AI – Machine Learning

Artificial Intelligence
Research Group

Linear methods for classification

- These decision boundaries are linear; this is what we will mean by linear methods for classification.

$$\Pr(G=1 \mid X=x) = \frac{\exp(\beta_0+\beta^T x)}{1+\exp(\beta_0+\beta^T x)}$$

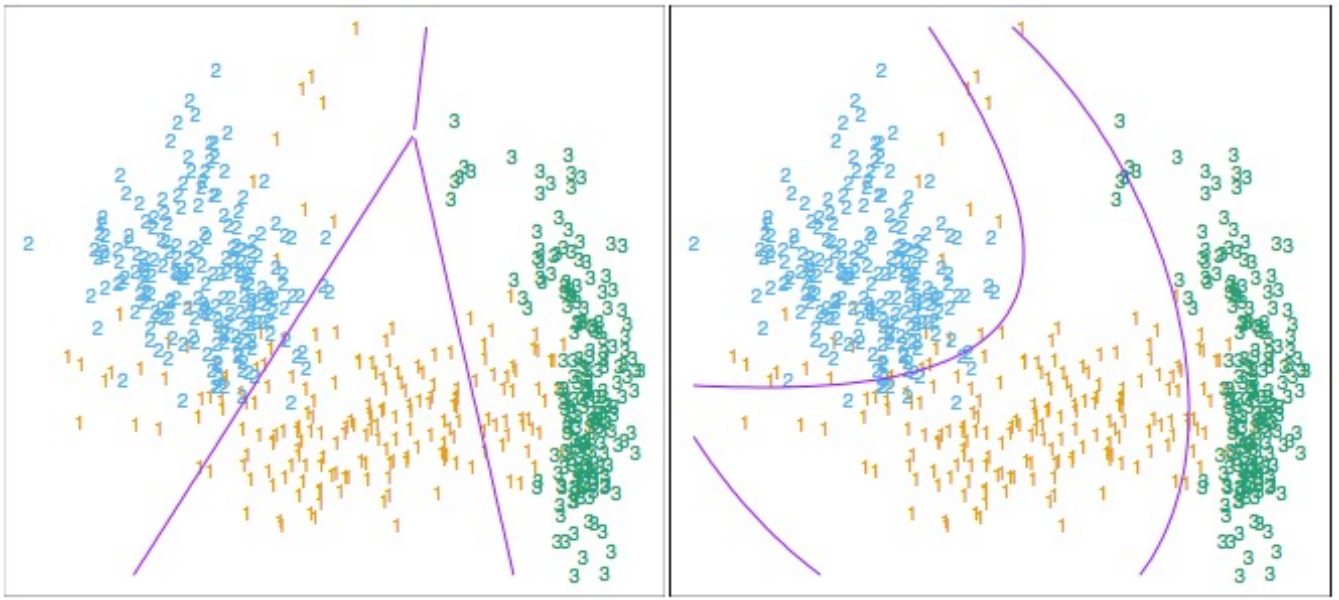
$$\Pr(G=2 \mid X=x) = \frac{1}{1+\exp(\beta_0+\beta^T x)}$$

Log-odds:

$$\log \frac{\Pr(G=1 \mid X=x)}{\Pr(G=2 \mid X=x)} = \beta_0+\beta^T x$$

Basis transformation

- We can expand our variable set $X_1, \ldots, X_p$ by including their squares and cross products
- This approach can be used with any basis transformation $h(X)$ where $h : \mathbb{R}^p \rightarrow \mathbb{R}^q, q > p$



Linear discriminant analysis

Suppose $f_k(x)$ is the ***class-conditional density*** of X in class $G = k$, and let π_k be the ***prior probability*** of class k, with $\sum_{k=1}^{K} \pi_k = 1$

$$Pr(G = k \mid X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

Suppose that we model each class density as ***multivariate Gaussian***

$$f_k(x) = \frac{1}{(2\pi)^{p/2} \mid \Sigma_k \mid^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}$$

Linear discriminant analysis

Linear discriminant analysis (LDA) arises in the special case when we assume that the classes have a common covariance matrix $\Sigma_k = \Sigma \forall k$

