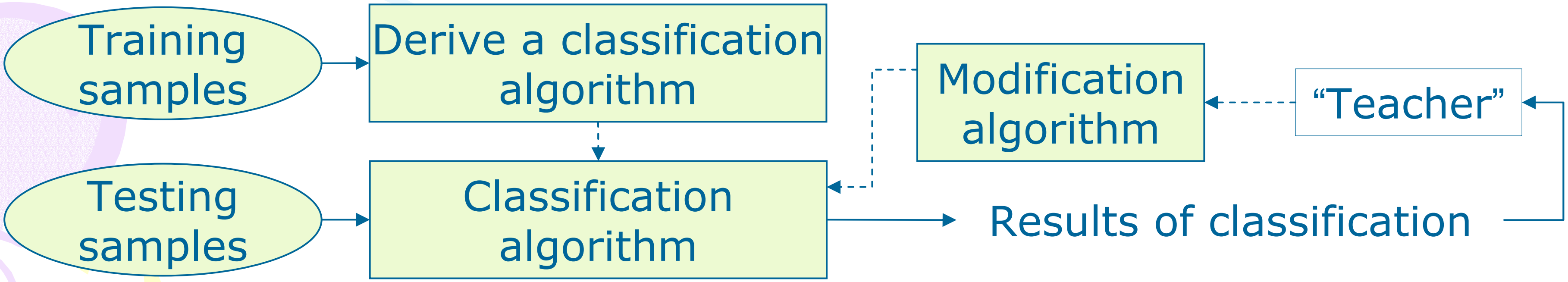
Decision Estimation

- Decision estimation and classification are ones of active research areas.
  - Classic measurements of the environment, ⋯ , AI (vision, speech recognition.)
    - ➢ Systems perform "pattern recognition" or "decision making".
  - Often the information is less than precise, and frequently the decision procedures are statistical in nature.
- Objects of interests are classified into one of classes.
  - These objects are "patterns": printed letters, biological cells,⋯
  - Systems learn the training data to classify the testing data.
    - Supervised vs. un-supervised.



Training samples

Derive a classification algorithm

Modification algorithm

“Teacher”

