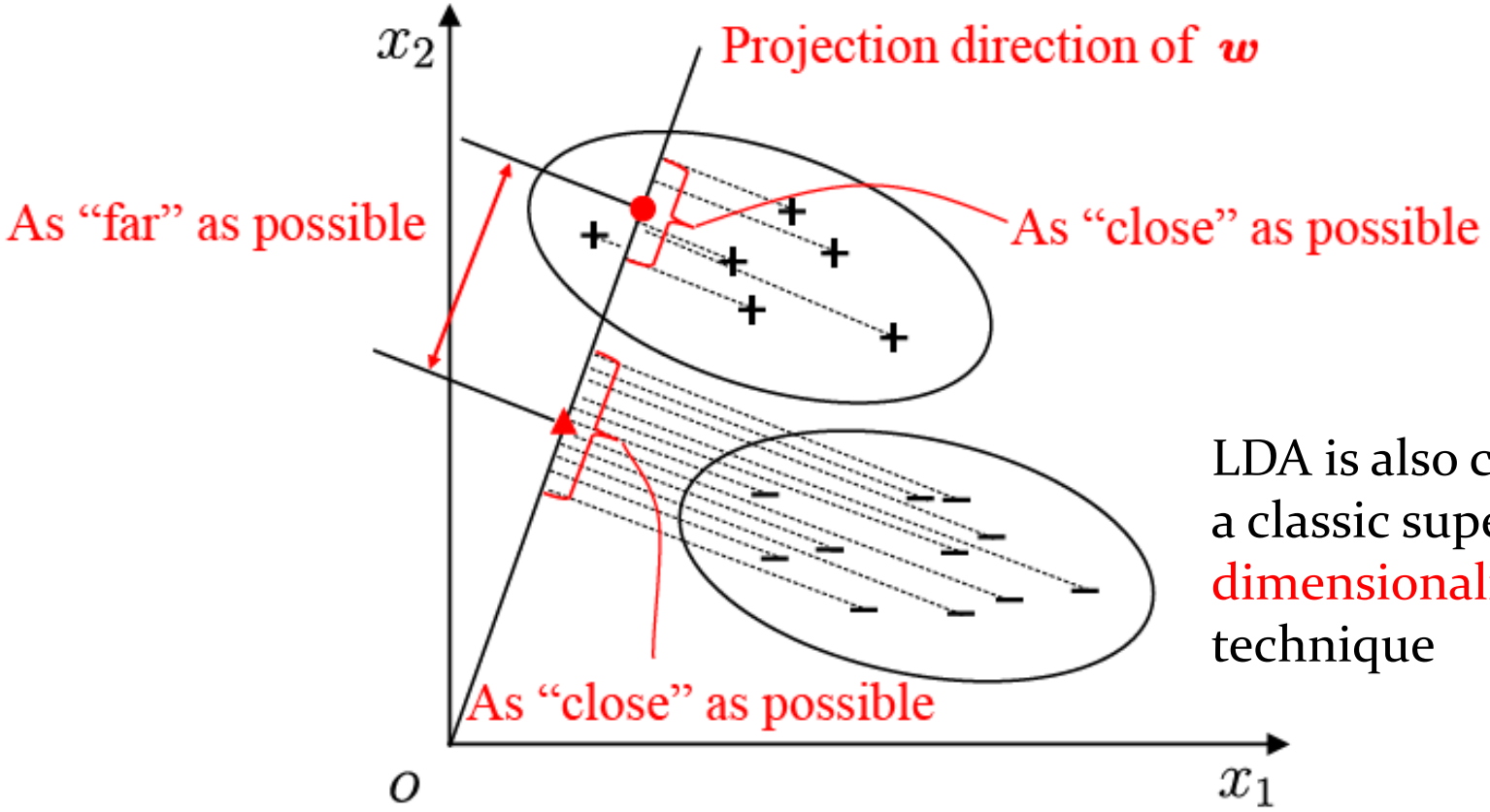
Lecture 5

Linear Discriminant Analysis

Linear Discriminant Analysis

- Linear Discriminant Analysis [Fisher, 1936]



LDA is also considered as a classic supervised dimensionality reduction technique

Linear Discriminant Analysis

- **The idea of LDA:** project the same class samples onto a line, while samples of different classes are far away from each other.
  - To make the projection of similar samples as close as possible, we can make the covariance of the projection points of similar samples as small as possible
  - To make the projection of samples from different classes as far away as possible, we can make the distance between the class centers as large as possible

- Some variables
  - the sample set of the i-th class X_i
  - the mean vector of the i-th class $\boldsymbol{\mu}_i$
  - the covariance matrix of the i-th class $\boldsymbol{\Sigma}_i$
  - the centers of those two classes samples $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$ and $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1$
  - the covariances of the two classes samples $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}$ and $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$

Linear Discriminant Analysis

- We have the objective to be maximized

$$J = \frac{\left\| \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\mu}_1 \right\|_2^2}{\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\Sigma}_0 \boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\Sigma}_1 \boldsymbol{w}}$$
$$= \frac{\boldsymbol{w}^{\mathrm{T}} \left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 \right) \left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 \right)^{\mathrm{T}} \boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}} \left(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1 \right) \boldsymbol{w}}$$

- The within-class scatter matrix

$$\mathbf{S}_w = \boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1$$
$$= \sum_{\boldsymbol{x} \in X_0} \left(\boldsymbol{x} - \boldsymbol{\mu}_0 \right) \left(\boldsymbol{x} - \boldsymbol{\mu}_0 \right)^{\mathrm{T}} + \sum_{\boldsymbol{x} \in X_1} \left(\boldsymbol{x} - \boldsymbol{\mu}_1 \right) \left(\boldsymbol{x} - \boldsymbol{\mu}_1 \right)^{\mathrm{T}}$$

- The between-class scatter matrix $\mathbf{S}_b = \left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 \right) \left(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1 \right)^{\mathrm{T}}$

Linear Discriminant Analysis

- Generalized Rayleigh quotient (广义瑞利商)

$$J = \frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}}$$

- Let $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1$, maximizing generalized Rayleigh quotient is equivalent to

$$\min_{\boldsymbol{w}} \; -\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}$$

$$\text{s.t. } \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1$$

- Using the method of Lagrangian multipliers

$$\mathbf{S}_b\boldsymbol{w} = \lambda\mathbf{S}_w\boldsymbol{w}$$

Linear Discriminant Analysis

- Since the direction of $\mathbf{S}_b w$ is always $\mu_0 - \mu_1$

$$\mathbf{S}_b \boldsymbol{w} = \lambda (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$$