In comparing two classes k and l, it is sufficient to look at the log-ratio

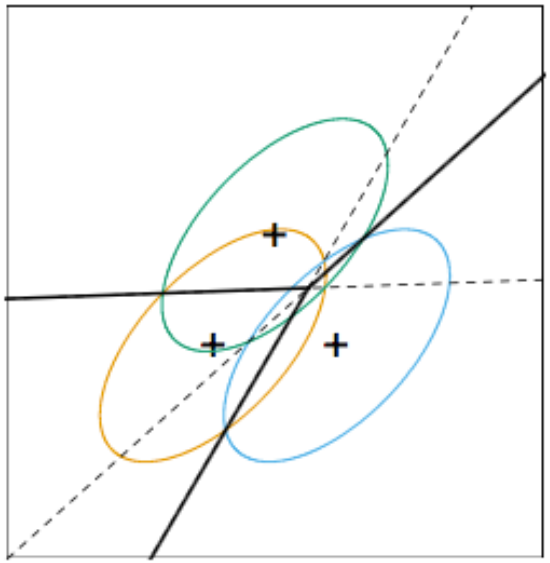
$$\log \frac{\Pr(G=k \mid X=x)}{\Pr(G=l \mid X=x)} = \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l}$$

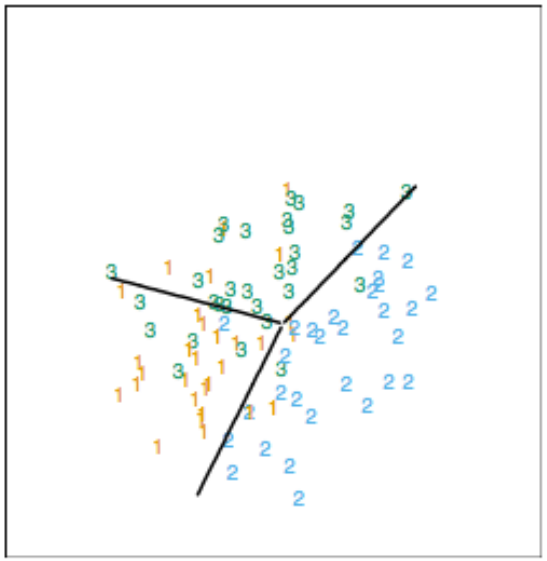
$$= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)^T \Sigma^{-1}(\mu_k - \mu_l) + x^T \Sigma^{-1}(\mu_k - \mu_l)$$

An equation linear in x. We see the linear discriminant functions

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

Linear discriminant analysis





$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

$$\hat{\pi}_k = N_k / N$$

$$\hat{\mu}_k = \sum_{g_i = k} x_i / N_k$$

$$\hat{\Sigma} = \sum_{k=1}^{K} \sum_{g_i = k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T / (N - K)$$

Quadratic discriminant function

If the Σ_k are not assumed to be equal, we then get ***quadratic discriminant function*** (QDA)

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log\pi_k$$

The decision boundary between each pair of classes k and l is described by a quadratic equation $\{x: \delta_k(x) = \delta_l(x)\}$.

The estimates for QDA are similar to those for LDA, except that separate covariance matrices must be estimated for each class. When p is large this can mean a dramatic increase in parameters.

In the STATLOG project LDA was among the top three classifiers for 7 of the 22 datasets, QDA among the top three for four datasets, and one of the pair were in the top three for 10 datasets

