Machine Learning

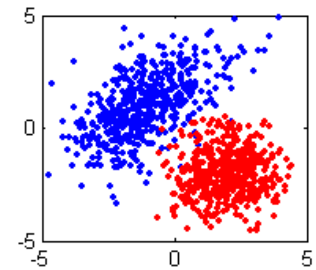
4771

Instructor: Tony Jebara

Topic 10

- Classification with Gaussians
- Regression with Gaussians
- Principal Components Analysis

Classification with Gaussians



- Have two classes, each with their own Gaussian:

$$\left\{\left(x_1,y_1\right),\ldots,\left(x_N,y_N\right)\right\} \quad x \in R^D \;\; y \in \left\{0,1\right\}$$

- Given parameters $\theta = \left\{\alpha, \mu_0, \Sigma_0, \mu_1, \Sigma_1\right\}$ we can generate iid data from $p\left(x,y \mid \theta\right) = p\left(y \mid \theta\right) p\left(x \mid y,\theta\right)$ by:

  1) flipping a coin to get y via Bernoulli $p\left(y \mid \theta\right) = \alpha^y \left(1-\alpha\right)^{1-y}$

  2) sampling an x from y'th Gaussian $p\left(x \mid y,\theta\right) = N\left(x \mid \mu_y, \Sigma_y\right)$

- Or, recover parameters from data using maximum likelihood

$$\begin{aligned} l\left(\theta\right) &= \log p\left(data \mid \theta\right) = \sum\nolimits_{i=1}^N \log p\left(x_i, y_i \mid \theta\right) \\ &= \sum\nolimits_{i=1}^N \log p\left(y_i \mid \theta\right) + \sum\nolimits_{i=1}^N \log p\left(x_i \mid y_i, \theta\right) \\ &= \sum\nolimits_{i=1}^N \log p\left(y_i \mid \alpha\right) + \sum\nolimits_{y_i \in 0} \log p\left(x_i \mid \mu_0, \Sigma_0\right) + \sum\nolimits_{y_i \in 1} \log p\left(x_i \mid \mu_1, \Sigma_1\right) \end{aligned}$$

Classification with Gaussians

- Max Likelihood can be done separately for the 3 terms

$$l = \sum_{i=1}^{N} \log p\left(y_i \mid \alpha\right) + \sum_{y_i \in 0} \log p\left(x_i \mid \mu_0, \Sigma_0\right) + \sum_{y_i \in 1} \log p\left(x_i \mid \mu_1, \Sigma_1\right)$$

- Count # of pos & neg examples (class prior): $\alpha = \frac{N_1}{N_0 + N_1}$
- Get mean & cov of negatives and mean & cov of positives:

$$\mu_0 = \frac{1}{N_0} \sum_{y_{i \in 0}} x_i \qquad \Sigma_0 = \frac{1}{N_0} \sum_{y_{i \in 0}} \left(x_i - \mu_0\right)\left(x_i - \mu_0\right)^T$$

$$\mu_1 = \frac{1}{N_1} \sum_{y_{i \in 1}} x_i \qquad \Sigma_1 = \frac{1}{N_1} \sum_{y_{i \in 1}} \left(x_i - \mu_1\right)\left(x_i - \mu_1\right)^T$$

- Given (x,y) pair, can now compute likelihood $p\left(x, y\right)$
- To make classification, a bit of Decision Theory
- Without x, can compute prior guess for y $p\left(y\right)$
- Give me x, want y, I need posterior $p\left(y \mid x\right)$
- Bayes Optimal Decision: $\hat{y} = \arg\max_{y=\{0,1\}} p\left(y \mid x\right)$
- Optimal iff we have true probability

Posterior gives Logistic

- Bayes Optimal Decision: $\hat{y} = \arg\max_{y=\{0,1\}} p(y \mid x)$
- To get conditional:

$$p(y \mid x) = \frac{p(x,y)}{p(x)} = \frac{p(x,y)}{\sum_y p(x,y)} = \frac{p(x,y)}{p(x,y=0) + p(x,y=1)}$$