Direction consistency

gives

$$\boldsymbol{w} = \mathbf{S}_w^{-1} (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$$

- Solve
  - Singular value decomposition $\mathbf{S}_w = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$

- Explain from the aspect of Bayesian decision theory
  - It can be proved that we have the optimal solution of LDA when both classes follow Gaussian distribution with the same prior and covariance

Extend LDA to multiclass classification

- The global scatter matrix

$$\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$$

$$= \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$

- The within-class scatter matrix $\mathbf{S}_w = \sum_{i=1}^{N} \mathbf{S}_{w_i}$

where $\mathbf{S}_{w_i} = \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$

- We have

$$\mathbf{S}_b = \mathbf{S}_t - \mathbf{S}_w$$

$$= \sum_{i=1}^{N} m_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

Extend LDA to multiclass classification

- Objective

$$\max_{\mathbf{W}} \frac{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}_b\mathbf{W}\right)}{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}_w\mathbf{W}\right)}$$

$$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{n} A_{ii}$$

where $\mathbf{W} \in \mathbb{R}^{d\times(N-1)}$

$$\mathbf{S}_b\mathbf{W} = \lambda\mathbf{S}_w\mathbf{W}$$

Concatenating the eigenvectors corresponding to the d' largest non-zero eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$ leads to the closed-form solution of W, where $d' \leq N-1$

- Since the projection reduces the data while considering the class information, LDA is also considered as a classic supervised dimensionality reduction technique.

Singular Value Decomposition

Singular Value Decomposition (奇异值分解，SVD)

Any real matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ can be decomposed as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$$

$\mathbf{U} \in \mathbb{R}^{m \times m}$ is a unitary matrix of order m satisfying $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$,

$\mathbf{V} \in \mathbb{R}^{n \times n}$ is a unitary matrix of order n satisfying $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$,

$\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is a $m \times n$ matrix with all of its elements take the value 0 except $(\mathbf{\Sigma})_{ii} = \sigma_i$, where σ_i are non-negative real numbers and $\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant 0$.

Singular Value Decomposition

- Applications of SVD, e.g., low-rank matrix approximation

to approximate a r-rank matrix $\mathbf{A}$ with a k-rank matrix $\widetilde{\mathbf{A}}$, where $k \leqslant r$,

$$\min_{\widetilde{\mathbf{A}} \in \mathbb{R}^{m \times n}} \|\mathbf{A} - \widetilde{\mathbf{A}}\|_F$$

$$\text{s.t. } \operatorname{rank}(\widetilde{\mathbf{A}}) = k.$$

SVD can provide an analytical solution:

- Performing SVD on A;
- obtain a matrix $\mathbf{\Sigma}_k$ by setting the $r - k$ smallest singular values in $\mathbf{\Sigma}$ to zero, (i.e., keep only the k largest singular values)
- the optimal solution: $\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{T}}$,

  $\mathbf{U}_k$ and $\mathbf{V}_k$ are, respectively, the first k columns of $\mathbf{U}$ and $\mathbf{V}$

LDA for classification

- *When the classes are well-separated*, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- *If n is small and the distribution of the predictors X is approximately normal in each of the classes*, the linear discriminant model is again more stable than the logistic regression.
- Linear discriminant analysis is popular when we have more than two classes, because it also provides low-dimensional views of the data.

Bayes Theorem for Classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probabilistic modeling. Here we focus on a simple result, known as Bayes theorem:

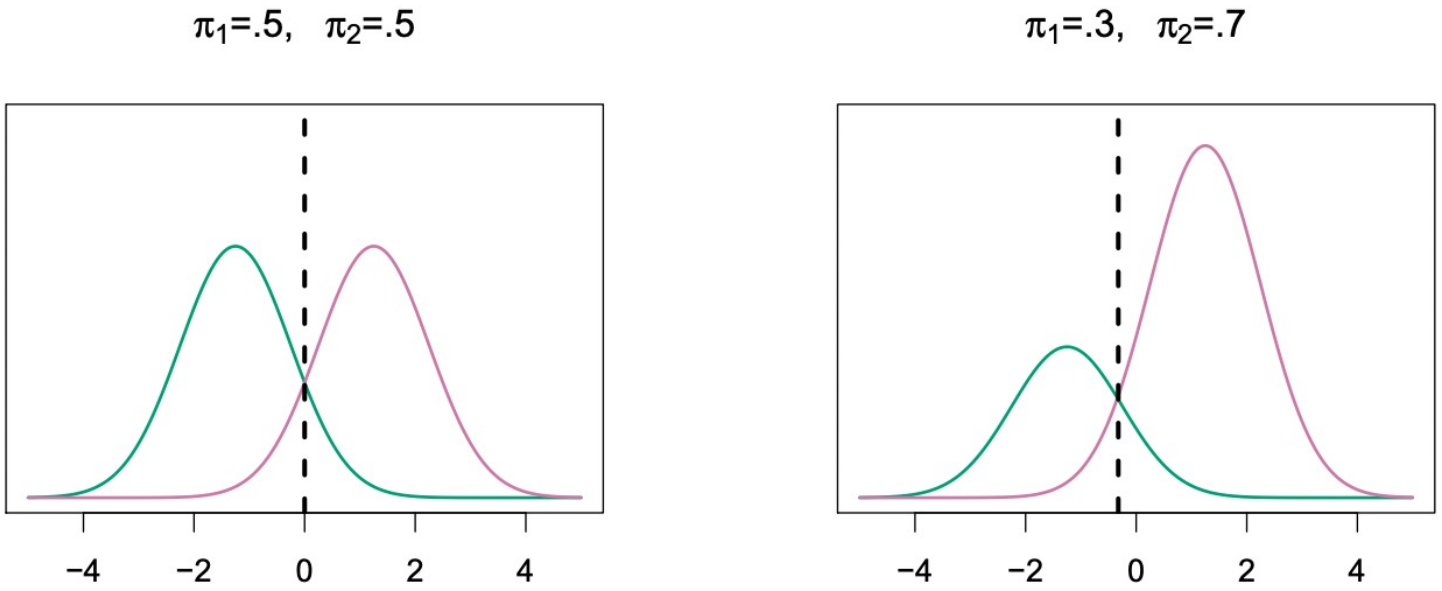
$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