Linear discriminant analysis

Fisher's problem therefore amounts to maximizing the Rayleigh quotient,

$$\max_a \frac{a^T \mathbf{B} a}{a^T \mathbf{W} a}$$

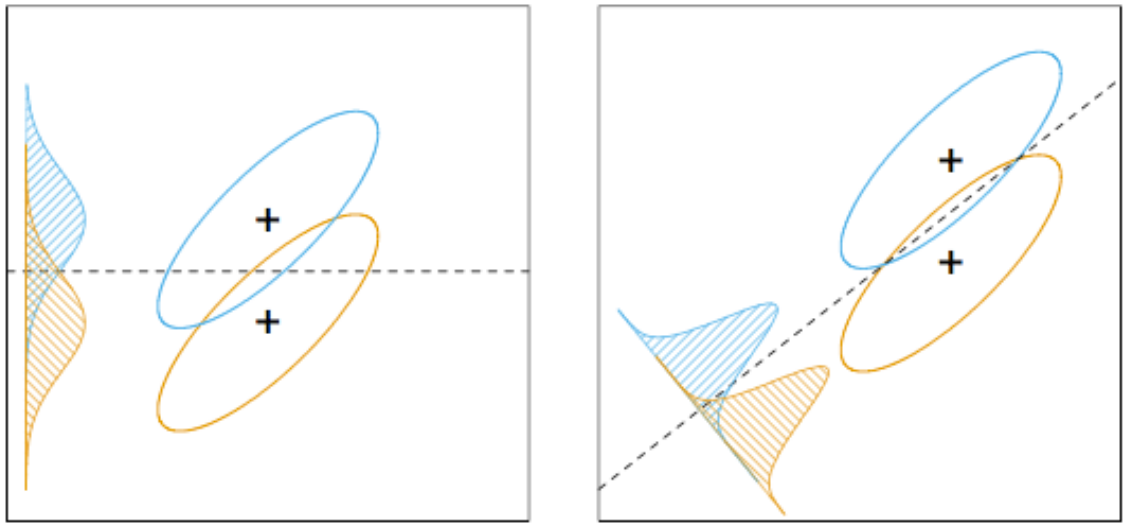
This is a generalized eigenvalue problem, with a given by the largest eigenvalue of $\mathbf{W}^{-1}\mathbf{B}$

$$\hat{\mathbf{B}} = \sum_{k=1}^{K} \frac{N_k}{N} (\hat{\mathbf{u}}_k - \hat{\mathbf{u}})(\hat{\mathbf{u}}_k - \hat{\mathbf{u}})^T$$

$$\hat{\mathbf{W}} = \sum_{k=1}^{K} \frac{N_k}{N} \hat{\mathbf{S}}_k \qquad \hat{\mathbf{S}}_k = \sum_{k=1}^{K} \frac{1}{N_k} (x_i^k - \hat{\mathbf{u}}_k)(x_i^k - \hat{\mathbf{u}}_k)^T$$

Where is **W** within-class covariance and **B** is between-class covariance.

Linear discriminant analysis



Although the line joining the centroids defines the direction of greatest centroid spread, the projected data overlap because of the covariance (left panel). The discriminant direction minimizes this overlap for Gaussian data (right panel).

If the π_k are not equal, moving the cut-point toward the smaller class will improve the error rate.

Logistic regression

The logistic regression model arises from the desire to model the posterior probabilities of the K classes via linear functions in x, while at the same time ensuring that they sum to one.

$$\log\frac{\Pr(G=1\mid X=x)}{\Pr(G=K\mid X=x)}=\beta_{10}+\beta_1^T x$$

$$\vdots$$

$$\log\frac{\Pr(G=K-1\mid X=x)}{\Pr(G=K\mid X=x)}=\beta_{(K-1)0}+\beta_{K-1}^T x$$

$$\log\Pr(G=k\mid X=x)=\frac{\exp(\beta_{k0}+\beta_k^T x)}{1+\sum_{l=1}^{K-1}(\beta_{l0}+\beta_l^T x)},\ k=1,\ldots,K-1$$

$$\log\Pr(G=K\mid X=x)=\frac{1}{1+\sum_{l=1}^{K-1}(\beta_{l0}+\beta_l^T x)}$$

Fitting logistic regression models

For the two-class case, the log-likelihood can be written

$$l(\beta) = \sum_{i=1}^{N} \{ y_i \log p(x_i;\beta) + (1-y_i) \log(1-p(x_i;\beta)) \}$$
$$= \sum_{i=1}^{N} \left\{ y_i \beta^T x_i - \log(1+e^{\beta^T x_i}) \right\}$$

To maximize the log-likelihood, we set its derivative to zero.