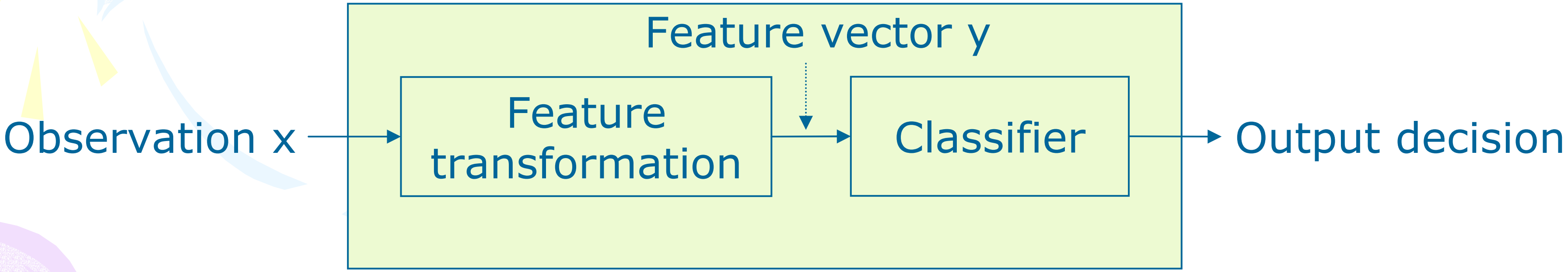
Testing samples

Classification algorithm

Results of classification

Classification Approach

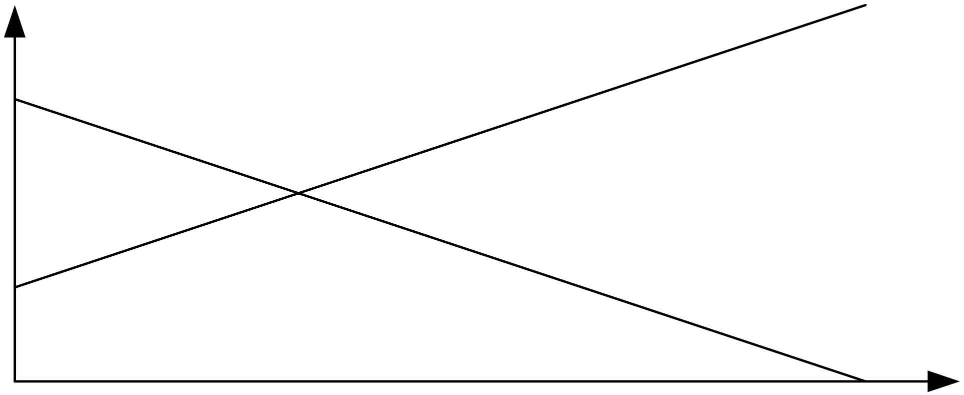
- The observation vector **x** is first transformed into another vector **y** whose components called **features**.
  - **Feature extraction**: the features are intended to be fewer in numbers than the observations.
  - However, they should collectively contain most discernible information for pattern classification.
    - Reduction of the observations to a smaller number of features is anticipated to help design a reliable decision rule.



  - Extraction procedures or transformations attempt to compute the components based on intuition or physical considerations of the problem. ⇒ dimensionality reduction.

Three-class Recognition Example

$$\begin{array}{lll} \text{if} & y_1 + 3y_2 < 9 & w_1 : \text{class of 1's,} \\ \text{else if} & -y_1 + 3y_2 > 3 & w_2 : \text{class of x's,} \\ \text{else} & & w_3 : \text{class of 0's.} \end{array}$$



Discriminant functions:

$$\begin{cases} g_1(\mathbf{y}) = -y_1 - 3y_2 + 9, \\ g_2(\mathbf{y}) = -y_1 + 3y_2 - 3, \\ g_3(\mathbf{y}) = g_1(\mathbf{y}) \cdot g_2(\mathbf{y}). \end{cases}$$

⇒

Decision Rule:

Choose w_i where $g_i(\mathbf{y}) = \max_j [g_j(\mathbf{y})].$

- Decision region R_i is the set $R_i = \{\mathbf{y} : g_i(\mathbf{y}) = \max_j [g_j(\mathbf{y})].$
  - Discriminant functions can be evaluated computationally.

- Decision boundaries are defined by $g_i(\mathbf{y}) = g_j(\mathbf{y}), \quad i \neq j.$

Probability Theory for Random Vectors

- Event A has the associated probability P(A).
  - P(not A) = 1 – P(A).
  - The joint probability of two events A & B, denoted as P(AB) or P(A and B) is the probability that A and B both occur simultaneously.
    - P(A or B) = P(A) + P(B) - P(A and B).

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

- Suppose **x** is a random vector.
  - Its distribution function F(**x**) is defined as

$$F_{\mathbf{x}}(\tilde{\mathbf{x}}) = F_{x_1,x_2,\cdots,x_n}(\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n) = P(\mathbf{x} \le \tilde{\mathbf{x}}) = P(x_1 \le \tilde{x}_1, x_2 \le \tilde{x}_2, \cdots, x_n \le \tilde{x}_n).$$

    - F(-∞)=0, F(+∞)=1.

  - Its density function f(**x**) is defined as

$$f_{\mathbf{x}}(\tilde{\mathbf{x}}) = \frac{dF_{\mathbf{x}}(\tilde{\mathbf{x}})}{d\mathbf{x}} = \left[\frac{\partial^n F_{\mathbf{x}}(\tilde{\mathbf{x}})}{\partial x_1 \partial x_2 \cdots \partial x_n}\right]_{\mathbf{x}=\tilde{\mathbf{x}}} \Leftrightarrow F_{\mathbf{x}}(\tilde{\mathbf{x}}) = \int_{-\infty}^{\tilde{\mathbf{x}}} f_{\mathbf{x}}(\mathbf{x})d\mathbf{x} = \int_{-\infty}^{\tilde{x}_1}\int_{-\infty}^{\tilde{x}_2}\cdots\int_{-\infty}^{\tilde{x}_n} f_{\mathbf{x}}(\mathbf{x})dx_1 dx_2 \cdots dx_n.$$

Joint Distribution and Density Functions

- Suppose **y** is another random vector.

