Logistic Regression

- Two classes $+1, -1$
- Model $p(y|\underline{x})$ (rather than $p(\underline{x}|y)$ as in LDA)

discriminative v. generative

$$p(1|\underline{x}) = \frac{\exp(\beta_0 + \underline{\beta}^T \underline{x})}{1 + \exp(\beta_0 + \underline{\beta}^T \underline{x})} \qquad \underline{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{bmatrix}$$

$$p(-1|\underline{x}) = \frac{1}{1 + \exp(\beta_0 + \underline{\beta}^T \underline{x})} = 1 - p(1|\underline{x})$$

$g(t) = \frac{e^t}{1+e^t}$ is "logistic" function

Estimating Parameters

Given training data $\mathcal{T} = \{(\underline{x}_i, y_i)\}_{i=1}^N$

find β_0 and $\underline{\beta}$ that maximize

$$h(\beta_0, \underline{\beta}) = \prod_{i=1}^N p(y_i|\underline{x}_i)$$

Maximum likelihood estimation – done using iterative optimization methods

Logistic Regression is a linear classifier:

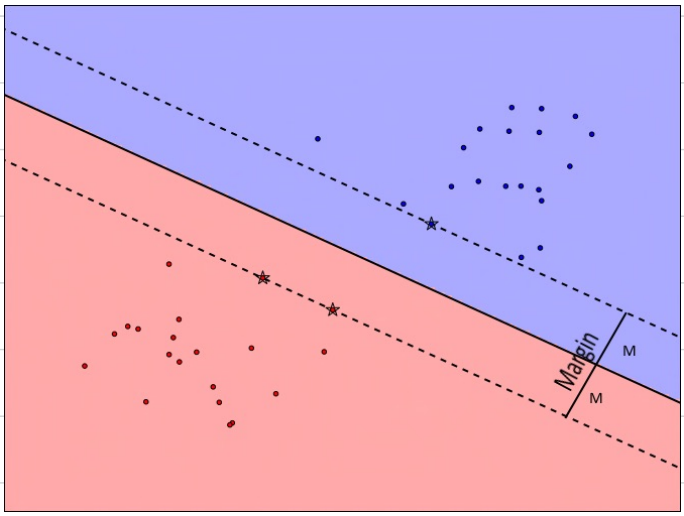
$$p(1|\underline{x}) \underset{-1}{\overset{1}{\gtrless}} p(-1|\underline{x}) \equiv \ln p(1|\underline{x}) \underset{-1}{\overset{1}{\gtrless}} \ln p(-1|\underline{x})$$

$$\equiv \hat{\beta}_0 + \hat{\underline{\beta}}^T \underline{x} \underset{-1}{\overset{1}{\gtrless}} 0$$

linear discriminant function

Support Vector Machine (SVM)

