)$$

- Maximizing the conditional likelihood/minimizing the cross-entropy

$$E(\mathbf{w}) = -\ln p(\mathbf{t}, \mathbf{X}, \mathbf{w}) = -\sum_{n=1}^{N} t_n \ln y_n + (1 - t_n)\ln(1 - y_n)$$

- E(**w**): convex, but no closed form solution!

  $y_n = \sigma(\mathbf{w}^T\boldsymbol{\phi}_n)$ is nonlinear in **w**

Logistic Regression (K=2): SGD

$y_n = \sigma(\underline{w}^T \underline{\phi}_n)$

- Stochastic Gradient Descent for cross-entropy:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} t_n \ln y_n + (1 - t_n) \ln(1 - y_n) = \sum_{n=1}^{N} E_n(\underline{w})$$

- Update rule given a random data point $(\mathbf{x}_n, t_n)$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n(\mathbf{w}^{(\tau)})^T$$

- Gradient: $\nabla E_n(\mathbf{w}) = \left(\frac{\partial E_n(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial E_n(\mathbf{w})}{\partial w_{M-1}} \right)$

- $$\frac{\partial E_n(\mathbf{w})}{\partial w_j} = \frac{\partial E_n(\underline{w})}{\partial y_n} \frac{\partial y_n}{\partial w_j} = \left(-\frac{t_n}{y_n} + \frac{1-t_n}{1-y_n} \right) \cdot \frac{\partial y_n}{\partial w_j}$$

- $$\frac{\partial y_n}{\partial w_j} = \frac{\partial}{\partial w_j} \sigma(\mathbf{w}^T \boldsymbol{\phi}_n)$$

Logistic Regression (K=2): SGD

$$\boxed{\frac{\partial \mathbf{w}^T \boldsymbol{\phi}_n}{\partial w_j} = \frac{\partial}{\partial w_j} \sum_{i=0}^{M-1} w_i \phi_{ni} = \phi_{nj}}$$

- $$\frac{\partial y_n}{\partial w_j} = \frac{\partial}{\partial w_j} \sigma(\mathbf{w}^T \boldsymbol{\phi}_n)$$
- Use $$\frac{\partial \sigma(a)}{\partial a} = \sigma(a)(1 - \sigma(a))$$
- $$\frac{\partial}{\partial w_j} \sigma(\mathbf{w}^T \boldsymbol{\phi}_n) = \overbrace{\sigma(\mathbf{w}^T \boldsymbol{\phi}_n)}^{y_n}(1 - \overbrace{\sigma(\mathbf{w}^T \boldsymbol{\phi}_n)}^{y_n}) \cdot \frac{\partial \mathbf{w}^T \boldsymbol{\phi}_n}{\partial w_j}$$

$$= y_n (1 - y_n) \phi_{nj}$$

- $$\frac{\partial E_n(\mathbf{w})}{\partial w_j} = -\frac{t_n}{y_n} \frac{\partial y_n}{\partial w_j} + \frac{1 - t_n}{1 - y_n} \frac{\partial y_n}{\partial w_j}$$

activation fn = link fn

Bishop 4.3.6

$$= -\frac{t_n}{\cancel{y_n}} \cdot \cancel{y_n} (1 - y_n) \phi_{nj} + \frac{1 - t_n}{\cancel{1 - y_n}} \cdot y_n \cancel{(1 - y_n)} \phi_{nj}$$

$$= -t_n \phi_{nj} + t_n \cancel{y_n} \phi_{nj} + y_n \phi_{nj} - t_n \cancel{y_n} \phi_{nj} = (y_n - t_n) \phi_{nj}$$

Logistic Regression (K=2): SGD

- Stochastic Gradient Descent for cross-entropy:

$$E(\mathbf{w}) = -\sum_{n=1}^{N} t_n \ln y_n + (1 - t_n) \ln(1 - y_n) = \sum_{n=1}^{N} E_n(\mathbf{w})$$

- Update rule given a random data point $(\mathbf{x}_n, t_n)$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n(\mathbf{w}^{(\tau)})^T$$