One writes this slightly differently for discriminant analysis:

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}, \quad \text{where}$$

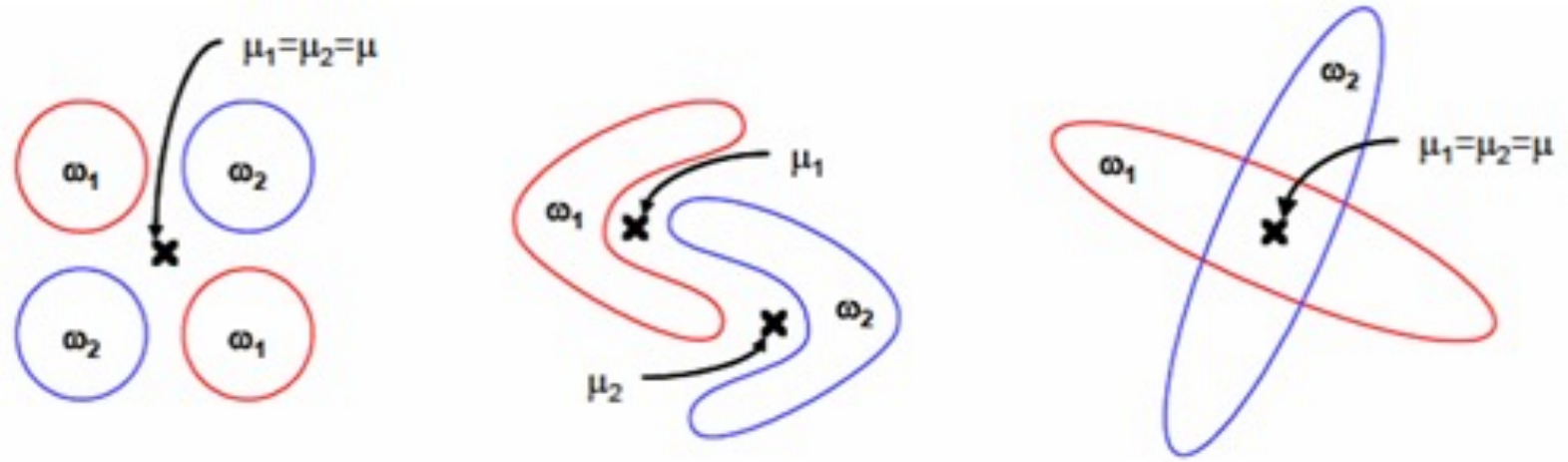
- $f_k(x) = \Pr(X = x \mid Y = k)$ is the density for X in class k. Here we will use normal densities for these, separately in each class.
- $\pi_k = \Pr(Y = k)$ is the marginal or prior probability for class k.

Classify to the highest density



- We classify a new point according to which density is highest
- When the priors are different, we take them into account as well, and compare $\pi_k \, f_k(x)$. On the right, we favor the pink class — the decision boundary has shifted to the left.

LDA cannot handle non-Gaussian data



Linear Discriminant Analysis (p = 1)

The Gaussian density has the form

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

Here μ_k is the mean, and σ_k^2 the variance (in class k). We will assume that all the $\sigma_k = \sigma$ are the same.

Plugging this into Bayes formula, we get a rather complex expression for $p_k(x) = \Pr(Y = k \mid X = x)$:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2}}$$

Happily, there are simplifications and cancellations.

Discriminant functions

To classify at the value $X = x$, we need to see which of the $p_k(x)$ is largest. Taking logs, and discarding terms that do not depend on k, we see that this is equivalent to assigning x to the class with the largest *discriminant score*:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

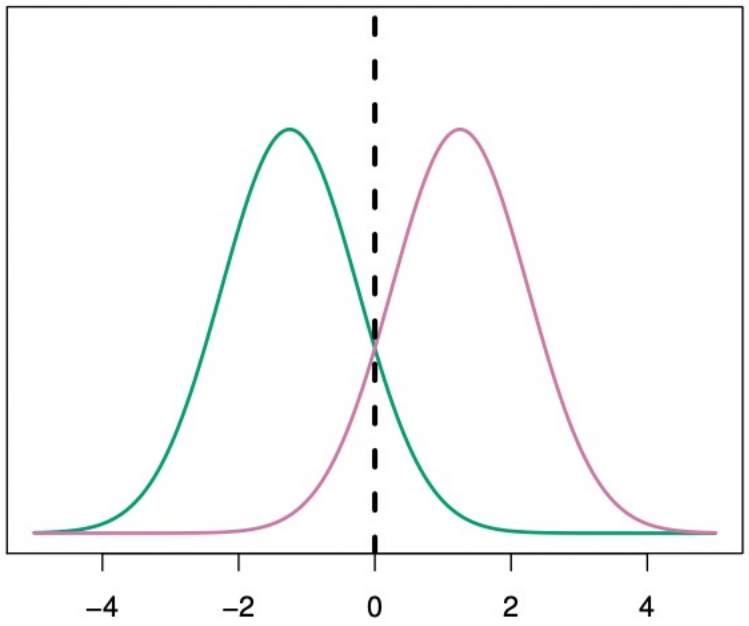
Note that $\delta_k(x)$ is a *linear* function of x.

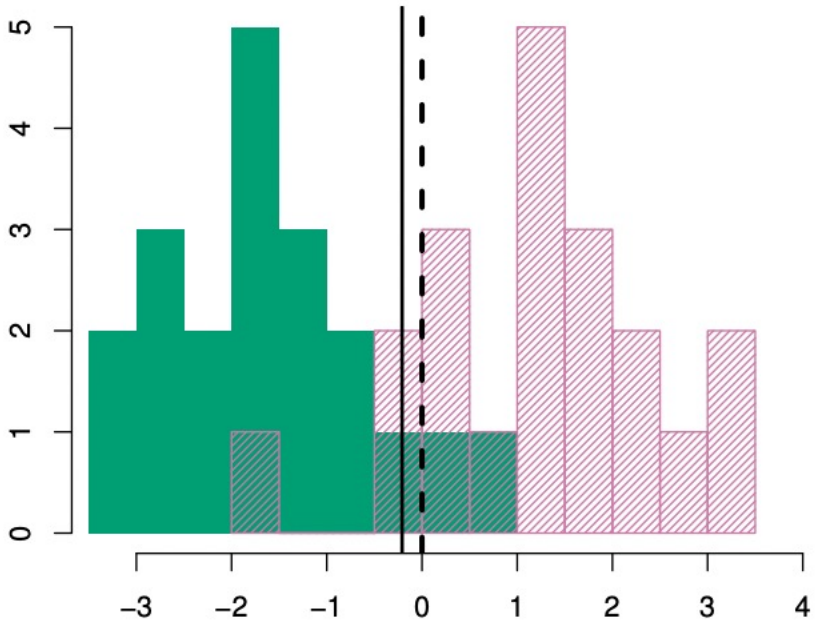
If there are $K = 2$ classes and $\pi_1 = \pi_2 = 0.5$, then one can see that the decision boundary is at

$$x = \frac{\mu_1 + \mu_2}{2}.$$

(See if you can show this)