$$\mathbf{y}=\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

- The joint distribution of **x** and **y** is defined by

$$F_{\mathbf{xy}}(\tilde{\mathbf{x}},\tilde{\mathbf{y}})=P(\mathbf{x}\le\tilde{\mathbf{x}},\mathbf{y}\le\tilde{\mathbf{y}}).$$

- The joint density is $f_{\mathbf{xy}}(\tilde{\mathbf{x}},\tilde{\mathbf{y}})=\dfrac{d^2F_{\mathbf{x}}(\tilde{\mathbf{x}})}{d\mathbf{x}d\mathbf{y}}=\left[\dfrac{\partial^m\partial^n F_{\mathbf{x}}(\tilde{\mathbf{x}})}{\partial x_1\partial x_2\cdots\partial x_n\partial y_1\partial y_2\cdots\partial y_m}\right]_{\mathbf{x}=\tilde{\mathbf{x}},\mathbf{y}=\tilde{\mathbf{y}}}$.

$$\Rightarrow\quad F_{\mathbf{xy}}(-\infty,-\infty)=0,\quad F_{\mathbf{xy}}(\infty,\infty)=1,\quad F_{\mathbf{xy}}(\tilde{\mathbf{x}},\infty)=F_{\mathbf{x}}(\tilde{\mathbf{x}}),\quad F_{\mathbf{xy}}(\infty,\tilde{\mathbf{y}})=F_{\mathbf{y}}(\tilde{\mathbf{y}}).$$

Marginal p.d.f.

- Example: the joint p.d.f. is $f_{\mathbf{xy}}(\tilde{\mathbf{x}},\tilde{\mathbf{y}})=\begin{cases}(\tilde{x}_1+3\tilde{x}_2)\tilde{y}_1 & 0\le\tilde{x}_1,\tilde{x}_2,\tilde{y}_1\le 1,\\ 0 & \text{otherwise.}\end{cases}$

$$\mathbf{x}=\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\qquad \mathbf{y}=\begin{pmatrix} y_1 \end{pmatrix}$$

$$f_{\mathbf{x}}(\tilde{\mathbf{x}})=\begin{cases}\int_0^1(\tilde{x}_1+3\tilde{x}_2)y_1dy_1=\tfrac{1}{2}(\tilde{x}_1+3\tilde{x}_2) & 0\le\tilde{x}_1,\tilde{x}_2\le 1,\\ 0 & \text{otherwise.}\end{cases}$$

Probability Functions jointly with Events

- The joint distribution of a random vector **x** and an event A is defined by

$$F_{\mathbf{x}A}(\tilde{\mathbf{x}}, A) = P(\mathbf{x} \le \tilde{\mathbf{x}}, A) = \sum\nolimits_{i=1}^{m} P(\mathbf{x} \le \tilde{\mathbf{x}}, A_i), \quad A = \bigcup\nolimits_{i=1}^{m} A_i.$$

- The conditional probability: $P(A \mid B) = \dfrac{P(A \cap B)}{P(B)} = \dfrac{P(B \mid A)P(A)}{P(B)}.$

Bayes's Rule

$$\Rightarrow \quad F_{\mathbf{x}|A}(\tilde{\mathbf{x}} \mid A) = P(\mathbf{x} \le \tilde{\mathbf{x}} \mid A) = \frac{P(\mathbf{x} \le \tilde{\mathbf{x}}, A)}{P(A)} = \frac{F_{\mathbf{x}A}(\tilde{\mathbf{x}}, A)}{P(A)}.$$

$$\Rightarrow \quad f_{\mathbf{x}|\mathbf{y}}(\tilde{\mathbf{x}} \mid \tilde{\mathbf{y}}) = \frac{f_{\mathbf{xy}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})}{f_{\mathbf{y}}(\tilde{\mathbf{y}})}. \qquad \Rightarrow \quad f_{\mathbf{xy}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = f_{\mathbf{x}}(\tilde{\mathbf{x}}) \cdot f_{\mathbf{y}}(\tilde{\mathbf{y}}), \text{if } \mathbf{x}, \mathbf{y} \text{ are independent.}$$

$$P(A_i \mid B) = \frac{P(A_i \cap B)}{P(B)} = \frac{P(B \mid A_i)P(A_i)}{\sum\nolimits_{i=1}^{m} P(B \mid A_i)}$$

