Classification

Probabilistic Discriminative Models I

T.L. Grobler
(with some edits by S. Kroon)

Classification

- Given **x** assign to one of k classes:
  - $C_j, j = 1,\ldots,k$
  - Assign prob $P(C_j|\mathbf{x})$
  - $C^* = \text{argmax}_{C_j} P(C_j|\mathbf{x})$
- Class prob, more useful than knowing max class prob.

Discriminative Approach

- Dispenses with: $p(\mathbf{x}|C_j)$
- Directly compute posterior $P(C_j|\mathbf{x})$
- Probabilistic Discriminative Models (PDM)

Two Classes: Shared Σ

$$P(C_1|\mathbf{x}) = \sigma(a(\mathbf{x})) = \frac{1}{1+\exp(-a(\mathbf{x}))}$$

$$\begin{aligned} a(\mathbf{x}) &= \ln \frac{p(\mathbf{x}|C_1)P(C_1)}{p(\mathbf{x}|C_2)P(C_2)} \\ &= \mathbf{w}^T\mathbf{x} + w_0 \end{aligned}$$

$$\mathbf{w} = \Sigma^{-1}(\mathbf{u}_1 - \mathbf{u}_2)$$

$$w_0 = -\frac{1}{2}\mathbf{u}_1^T\Sigma^{-1}\mathbf{u}_1 + \frac{1}{2}\mathbf{u}_2^T\Sigma^{-1}\mathbf{u}_2 + \ln\frac{P(C_1)}{P(C_2)}$$

Motivation for PDMs

- Compute Weights Indirectly (PGM)
  - Estimate Gaussian class-conditionals pdfs
  - Shared Σ
  - $a(\mathbf{x})$ is linear
  - Compute weights (decision boundary) from pdfs
- Compute Weights Directly (PDM)
  - $\mathbf{w}$ maps $\mathbf{x}$ to $P(C|\mathbf{x})$
  - No need for class-conditional pdfs
  - Why not directly compute $\mathbf{w}$?
  - Maximize likelihood

Redefine **x** and **w** etc...

$$\mathbf{x} = [1 \; x_1 \; x_2 \; \cdots \; x_d]^T \qquad \mathbf{w} = [w_0 \; w_1 \; \cdots \; w_d]^T$$

$$P(C_1|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x})$$

$$\mathbf{w}^T\mathbf{x} = 0$$

$$X = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \qquad \mathbf{y} = \{y_1, \cdots, y_n\}$$

$$D = [X, \mathbf{y}]$$

Likelihood

$$p(D|\mathbf{w}) = p(X, \mathbf{y}|\mathbf{w})$$

$$= p(\mathbf{y}|X, \mathbf{w})p(X|\mathbf{w})$$

$P(A, B) = P(A|B)P(B)$

Prob Product Rule

Independence x 3

$$= p(X) \prod_{n=1}^{N} p(y_n|X, \mathbf{w})$$

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

Bernoulli ($y_* \varepsilon \{0,1\}$)

$$= p(X) \prod_{n=1}^{N} P(C_1|\mathbf{x}_n, \mathbf{w})^{y_n} (1 - P(C_1|\mathbf{x}_n, \mathbf{w}))^{1-y_n}$$

$$= p(X) \prod_{n=1}^{N} \sigma(\mathbf{w}^T x_n)^{y_n} (1 - \sigma(\mathbf{w}^T x_n))^{1-y_n}$$

Negative log-likelihood

$$E(\mathbf{w}) = -\ln P(D|\mathbf{w}) = -\sum_{n=1}^{N}\{y_n \ln \sigma(\mathbf{w}^T\mathbf{x}_n) + (1-y_n)\ln(1-\sigma(\mathbf{w}^T\mathbf{x}_n))\} - \ln P(X)$$

$$\frac{\partial \sigma}{\partial x} = \sigma(1-\sigma)$$

+Chain Rule

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = -\sum_{n=1}^{N}\left\{y_n \frac{\sigma(1-\sigma)}{\sigma} - (1-y_n)\frac{\sigma(1-\sigma)}{1-\sigma}\right\}\frac{\partial \mathbf{w}^T\mathbf{x}_n}{\partial \mathbf{w}}$$

$$= -\sum_{n=1}\{y_n - y_n\sigma - \sigma + y_n\sigma\}\mathbf{x}_n$$

$$= \sum_{n=1}\{\sigma(\mathbf{w}^T\mathbf{x}_n) - y_n\}\mathbf{x}_n$$

$$\frac{\partial y}{\partial \mathbf{x}} = \left[\frac{\partial y}{\partial x_1} \quad \cdots \quad \frac{\partial y}{\partial x_d}\right]$$

scalar-by-vector notation

Autor: 01.03.22

Minimizing

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \boldsymbol{\Delta} E(\mathbf{w}) = \mathbf{0}$$

Newton-Raphson

Building Intiution

- Samples
  - From both classes
    - Achieve Maximum Seperation
  - Far from decision boundary
    - $P(C_1|\mathbf{x}_n,\mathbf{w}) \approx \sigma(\mathbf{w}^T\mathbf{x}_n) \approx 1$
    - $\sigma(\mathbf{w}^T\mathbf{x}_n) - 1 \approx 0$
    - Little influence
  - Close to decision boundary
    - Large influence

Problems

- Weights become too large
  - $P(C_1|\mathbf{x}_n,\mathbf{w}) \rightarrow 1$
  - $\mathbf{w}^T\mathbf{x}_n \rightarrow \infty$
- Overfitting
  - Samples at boundary have large influence
  - Boundary from training data
    - Fails to generalize
    - Too specific

Solutions

- Constrained optimization
  - $\mathbf{w}^T\mathbf{w} = 1$
- Add penalty term
  - regularization