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i (y_i - p(x_i;\beta)) = 0$$

The Newton-Raphson algorithm

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial l(\beta)}{\partial \beta}$$

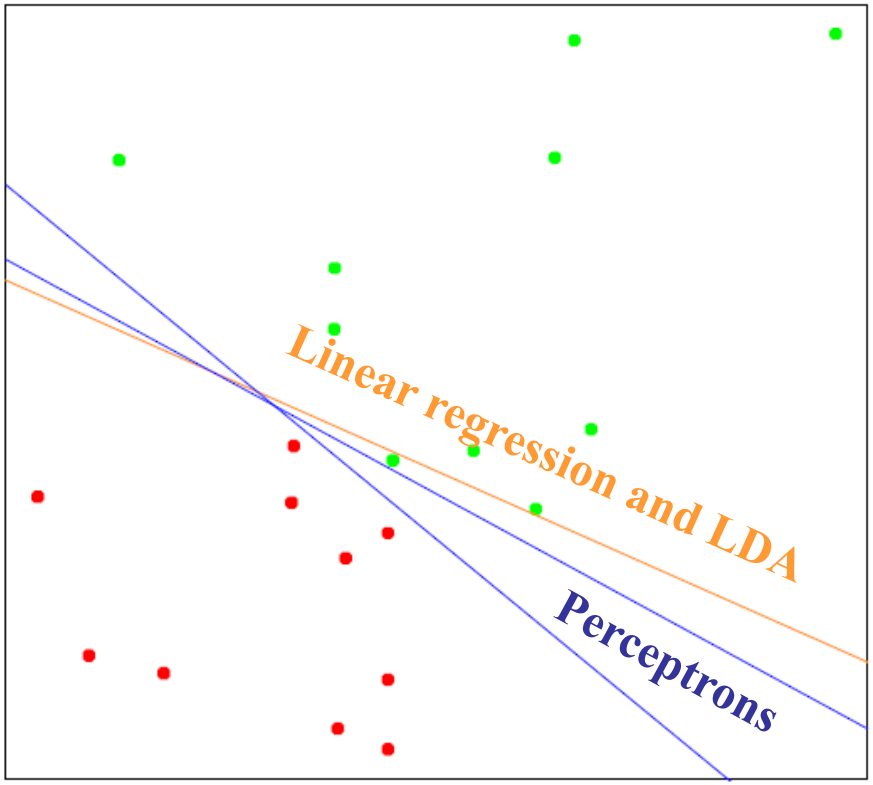
Regularized logistic regression

The L_1 penalty used in the lasso can be used for variable selection and shrinkage with any linear regression model.

$$\max_{\beta_0 \beta} = \left\{ \sum_{i=1}^{N} \left[y_i(\beta_0 + \beta^T x_i) - \log(1 + e^{\beta_0 + \beta^T x_i}) \right] - \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

This criterion is concave, and a solution can be found using nonlinear programming methods.

Separating hyperplanes



The orange line is the least squares solution to the problem, by regressing the $-1/1$ response Y on X; the line is given by

$$x = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 = 0$$

Rosenblatt's perceptron

The perceptron learning algorithm tries to find ***a separating hyperplane*** by minimizing the distance of misclassified points to the ***decision boundary***.

$$D(\beta, \beta_0) = -\sum_{i \in M} y_i (x_i^T \beta + \beta_0)$$

$$\partial \frac{D(\beta, \beta_0)}{\partial \beta} = -\sum_{i \in M} y_i x_i$$

$$\partial \frac{D(\beta, \beta_0)}{\partial \beta_0} = -\sum_{i \in M} y_i$$

The algorithm uses ***stochastic gradient descent*** to minimize this piecewise linear criterion.

$$\begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} \leftarrow \begin{pmatrix} \beta \\ \beta_0 \end{pmatrix} + \rho \begin{pmatrix} y_i x_i \\ y_i \end{pmatrix}$$