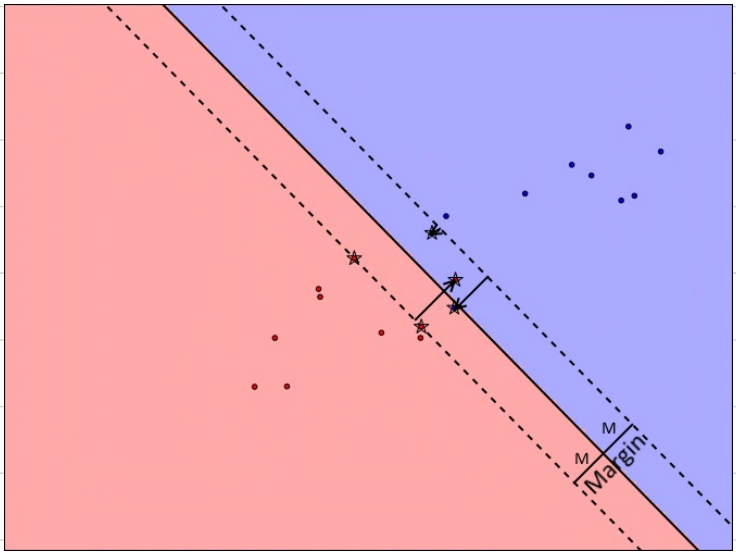
Linearly Separable Training Data



Infinitely many linear boundaries separating classes

SVM maximizes margin M between classes

Training Data Not Linearly Separable



1. Fix decision boundary and margin m
2. Compute distances of points on "wrong" side of margin to margin boundary ξ_i^*
3. Compute penalty $\xi_i = \frac{\xi_i^*}{m}$
4. Maximize m subject to $\sum_i \xi_i \leq C$ to get M
5. Choose boundary to maximize M

Naive Bayes Classifier

- Recall $\hat{y}_{Bayes} = \arg\max_{y=1,\dots,M} \pi_y \, p(\underline{x} \mid y)$

  Feature vector $\underline{x} = [x_1 \; x_2 \; \cdots \; x_d]^T$

- Naive Bayes assumes that $x_1, x_2, \dots, x_d$ are independent, conditioned on label y, i.e.

  $$p(\underline{x} \mid y) = p(x_1 \mid y) \cdot p(x_2 \mid y) \cdots p(x_d \mid y)$$

- Get estimates $\hat{\pi}_y$, $\hat{p}(x_1 \mid y)$, $\hat{p}(x_2 \mid y) \cdots \hat{p}(x_d \mid y)$ for each $y = 1, 2, \dots, M$. Then,

  $$\hat{y}_{NB} = \arg\max_{y=1,2,\dots,M} \hat{\pi}_y \prod_{j=1}^{d} \hat{p}(x_j \mid y)$$

  $$= \arg\max_{y=1,2,\dots,M} \ln \hat{\pi}_y + \sum_{j=1}^{d} \ln \hat{p}(x_j \mid y)$$

Pros and Cons of NB Classifier

+ estimating $p(\underline{x} \mid y)$ much harder than estimating $p(x_1 \mid y), p(x_2 \mid y), \dots p(x_d \mid y)$ separately

+ Can mix different kinds of features: continuous real-valued, discrete, categorical

− independence assumption is simplistic

Finding $\hat{p}(x_j|y)$ using training data

Convenient to use parametric model:

$$p(x_j|y) = p(x_j|y, \theta_{jy})$$

Then, $\hat{p}(x_j|y) = p(x_j|y, \hat{\theta}_{jy})$

Example (continuous x_j)

$$p(x_j|y, \theta_{jy}) = \frac{1}{\sqrt{2\pi\sigma_{jy}^2}} e^{-\frac{(x_j-\mu_{jy})^2}{2\sigma_{jy}^2}} \leftarrow N(\mu_{jy}, \sigma_{jy}^2)$$

$$\theta_{jy} = (\mu_{jy}, \sigma_{jy}^2)$$

Given training data $\{(x_i, y_i)\}_{i=1}^N$,

$$\hat{\mu}_{jy} = \frac{\sum_{i:y_i=y} x_{ij}}{N_y}, \quad \hat{\sigma}_{jy}^2 = \frac{\sum_{i:y_i=y} (x_{ij}-\hat{\mu}_{jy})^2}{N_y \leftarrow \text{or } N_y - 1}$$

Example (discrete or categorical x_j)

x_j takes values in set $\{a_1, a_2, \ldots, a_\ell\}$ with probabilities $s_{1,jy}, s_{2,jy}, \ldots s_{\ell jy}$, for class y.

$$\theta_{jy} = (s_{1,jy}, s_{2,jy}, \ldots s_{\ell,jy}), \quad \hat{\theta}_{jy} = (\hat{s}_{1,jy}, \hat{s}_{2,jy}, \ldots \hat{s}_{\ell,jy})$$

$$\hat{s}_{k,jy} = \frac{\#\text{ training data in class } y \text{ with } x_{ij} = a_k}{N_y}$$

(see typed notes for Laplacian smoothing so $\hat{s} \neq 0$)