Machine Learning

Regression-Based Classification & Gaussian Discriminant Analysis

Logistic Regression

- Linear regression provides a nice representation and an efficient solution to a regression problem
  - Can we apply this representation to classification ?
    - Using linear regression directly leads to too many output values that do not match any class well
    - Logistic regression tries to address this by applying a logistic function in order to achieve outputs that are closer to class values

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

Logistic Regression

- For two classes we can map the class labels to 1 and 0, respectively
  - Using this we can interpret the output as the probability to belong to a given class

    $$P_\theta(y=1 \mid x) = h_\theta(x)$$

    $$P_\theta(y=0 \mid x) = 1 - h_\theta(x)$$

    - Simplified this gives

      $$p_\theta(y \mid x) = h_\theta(x)^y \left(1 - h_\theta(x)\right)^{1-y}$$

Logistic Regression

- This gives the likelihood of the parameters as
$$L(\theta) = p(\theta \mid D) = \prod_{i=1}^{n} p_\theta(y^{(i)} \mid x^{(i)})$$
$$= \prod_{i=1}^{n} h_\theta(x^{(i)})^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{1-y^{(i)}}$$

- Converted to log likelihood
$$\log L(\theta) = \log\left(\prod_{i=1}^{n} h_\theta(x^{(i)})^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{1-y^{(i)}}\right)$$
$$= \sum_{i=1}^{n} \log\left(h_\theta(x^{(i)})^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{1-y^{(i)}}\right)$$

Logistic Regression

- Solving for the maximum likelihood optimization using stochastic gradient descent

$$\frac{\partial}{\partial\theta_j}\log L(\theta) = \frac{\partial}{\partial\theta_j}\log\left(h_\theta(x)^y (1-h_\theta(x))^{1-y}\right)$$

$$= \left(y\frac{1}{h_\theta(x)} - (1-y)\frac{1}{1-h_\theta(x)}\right)\frac{\partial}{\partial\theta_j}h_\theta(x)$$

$$= \left(y\frac{1}{h_\theta(x)} - (1-y)\frac{1}{1-h_\theta(x)}\right)g(\theta^T x)(1-g(\theta^T x))\frac{\partial}{\partial\theta_j}\theta^T x$$

$$= \left(y(1-g(\theta^T x)) - (1-y)\,g(\theta^T x)\right)x_j$$

$$= \left(y - g(\theta^T x)\right)x_j = \left(y - h_\theta(x)\right)x_j$$

Logistic Regression

- Gives us a stochastic gradient descent learning method for logistic regression

$$\theta := \theta + \alpha \left(y - h_\theta(x) \right) x^{(i)}$$

  - This allows us to use regression for classification problems

Softmax Regression

- Logistic regression allows us to address a classification problem with two classes
  - How can we address classification with more than 2 classes ?
    - Multiclass classifier – need to compute a probability for each of the classes
      - Corresponds to multinomial distribution

$$h_\theta(x)_j = g_j(\theta^T x) = \frac{e^{\theta_j^T x}}{\sum_k e^{\theta_k^T x}}$$

Softmax Regression

- For each class we can interpret the output as the probability to belong to a given class

$$P_\theta(y=k \mid x) = h_\theta(x)_k$$

  - Simplified this gives

$$p_\theta(y \mid x) = \prod_k h_\theta(x)_k^{\delta_{k,y}}$$

- This gives the likelihood of the parameters as

$$L(\theta) = \prod_i p_\theta(y^{(i)} \mid x^{(i)})$$

Softmax Regression

- Converted to log likelihood

$$\log L(\theta) = \sum_{i-1}^{n} \log p_\theta(y^{(i)} \mid x^{(i)}) = \sum_{i-1}^{n} \log \prod_k h_\theta(x^{(i)})_k^{\delta_{k,y^{(i)}}}$$

$$= \sum_{i-1}^{n} \log \prod_k \left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_k e^{\theta_k^T x^{(i)}}} \right)^{\delta_{k,y^{(i)}}}$$

- Derivative is slightly more complex
  - Can be maximized using gradient ascent

Generative Approaches

- Logistic and softmax regression are discriminative approaches to classification using a linear parameter representation
  - Can we build generative algorithms that provide similar classification ?
    - To build a generative algorithms we need to be able to predict a probability density for the input for a class
      - Gaussian Naïve Bayes made the restrictive assumption that all dimensions are independent
      - We can relax this and assume a multivariate Gaussian to give us Gaussian Discriminant Analysis