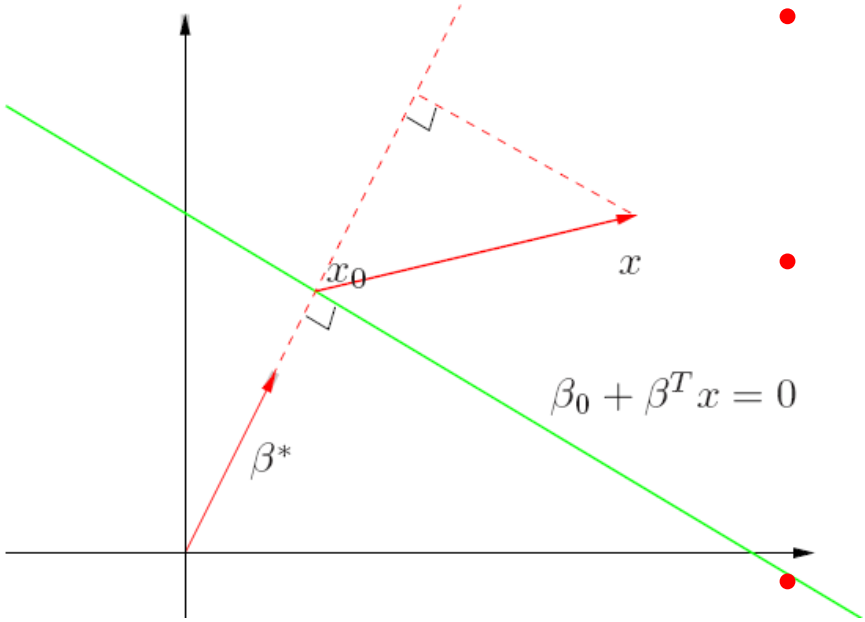
Rosenblatt's perceptron

There are a number of problems with this algorithm:

- When the data are separable, there are ***many solutions***, and which one is found depends on the starting values.
- The "finite" number of steps can be very large. The smaller the gap, the ***longer the time*** to find it.
- When the data are not separable, the algorithm will ***not converge***, and ***cycles*** develop. The cycles can be long and therefore hard to detect.

A rather elegant solution to the first problem is to add ***additional constraints*** to the separating hyperplane (Note that perceptron may lead to many solutions when the data are separable).

The linear algebra of a hyperplane



- The figure depicts a hyperplane or affine set L defined by the equation $f(x) = \beta_0 + \beta^T x = 0$.
- For any two points x_1 and x_2 lying in L, $\beta^T(x_1 - x_2) = 0$, and hence $\beta^* = \beta/\|\beta\|$ is the vector normal to surface of L.
- For any point x in L, $\beta^T x = -\beta_0$.
- The signed distance of any point x to L is given by.

$$\beta^{*T}(x - x_0) = \frac{1}{\|\beta\|}\left(\beta^T x + \beta_0\right)$$

$$= \|f'(x)\|^{-1} f(x)$$

Optimal separating hyperplanes

The optimal separating hyperplane separates the two classes and ***maximizes the distance*** to the closest point from either class.

$$\max_{\beta,\beta_0,\|\beta\|=1} M$$

$$\text{subject to} \quad y_i(x_i^T\beta+\beta_0) \geq M, i=1,\ldots,N$$

The set of conditions ensure that all the points are at least a signed distance M from the decision boundary defined by β and β_0, and we seek the largest such M and associated parameters.

We can get rid of the $\|\beta\| = 1$ constraint by replacing the conditions with

$$\frac{1}{\|\beta\|} y_i(x_i^T\beta+\beta_0) \geq M$$

$$y_i(x_i^T\beta+\beta_0) \geq M\|\beta\|$$

Optimal separating hyperplanes

We can arbitrarily set $||\beta|| = 1/M$.

$$\min_{\beta,\beta_0} \frac{1}{2} || \beta ||^2$$

$$\text{subject to } \; y_i(x_i^T \beta + \beta_0) \geq 1, i = 1, \ldots, N$$

This is a convex optimization problem. The Lagrange function, to be minimized is:

$$L_P = \frac{1}{2} || \beta ||^2 - \sum_{i=1}^{N} \alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - 1 \right]$$