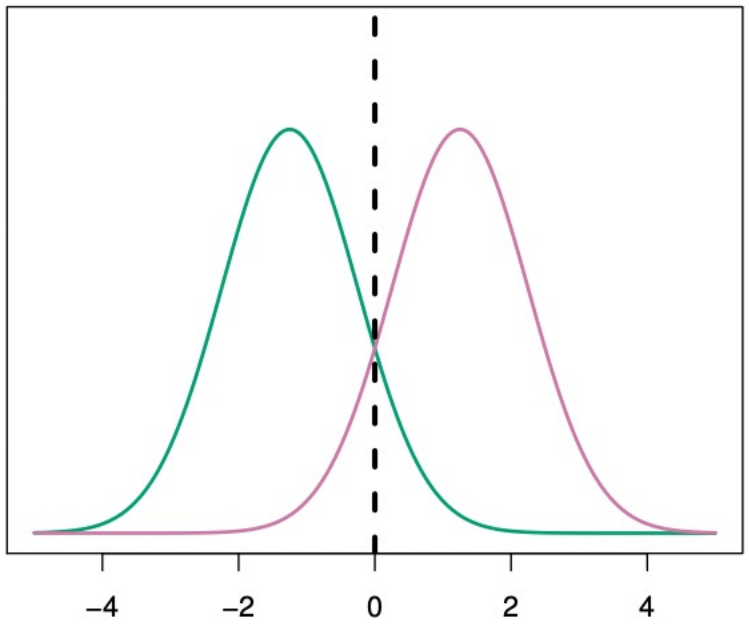
Discriminant functions

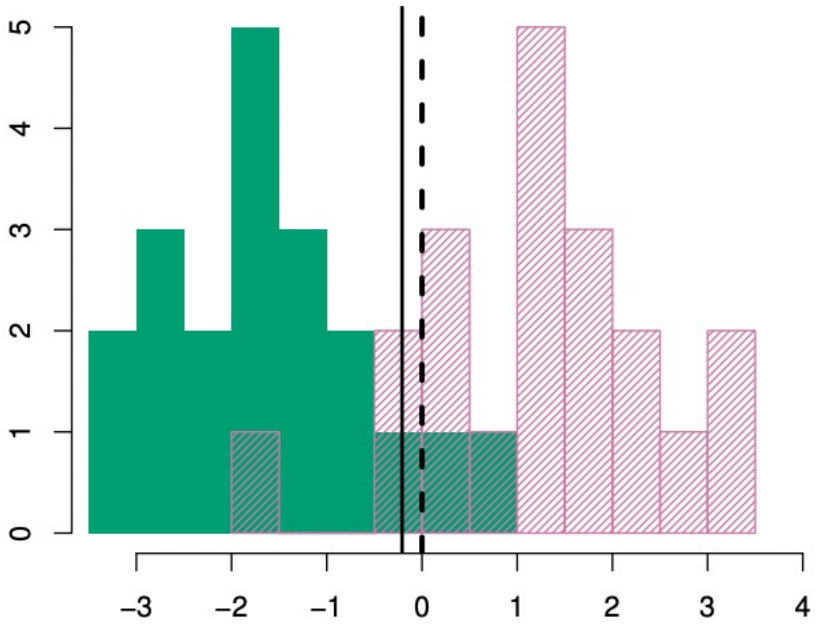




Example with $\mu_1 = -1.5, \mu_2 = 1.5, \pi_1 = \pi_2 = 0.5$, and $\sigma^2 = 1$

Discriminant functions





- Example with $\mu_1 = -1.5, \mu_2 = 1.5, \pi_1 = \pi_2 = 0.5$, and $\sigma^2 = 1$
- Typically we don't know these parameters; we just have the training data. In that case we simply estimate the parameters and plug them into the rule.

Estimating the parameters

$$\hat{\pi}_k = \frac{n_k}{n}$$

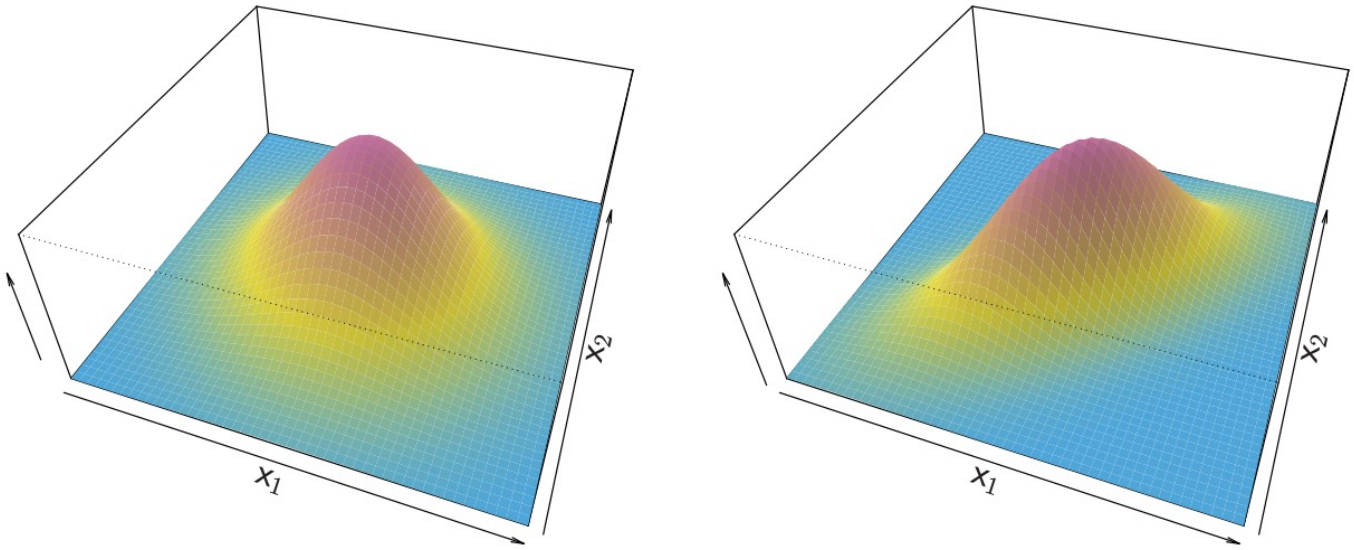
$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:\, y_i = k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:\, y_i = k} (x_i - \hat{\mu}_k)^2$$

$$= \sum_{k=1}^{K} \frac{n_k - 1}{n-K} \cdot \hat{\sigma}_k^2$$

where $\hat{\sigma}_k^2 = \frac{1}{n_k - 1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$ is the usual formula for the estimated variance in the k-th class.