Gaussian Discriminant Analysis

- GDA assumes that the probability density function for the data in a class follows a multivariate Gaussian

$$p(x \mid y) = p_{\mu_y,\Sigma_y}(x \mid y) = N(x; \mu_y, \Sigma_y) = \frac{1}{(2\pi)^{\frac{m}{2}} \left|\Sigma_y\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_y)^T \Sigma_y^{-1} (x-\mu_y)}$$

  - Prior is assumed to be from a Bernoulli distribution

$$p(y) = \phi^y (1-\phi)^{1-y}$$

  - Using Bayes law a log likelihood function for the parameters can be derived

$$L(\phi, \mu_0, \Sigma_0, \mu_1, \Sigma_1) = \log \prod_i p_{\mu_{y^{(i)}},\Sigma_{y^{(i)}}}(x^{(i)} \mid y^{(i)})\, p(y^{(i)})$$

Gaussian Discriminant Analysis

- Learning the parameter takes again the form of finding maximum likelihood parameters given the data

$$\phi = \frac{\#(x^{(i)}, y^{(i)}) : y^{(i)} = 1}{\#(x^{(i)}, y^{(i)})}$$

$$\mu_y = \frac{\sum_{i:y^{(i)}=y} x^{(i)}}{\#(x^{(i)}, y^{(i)}) : y^{(i)} = y}$$

$$\Sigma_y = \frac{\sum_{i:y^{(i)}=y} (x^{(i)} - \mu_y)(x^{(i)} - \mu_y)^T}{\#(x^{(i)}, y^{(i)}) : y^{(i)} = y}$$

Gaussian Discriminant Analysis

- The log likelihood of a class can be written as

$$\log\left(p_{\phi,\mu_y,\Sigma_y}(y\,|\,x)\right)=\log\left(\frac{p_{\mu_y,\Sigma_y}(x\,|\,y)p(y)}{\alpha_y}\right)$$

$$=\log\left(\frac{1}{(2\pi)^{\frac{m}{2}}\left|\Sigma_y\right|^{\frac{1}{2}}}e^{-\frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y)}\right)+\log\left(\phi^y(1-\phi)^{1-y}\right)-\log\left(\alpha_y\right)$$

$$=-\frac{1}{2}(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y)-\log(2\pi)^{\frac{m}{2}}-\frac{1}{2}\log\left|\Sigma_y\right|+y\log\phi+(1-y)\log(1-\phi)-\log\alpha_y$$

- This forms a decision boundary

$$\log\left(p_{\phi,\mu_0,\Sigma_0}(y=0\,|\,x)\right)-\log\left(p_{\phi,\mu_1,\Sigma_1}(y=1\,|\,x)\right)<0$$

$$\rightarrow (x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)+\log\left|\Sigma_0\right|-(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)-\log\left|\Sigma_1\right|>T$$

Linear Discriminant Analysis

- If we make the homoscedastic assumption, i.e. if we assume that the (co)variances of the two classes are identical this results in linear discriminant analysis (LDA)

$$\log L(\phi, \mu_0, \mu_1, \Sigma) = \alpha + \log \prod_i p_{\mu_{y^{(i)}}, \Sigma}(x^{(i)} \mid y^{(i)}) p(y^{(i)})$$