Prior density

$$\Rightarrow \quad f_{\mathbf{x}|\mathbf{y}}(\tilde{\mathbf{x}} \mid \tilde{\mathbf{y}}) = \frac{f_{\mathbf{xy}}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})}{f_{\mathbf{y}}(\tilde{\mathbf{y}})} = \frac{f_{\mathbf{y}|\mathbf{x}}(\tilde{\mathbf{y}} \mid \tilde{\mathbf{x}}) f_{\mathbf{x}}(\tilde{\mathbf{x}})}{\int_{-\infty}^{\infty} f_{\mathbf{y}|\mathbf{x}}(\tilde{\mathbf{y}} \mid \tilde{\mathbf{x}}) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}}.$$

Posterior density

Likelihood Ratio Test

- The hypothesis that a given pattern **x** belongs one of N_c classes is tested to minimize the probability of error.

$$\text{If } f_{w_1|\mathbf{x}}(w_1 \mid \tilde{\mathbf{x}}) = \frac{f_{\mathbf{x}|w_1}(\tilde{\mathbf{x}} \mid w_1)P(w_1)}{f_{\mathbf{x}}(\tilde{\mathbf{x}})} > f_{w_2|\mathbf{x}}(w_2 \mid \tilde{\mathbf{x}}) = \frac{f_{\mathbf{x}|w_2}(\tilde{\mathbf{x}} \mid w_2)P(w_2)}{f_{\mathbf{x}}(\tilde{\mathbf{x}})} \quad \Rightarrow \quad w_1 \text{ class,}$$

$$\text{else } \ w_2 \ \text{ class.}$$

- The decision rule can be $L(\tilde{\mathbf{x}}) \triangleq \dfrac{f_{\mathbf{x}|w_1}(\tilde{\mathbf{x}} \mid w_1)}{f_{\mathbf{x}|w_2}(\tilde{\mathbf{x}} \mid w_2)} \underset{w_2}{\overset{w_1}{\gtrless}} \dfrac{P(w_2)}{P(w_1)}.$

Likelihood Ratio

- Example: the conditional p.d.f. are $\begin{cases} f_{\mathbf{x}|w_1}(\tilde{\mathbf{x}} \mid w_1) = \frac{1}{\sqrt{2\pi}}\exp[-\frac{1}{2}(x-4)^2] \\ f_{\mathbf{x}|w_2}(\tilde{\mathbf{x}} \mid w_2) = \frac{1}{\sqrt{2\pi}}\exp[-\frac{1}{2}(x-10)^2] \end{cases}.$

$P(w_1) = P(w_2).$

$$\Rightarrow \quad L(\tilde{\mathbf{x}}) \triangleq \exp[-\tfrac{1}{2}(x-4)^2 + \tfrac{1}{2}(x-10)^2] \underset{w_2}{\overset{w_1}{\gtrless}} 1 \quad \Rightarrow \quad (x-4)^2 - (x-10)^2 \underset{w_2}{\overset{w_1}{\lessgtr}} 0.$$

$$\Rightarrow \quad \tilde{\mathbf{x}} = x \underset{w_2}{\overset{w_1}{\lessgtr}} 7.$$

⇒ Usually, the **log likelihood Ratio** is used.

Probability of Misclassification

- The probability of error, i.e. Bayes risk, determines the quality of a decision rule.
  - A lower value implies a better rule.

$$P(error) = \int_{-\infty}^{\infty} P(error \mid \mathbf{x}) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = P(error \mid w_1) P(w_1) + P(error \mid w_2) P(w_2).$$

$$\varepsilon_1 = P(error \mid w_1) = P(\text{choose } w_2 \mid w_1) = \int_{R_2} f_{\mathbf{x}|w_1}(\mathbf{x} \mid w_1) d\mathbf{x}.$$

$$\varepsilon_2 = P(error \mid w_2) = P(\text{choose } w_1 \mid w_2) = \int_{R_1} f_{\mathbf{x}|w_2}(\mathbf{x} \mid w_2) d\mathbf{x}.$$

- Bayes risk under Multiple hypotheses can be defined:

$$P(error \mid \mathbf{x}) = \sum_{j=1, j \neq i}^{N_c} P(w_j \mid \mathbf{x}) = 1 - P(w_i \mid \mathbf{x}), \quad \text{if } \mathbf{x} \in R_i.$$