Linear Discriminant Analysis (p > 1)



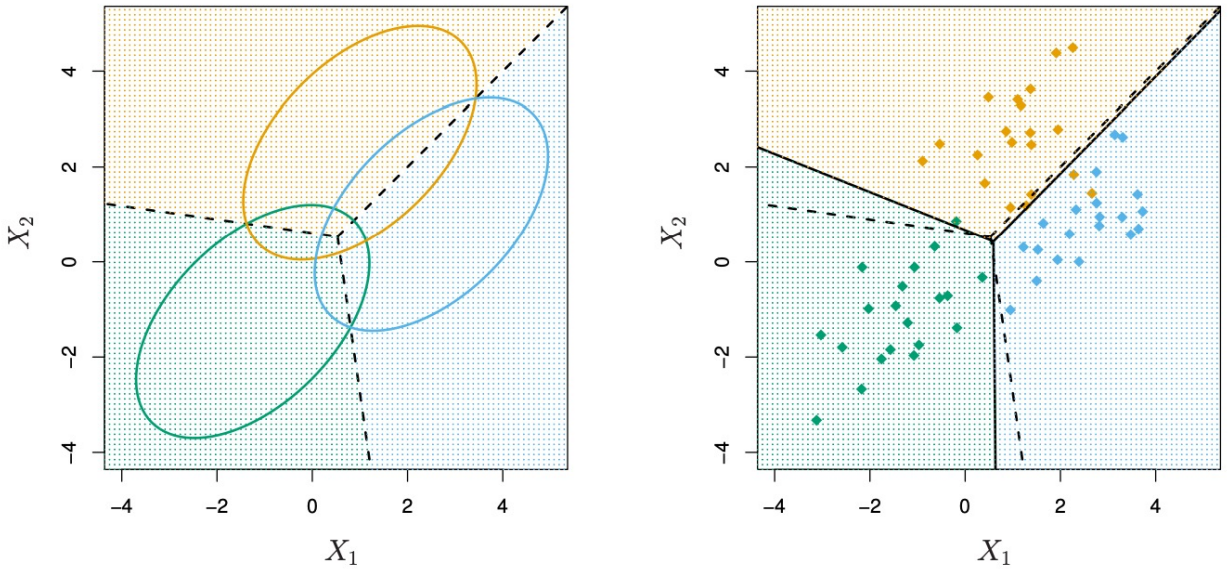
$$\text{Density: } f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)}$$

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

Despite its complex form,

$\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \ldots + c_{kp}x_p$ — a linear function.

Illustration: p = 2 and K = 3 classes



Here $\pi_1 = \pi_2 = \pi_3 = 1/3$.

The dashed lines are known as the *Bayes decision boundaries*. Were they known, they would yield the fewest misclassification errors, among all possible classifiers.

Logistic Regression versus LDA

For a two-class problem, one can show that for LDA

$$\log\left(\frac{p_1(x)}{1-p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1 x_1 + \ldots + c_p x_p$$

So it has the same form as logistic regression.

The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on Pr*(Y |X)* (known as *discriminative learning*).
- LDA uses the full likelihood based on Pr*(X, Y)* (known as *generative learning*).
- Despite these differences, in practice the results are often very similar.

Multi-class Classification

Multiclass Classification

- Multiclass Classification learning methods
  - Some binary classification methods can be directly extended to accommodate multiclass cases
  - Apply some strategies to solve multiclass classification problems with any existing binary classification methods （more general）
    - Decompose the problem and then train a binary classifier for each divided binary classification problem
    - Ensemble the outputs collected from all binary classifiers into the final multiclass predictions
- Dividing strategies
  - One vs. One (OvO)
  - One vs. Rest (OvR)
  - Many vs. Many (MvM)