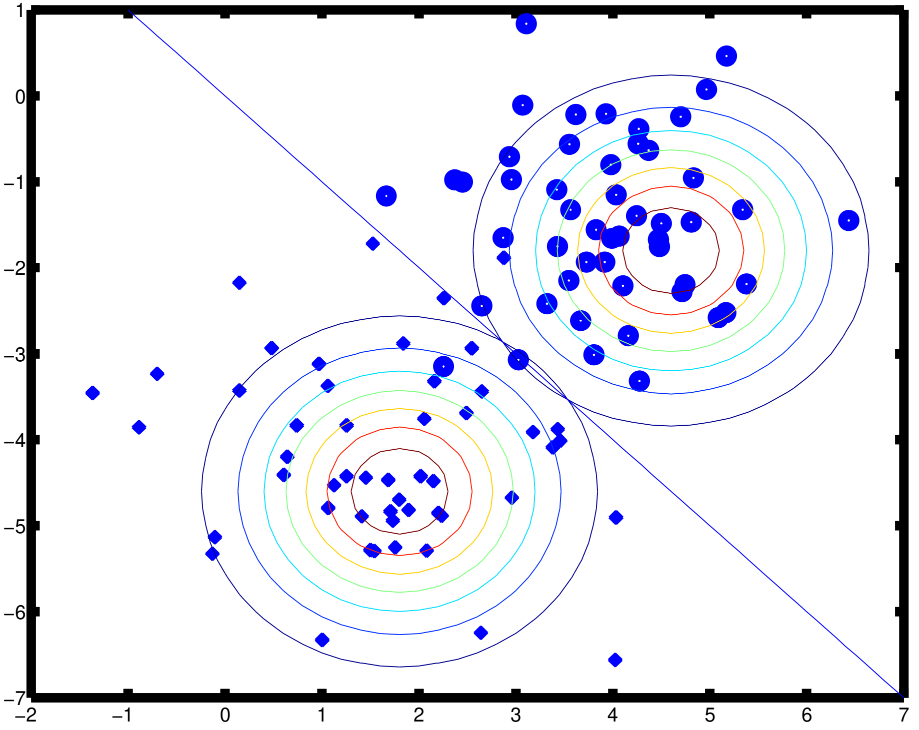
  - Parameter maximization for the (co)variance is

$$\Sigma = \frac{\sum_i (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T}{\#(x^{(i)}, y^{(i)})}$$

    - More stable for small amounts of data

Linear Discriminant Analysis

- With equal (co)variances the discriminant surface becomes linear



Gaussian Discriminant Analysis

- In the decision boundary linearity arises as

$$\log\left(p_{\phi,\mu_0,\Sigma}(y=0\,|\,x)\right)-\log\left(p_{\phi,\mu_1,\Sigma}(y=1\,|\,x)\right)<0$$

$$\rightarrow (x-\mu_0)^T\Sigma^{-1}(x-\mu_0)+\log|\Sigma|-(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)-\log|\Sigma|>T$$

$$\rightarrow (x-\mu_0)^T\Sigma^{-1}(x-\mu_0)-(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)>T$$

$$\rightarrow -2\mu_0^T\Sigma^{-1}x+2\mu_1^T\Sigma^{-1}x>T-\mu_0^T\Sigma^{-1}\mu_0+\mu_1^T\Sigma^{-1}\mu_1$$

$$\rightarrow (\mu_1-\mu_0)\Sigma^{-1}x>\frac{1}{2}\left(T-\mu_0^T\Sigma^{-1}\mu_0+\mu_1^T\Sigma^{-1}\mu_1\right)$$

Linear Discriminant Analysis

- In the case of linear discriminant analysis the class likelihood can be rewritten as

$$p_{\phi,\mu_0,\mu_1,\Sigma}(y=1\,|\,x) = \frac{1}{1+e^{-\theta(\phi,\mu_y,\Sigma)^T x}}$$

  - Solution to linear discriminant analysis has similar form as the representation for logistic regression
    - LDA is more restrictive (only allows some types of θ)
      - If the distributions are Gaussian with same (co)variance then LDA will find the solution more efficiently and usually more precisely
      - Logistic regression is more robust (it covers more than just Gaussian distributions and less sensitive to modeling errors

Quadratic Discriminant Analysis

- If we do not make the homoscedastic assumption, the discrimination surface becomes quadratic resulting in Quadratic Discriminant Analysis (QDA)

$$\log\left(p_{\phi,\mu_0,\Sigma_0}(y=0\,|\,x)\right)-\log\left(p_{\phi,\mu_1,\Sigma_1}(y=1\,|\,x)\right)<0$$

$$\rightarrow (x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)+\log|\Sigma_0|-(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)-\log|\Sigma_1|>T$$

  - QDA is more flexible but also more complex
    - Requires more data to train

Logistic Regression and Linear Discriminant Analysis

- Logistic regression and Linear discriminant analysis are frequently used for classification
  - Logistic regression provides an effective discriminative classification approach
    - Same learning rule as for linear regression
  - LDA provides generative classification that, if its assumptions are met, solves the same problem
  - Softmax regression generalizes logistic regression
    - Harder to train
  - QDA provides a more general generative classifier
    - Requires more data