- Check which is greater: $p(y=0 \mid x) \geq ? \leq p(y=1 \mid x)$
- Or check if this is > 0.5

$$p(y=1 \mid x) = \frac{p(x,y=1)}{p(x,y=0) + p(x,y=1)}$$

$$= \frac{1}{\frac{p(x,y=0)}{p(x,y=1)} + 1}$$

$$= \frac{1}{\exp\left(-\log \frac{p(x,y=1)}{p(x,y=0)}\right) + 1}$$

- Get logistic squashing function of log-ratio of probability models

$$= sigmoid\left(\log \frac{p(x,y=1)}{p(x,y=0)}\right)$$

Linear or Quadratic Decisions

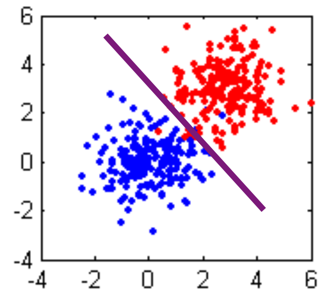
- Example cases, plotting decision boundary when = 0.5

$$p\left(y=1 \mid x\right) = \frac{p\left(x,y=1\right)}{p\left(x,y=0\right)+p\left(x,y=1\right)}$$

$$= \frac{\alpha N\left(x \mid \mu_1,\Sigma_1\right)}{\left(1-\alpha\right)N\left(x \mid \mu_0,\Sigma_0\right)+\alpha N\left(x \mid \mu_1,\Sigma_1\right)}$$

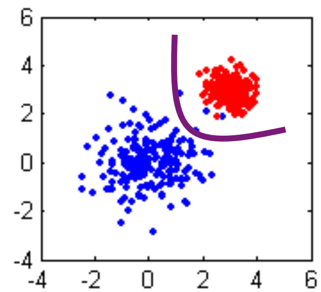
- If covariances are equal:

  linear decision



- If covariances are different:

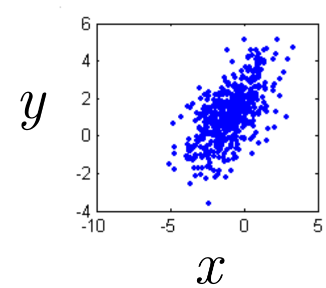
  quadratic decision



Regression with Gaussians

- Have input and output, each Gaussian:

$$\left\{\left(x_1, y_1\right), \ldots, \left(x_N, y_N\right)\right\} \quad x \in R^{D_x} \;\; y \in R^{D_y}$$



concatenate $z_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$

$$p\left(z \mid \mu, \Sigma\right) = \frac{1}{\left(2\pi\right)^{D/2}\sqrt{\left|\Sigma\right|}} \exp\left(-\tfrac{1}{2}\left(z-\mu\right)^T \Sigma^{-1}\left(z-\mu\right)\right)$$

- Maximum Likelihood is as usual for a multivariate Gaussian

$$\mu = \tfrac{1}{N}\sum_{i=1}^{N} z_i \qquad \Sigma = \tfrac{1}{N}\sum_{i=1}^{N}\left(z_i-\mu\right)\left(z_i-\mu\right)^T$$

- Bayes optimal decision: $\hat{y} = \arg\max_{y \in \mathbb{R}} p\left(y \mid x\right)$
- Or we can use: $\hat{y} = E_{p(y|x)}\left\{y\right\}$
- Have joint, need conditional: $p\left(y \mid x\right) = \frac{p\left(x,y\right)}{p\left(x\right)} = \frac{p\left(x,y\right)}{\int_y p\left(x,y\right)}$

Gaussian Marginals/Conditionals

- Conditional & marginal from joint: $p(y \mid x) = \frac{p(x,y)}{p(x)} = \frac{p(x,y)}{\int_y p(x,y)}$

- Conditioning the Gaussian:

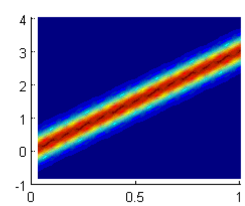
$$p(z \mid \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma|}} \exp\left(-\tfrac{1}{2}(z-\mu)^T \Sigma^{-1} (z-\mu)\right)$$

$$p(x,y) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma|}} \exp\left(-\tfrac{1}{2}\left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}\right)^T \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}^{-1} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}\right)\right)$$

$$p(x) = \frac{1}{(2\pi)^{D_x/2}\sqrt{|\Sigma_{xx}|}} \exp\left(-\tfrac{1}{2}(x-\mu_x)^T \Sigma_{xx}^{-1} (x-\mu_x)\right)$$