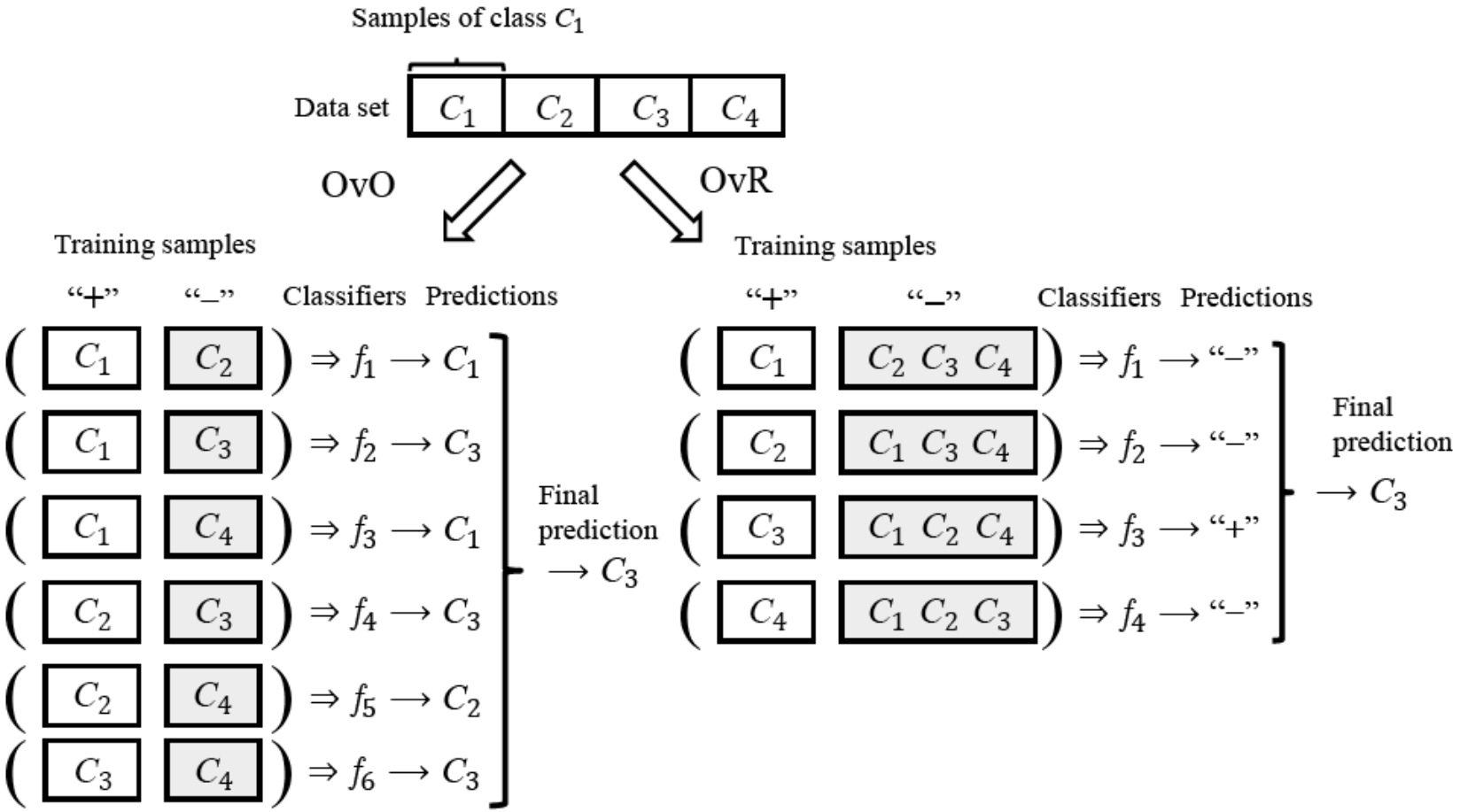
Multiclass Classification - OvO

- In the decomposing phase
  - puts the N classes into pairs
    - $N(N-1)/2$ binary classification tasks
  - trains a classifier for each task
    - $N(N-1)/2$ classifiers
- In the testing phase
  - a new sample is classified by all classifiers
    - $N(N-1)/2$ classification outputs
  - the final prediction can be made via voting
    - the predicted class is the one received the most votes

Multiclass Classification - OvR

- In the decomposing phase
  - consider each class as positive in turn, and the rest classes are considered as negative
    - *N* binary classification tasks
  - trains a classifier for each task
    - *N* classifiers
- In the testing phase
  - a new sample is classified by all classifiers
    - *N* classification outputs
  - the prediction confidences are usually assessed
    - the class with the highest confidence is used as the classification result

Multiclass Classification – A comparison between OvO and OvR



Multiclass Classification – A comparison between OvO and OvR

OvO

- Train $N(N-1)/2$ classifiers，the memory and testing time costs are often higher
- Each classifier uses only samples of two classes. Hence, the computational cost of training OvO is lower

OvR

- Train N classifiers，the memory and testing time costs are often lower
- Each classifier uses all training samples. Hence, the computational cost of training OvO is higher

As for the prediction performance, it depends on the specific data distribution, and in most cases, the two methods have similar performance.

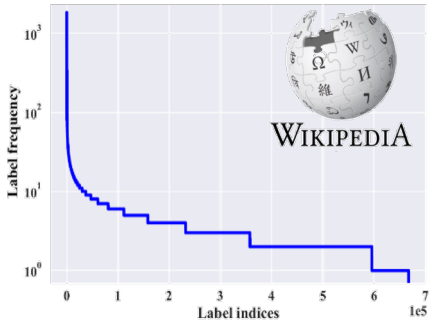
Recent Progress

Class Imbalance Problem

- **Deep Learning**

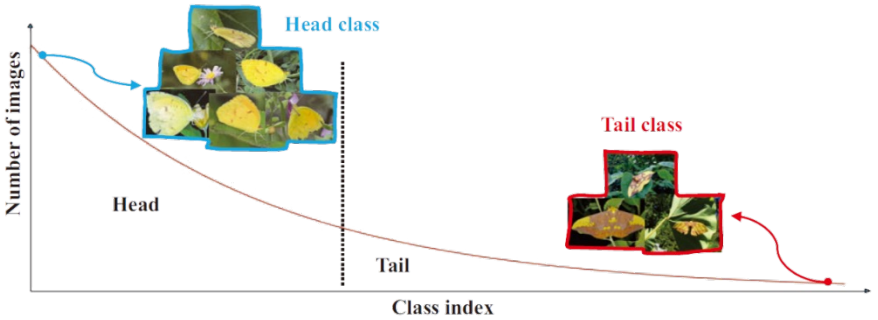
enablers →
- Faster computers
- Algorithmic improvements
- Access to large amounts of data

- **Real-world class-imbalanced (long-tailed) class distribution**



Zipf's law

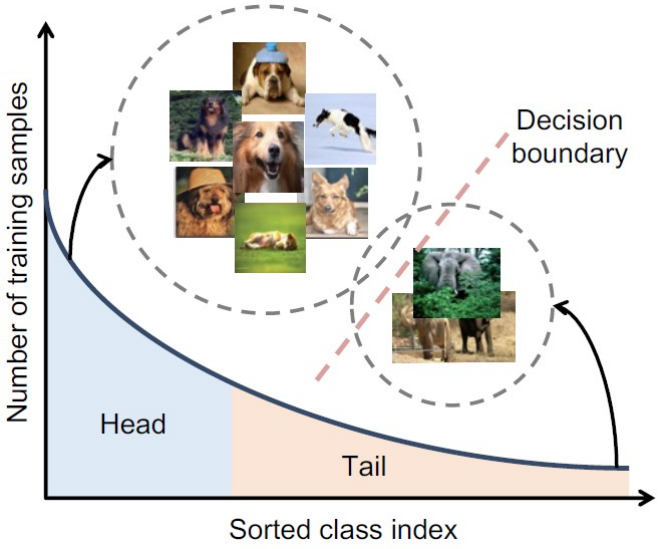
(Power law)