- $\dfrac{\partial E_n(\mathbf{w})}{\partial w_j} = (y_n - t_n)\phi_j(\mathbf{x}_n)$

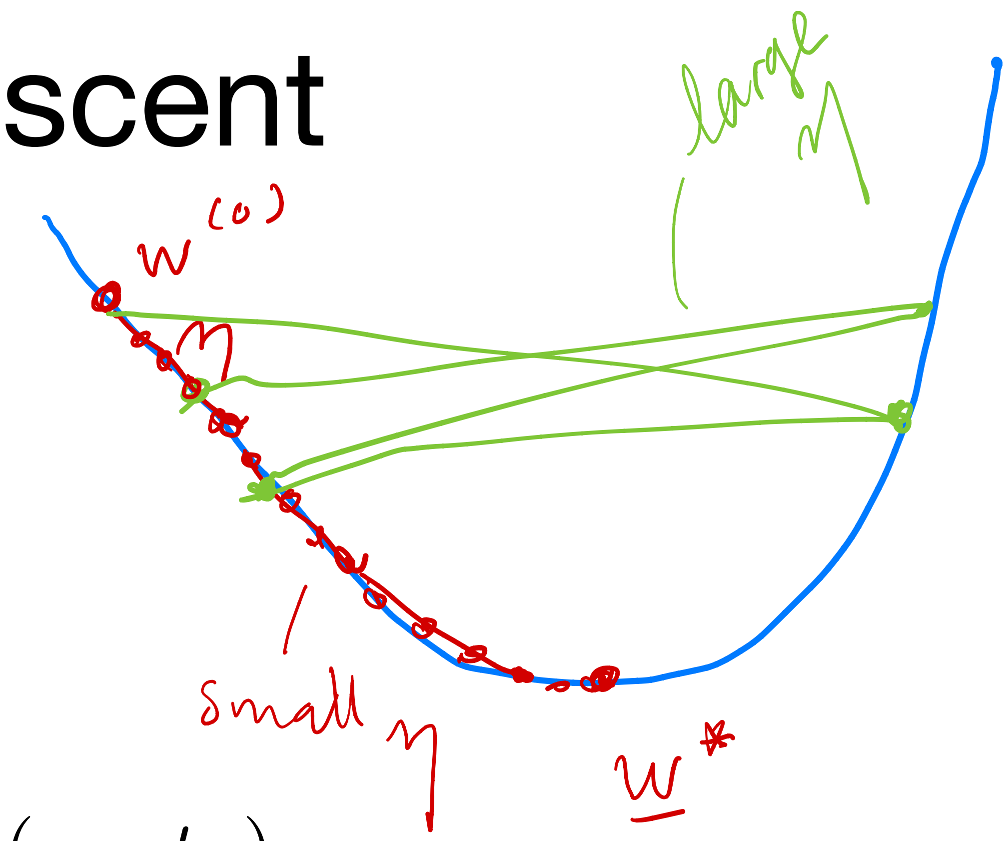
- Gradient: $\nabla E_n(\mathbf{w})^T = \left(\dfrac{\partial E_n(\mathbf{w})}{\partial w_0}, \ldots, \dfrac{\partial E_n(\mathbf{w})}{\partial w_{M-1}} \right)^T = (y_n - t_n)\boldsymbol{\phi}_n$

- Update rule:

$$\underline{\mathbf{w}}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \, (y_n - \underline{t_n}) \underline{\phi_n}$$

perceptron

Stochastic Gradient Descent



1. Initialize $\mathbf{w}^{(0)}$
2. Choose a learning rate η
3. While $||\mathbf{w}^{(\tau+1)} - \mathbf{w}^{(\tau)}|| > \varepsilon$
   I. Choose a random data point $(\mathbf{x}_n, t_n)$
   II. Update **w:**

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \, (y_n^{(\tau)} - t_n)\boldsymbol{\phi}(\mathbf{x}_n)$$

- If η too large: no convergence
- If η too small: very slow convergence
- Converged **w***: estimate of minimizer of E(**w**)!