$$= N(x \mid \mu_x, \Sigma_{xx})$$

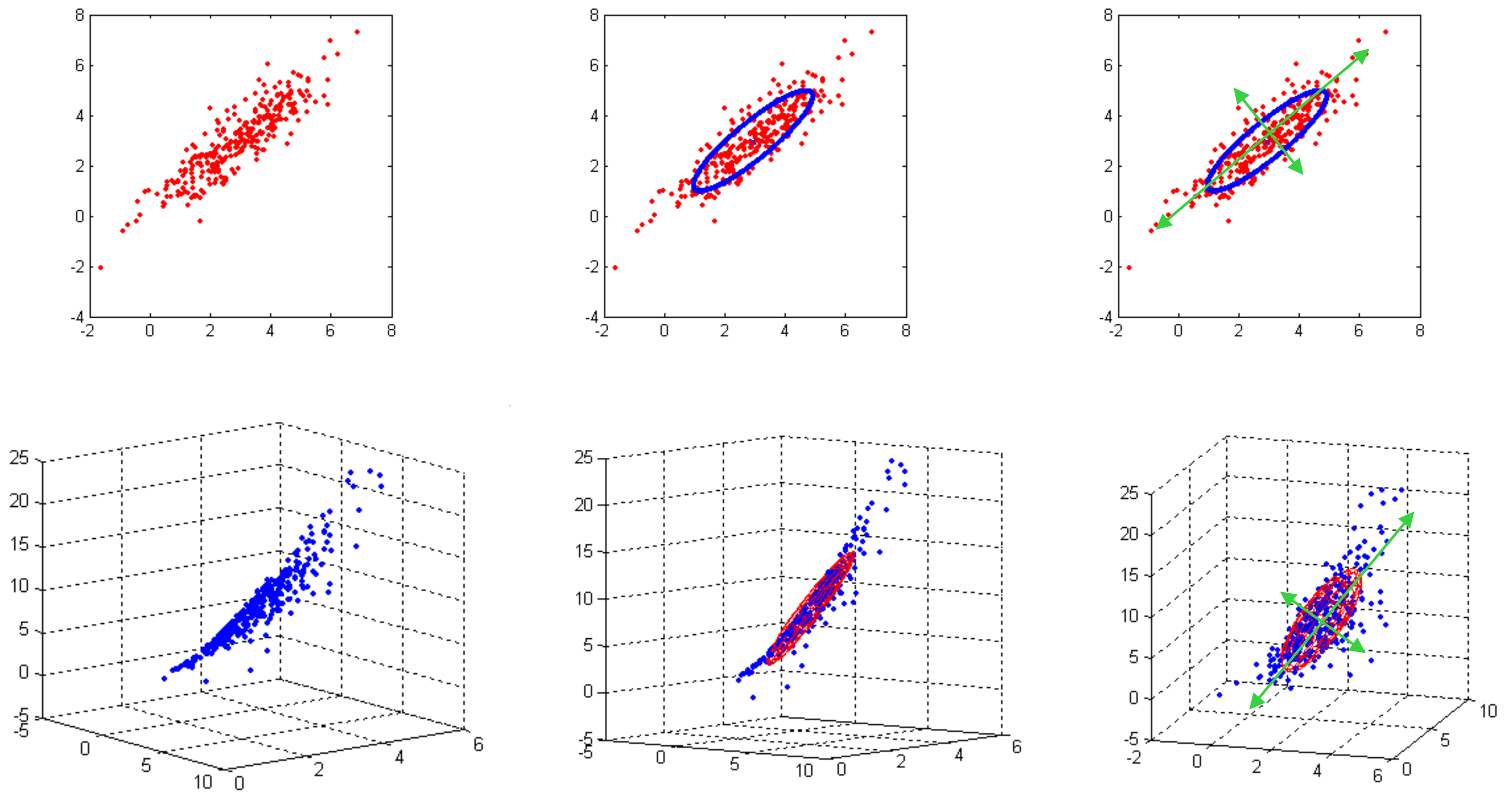
$$p(y \mid x) = N\left(y \mid \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}(x-\mu_x), \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}\right)$$



- Here argmax is expectation which is conditional mean: $\hat{y} = \mu_y + \Sigma_{yx}\Sigma_{xx}^{-1}(x-\mu_x)$

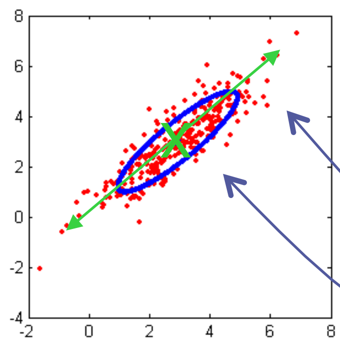
Principal Components Analysis

- Gaussians: for Classification, Regression… & Compression!
- Data can be constant in some directions, changes in others
- Use Gaussian to find directions of high/low variance