Setting the derivatives to zero, we obtain:

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i, \quad 0 = \sum_{i=1}^{N} \alpha_i y_i,$$

Optimal separating hyperplanes

Substituting these we obtain the so-called Wolfe dual

$$L_P = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$$

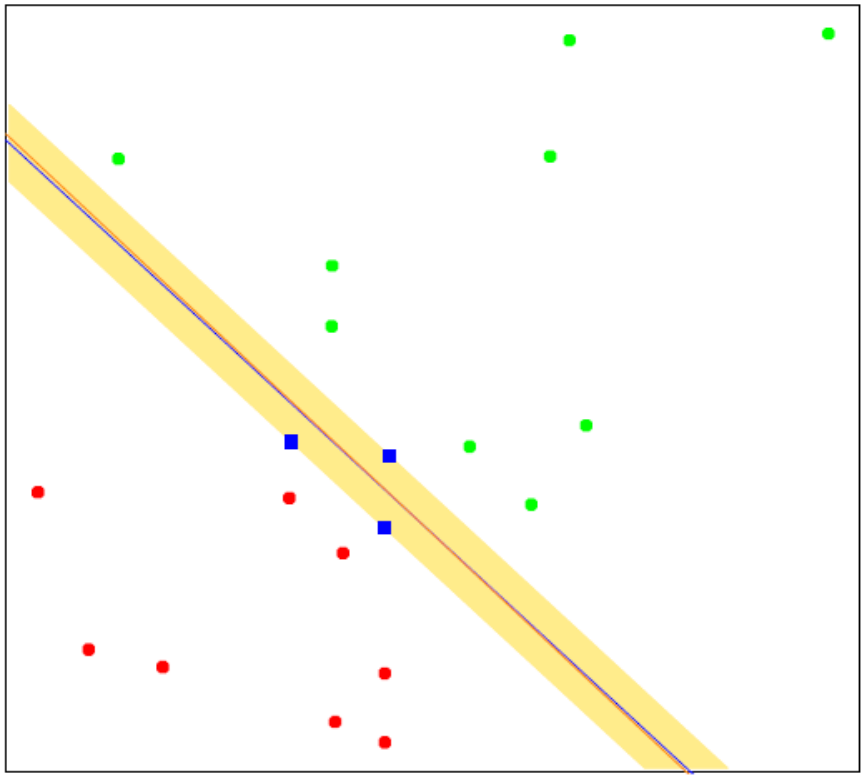
subject to $\alpha_i \geq 0$

In addition the solution must satisfy the Karush-Kuhn-Tucker conditions which include:

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i, \; 0 = \sum_{i=1}^{N} \alpha_i y_i, \text{ and } \alpha_i \left[y_i \left(x_i^T \beta + \beta_0 \right) - 1 \right] = 0 \; \forall i$$

- If $\alpha_i > 0$, then , $y_i \left(x_i^T \beta + \beta_0 \right) = 1$ or in other words, x_i is on the boundary of the slab;
- If $y_i \left(x_i^T \beta + \beta_0 \right) > 1$, x_i is not on the boundary of the slab, and $\alpha_i = 0$.

Optimal separating hyperplanes



The shaded region delineates the ***maximum margin*** separating the two classes. There are ***three support points*** indicated, which lie on the boundary of the margin, and the optimal separating hyperplane (blue line) bisects the slab.

Discussion

- The intuition is that a ***large margin*** on the training data will lead to good separation on the test data.
- The description of the solution in terms of support points seems to suggest that the optimal hyperplane focuses more on the ***points*** than ***count***, and is more ***robust*** to model ***misspecification***.
- The LDA solution, on the other hand, depends on ***all of the data***, even points far away from the decision boundary.
- Note, however, that the identification of these ***support points*** required the use of ***all the data***.
- If the classes are really ***Gaussian***, then LDA is ***optimal***, and separating hyperplanes will pay a price for focusing on the (***noisier***) data at the boundaries of the classes.

Any questions?

AI Research Group
Fudan University