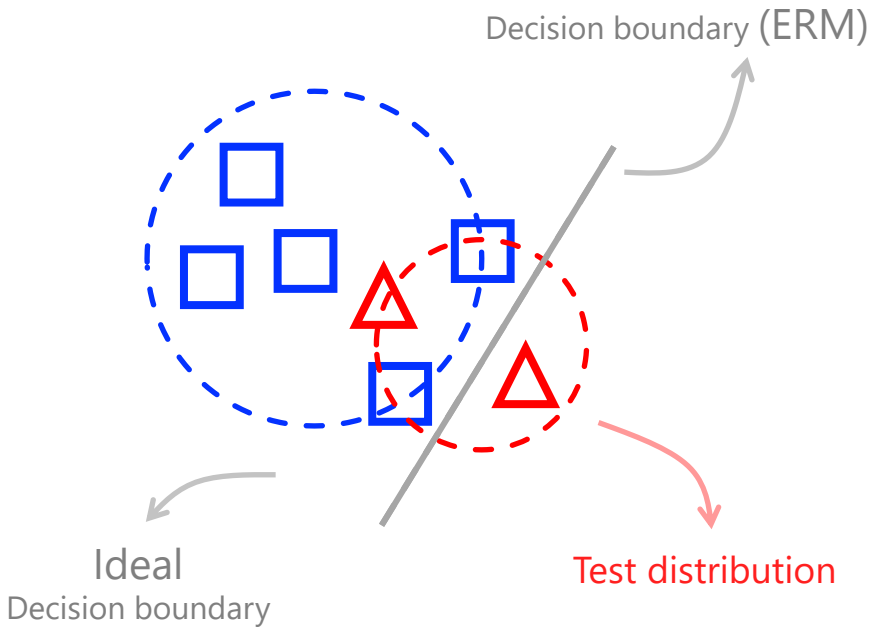
Class Imbalance Problem

Challenge

—— model biased towards head classes



Take an example

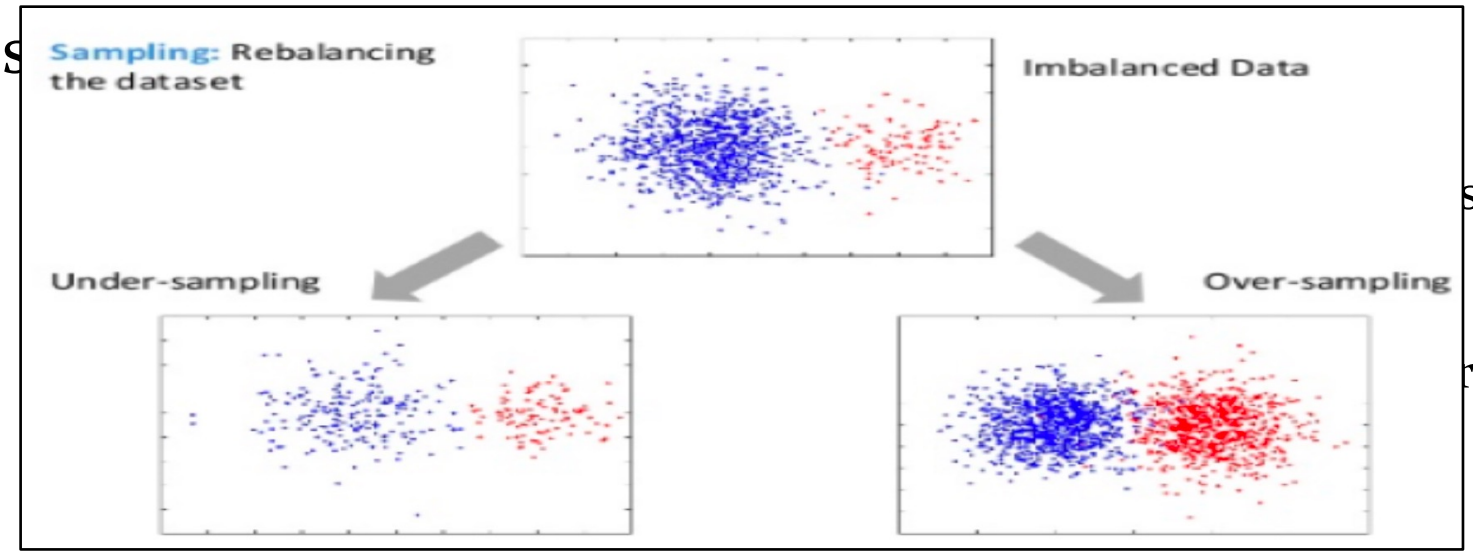


Class Imbalance Problem

- Class imbalance
  - a significantly different number of samples for each class.（the positive class is the minority）

the classes are balanced $\frac{y}{1-y} > 1$ ⇒ $\frac{y}{1-y} > \frac{m^+}{m^-}$ the observed class ratio