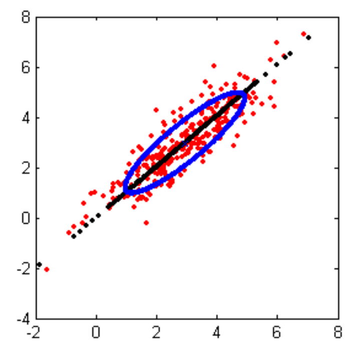


Principal Components Analysis

- Idea: instead of writing data in all its dimensions, only write it as mean + steps along one direction

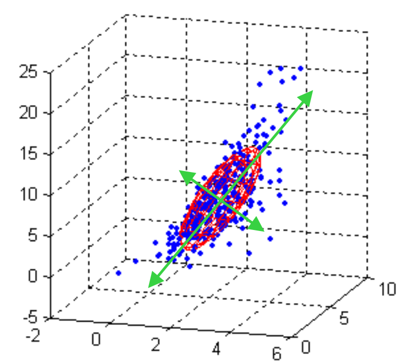


$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \approx \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} + c_i \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$



- More generally, keep a subset of dimensions C from D (i.e. 2 of 3)

$$\vec{x}_i \approx \vec{\mu} + \sum_{j=1}^{C} c_{ij} \vec{v}_j$$



- Compression method: $\vec{x}_i \gg \vec{c}_i$
- Optimal directions: along eigenvectors of covariance
- Which directions to keep: highest eigenvalues (variances)

Principal Components Analysis

- If we have eigenvectors, mean and coefficients:

$$\vec{x}_i \approx \vec{\mu} + \sum_{j=1}^{C} c_{ij} \vec{v}_j$$

- Get eigenvectors (use eig() in Matlab): $\Sigma = V \Lambda V^T$

$$\begin{bmatrix} \Sigma(1,1) & \Sigma(1,2) & \Sigma(1,3) \\ \Sigma(1,2) & \Sigma(2,2) & \Sigma(2,3) \\ \Sigma(1,3) & \Sigma(2,3) & \Sigma(3,3) \end{bmatrix} = \begin{bmatrix} [\vec{v}_1] & [\vec{v}_2] & [\vec{v}_3] \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} [\vec{v}_1] & [\vec{v}_2] & [\vec{v}_3] \end{bmatrix}^T$$

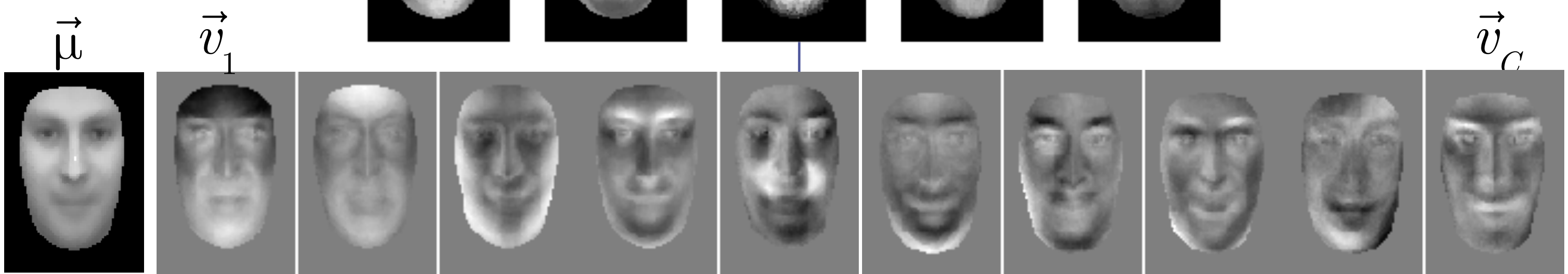
- Eigenvectors are orthonormal: $\vec{v}_i^T \vec{v}_j = \delta_{ij}$
- In coordinates of v, Gaussian is diagonal, cov = Λ
- All eigenvalues are non-negative $\lambda_i \geq 0$
- Higher eigenvalues are higher variance, use the top C ones

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \geq \ldots$$

- To compute the coefficients: $c_{ij} = \left(\vec{x}_i - \vec{\mu}\right)^T \vec{v}_j$

Eigenfaces

$$\left\{x_1,\ldots,x_N\right\}$$



ENCODE

$$\boxed{c_{ij} = \left(\vec{x}_i - \vec{\mu}\right)^T \vec{v}_j}$$

DECODE

$$\left\{\left(\hat{x}_1 = \mu + \sum_{j=1}^{C} c_{1j}\vec{v}_j\right),\ldots,\left(\hat{x}_N = \mu + \sum_{j=1}^{C} c_{Nj}\vec{v}_j\right)\right\}$$