- Res
  -
  -
  -

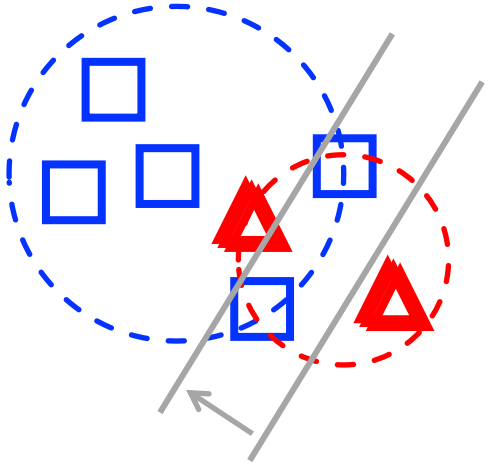


sses

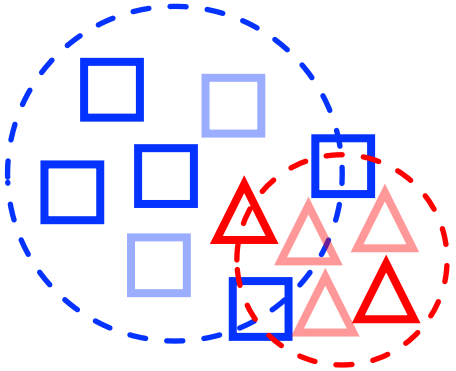
re

Class Imbalance Problem

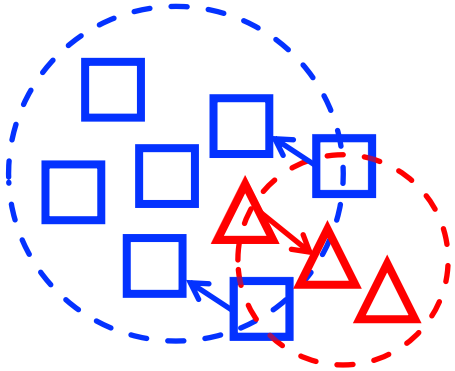
Class Re-balancing



Data Augmentation

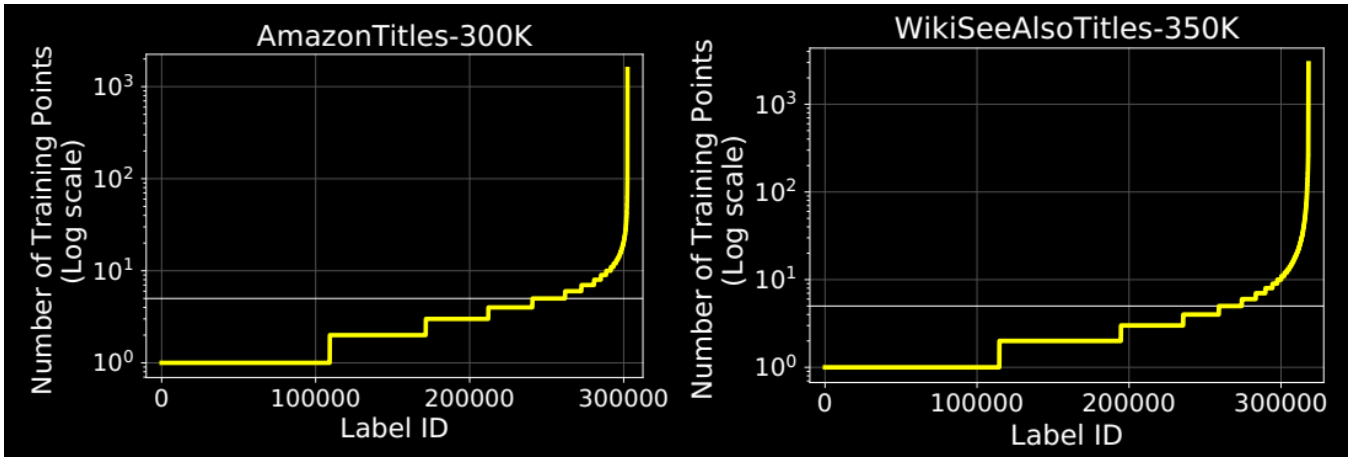


Class-balanced Loss Function

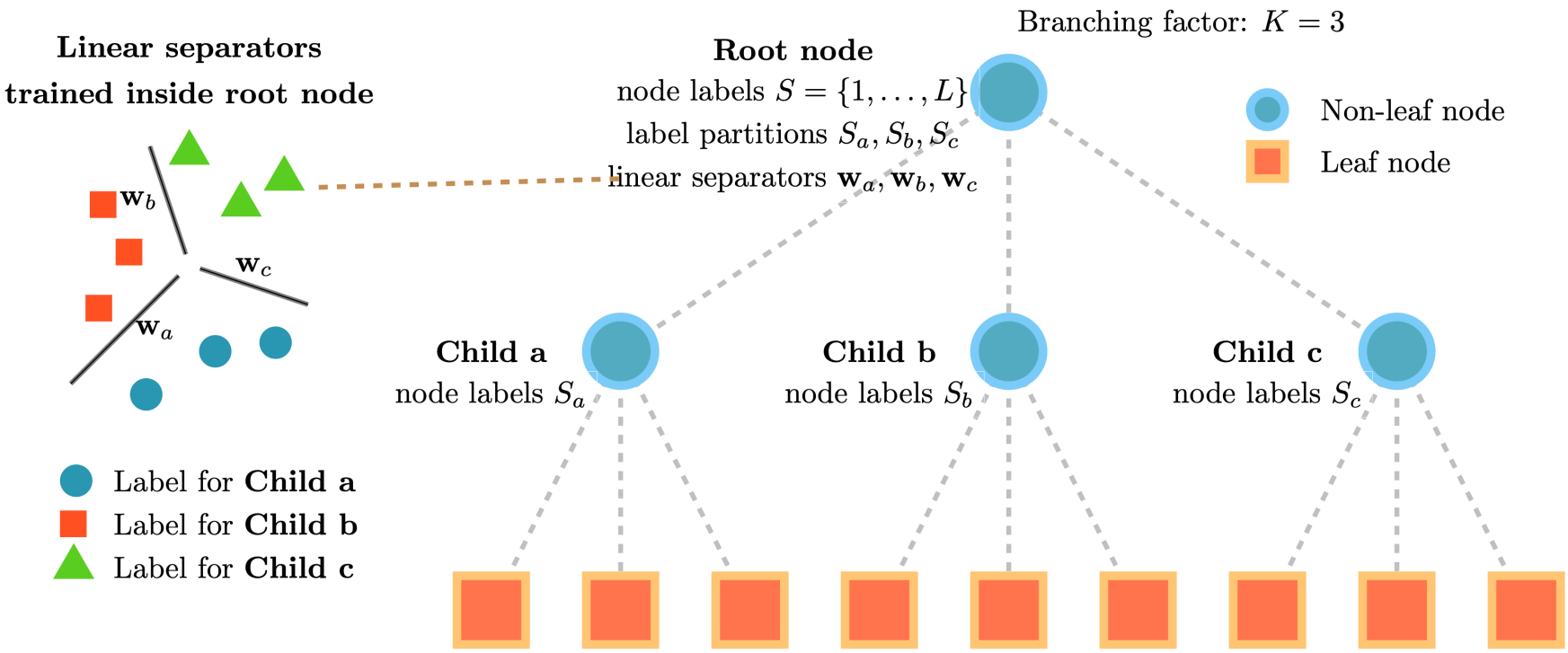


Extreme Classification

	Dataset	# of Train Points	# of Labels	# of Test Points
Benchmark*	LF-AmazonTitles-131K	294,805	131,073	134,835
	LF-WikiSeeAlsoTitles-320K	693,082	312,330	177,515
	LF-AmazonTitles-1.3M	2,248,619	1,305,265	970,237
Bing	LF-P2PTitles-300K	1,366,429	300,000	585,602
	LF-P2PTitles-2M	2,539,009	1,640,898	1,088,146



Label Tree



- recursively partition the set of classes into subsets
- train a multi-class (linear) classifier for inference