  - R_i should be defined to be the region where $P(w_i|\mathbf{x})$ is largest.

$$\text{Choose } w_i \text{ where } P(w_i \mid \mathbf{x}) = \max_j [P(w_j \mid \mathbf{x})].$$

Distance Functions

- There are several ways to measure the distance $d(\mathbf{x},\mathbf{y})$ between two vectors **x** & **y**.
  - Generally, a distance function is any scalar-valued function satisfying the following conditions:
    - $d(\mathbf{x},\mathbf{y})>0$ for $\mathbf{x}\neq\mathbf{y}$; $d(\mathbf{x},\mathbf{y})=0$ if $\mathbf{x}=\mathbf{y}$.
    - $d(\mathbf{x},\mathbf{y})= d(\mathbf{y},\mathbf{x})$.
    - [Triangular inequality] $d(\mathbf{x},\mathbf{y})+d(\mathbf{y},\mathbf{z}) \geq d(\mathbf{x},\mathbf{z})$.
  - Euclidean distance: $d_E(\mathbf{x},\mathbf{y}) = |\mathbf{x}-\mathbf{y}| = \left(\sum_{i=1}^{n}(x_i - y_i)^2\right)^{\frac{1}{2}}.$
  - Maximum value distance: $d_M(\mathbf{x},\mathbf{y}) = \max_i |x_i - y_i|.$
  - Absolute value distance (city block): $d_A(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n}|x_i - y_i|.$

Linear Transformation

- If **x** is a vector in X and **y** is the corresponding (mapped) vector in Y, then **y** = A **x**.
- Matrix A is said to be **positive definite**
  - if the quadratic product, $\mathbf{x}^T A\mathbf{x}$, is strictly greater than zero for all non-zero vector **x**.

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j .$$

- Matrix A is said to be **positive semidefinite**
  - if the quadratic product, $\mathbf{x}^T A\mathbf{x}$, ≥ 0 for all non-zero vector **x**.

- Example: the positive definite matrix $\mathbf{A} = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}.$

$$\Leftrightarrow \quad \mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j = x_1^2 - 2x_1x_2 + 2x_2^2 = (x_1 - x_2)^2 + x_2^2 > 0, \forall \mathbf{x} \neq \mathbf{0}.$$

$\mathbf{B} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & -1 \\ 3 & 3 & 0 \end{pmatrix}$ is not a positive definite matrix.

Differentiation w.r.t. Vectors

- If s is a scalar function of a vector **x** ∈ R^n,
  - the derivative of s w.r.t. **x** is defined as the vector (**gradient**)

$$\frac{ds}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial s}{\partial x_1} & \frac{\partial s}{\partial x_2} & \cdots & \frac{\partial s}{\partial x_n} \end{bmatrix}^T .$$

- If **s** is a vector ∈ R^m,
  - the derivative of s w.r.t. **x** is the matrix $\frac{d\mathbf{s}}{d\mathbf{x}} = \begin{pmatrix} \frac{\partial s_1}{\partial x_1} & \cdots & \frac{\partial s_m}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_1}{\partial x_n} & \cdots & \frac{\partial s_m}{\partial x_n} \end{pmatrix}.$

- If **r** is a linear transformation of **s**,
  - Namely, **r**=**Bs**, then $\frac{d\mathbf{r}}{d\mathbf{x}} = \mathbf{B}\frac{d\mathbf{s}}{d\mathbf{x}}.$

Chained rule

- If **r** is a nonlinear transformation of **s**, $\frac{d\mathbf{r}}{d\mathbf{x}} = \frac{d\mathbf{r}}{d\mathbf{s}}\frac{d\mathbf{s}}{d\mathbf{x}}.$

- For the quadratic product, $\frac{d(\mathbf{x}^{\mathrm{T}}\mathbf{B}\mathbf{x})}{d\mathbf{x}} = \cdots = (\mathbf{B}+\mathbf{B}^{\mathrm{T}})\mathbf{x} \underset{\uparrow}{=} 2\mathbf{B}\mathbf{x}.$

If **B** is symmetric

Correlation and Covariance Matrices

- The correlation matrix **R** of a random vector **x** is

$$\mathbf{R} = E(\mathbf{x}\mathbf{x}^{\mathrm{T}}) = (r_{ij}), \quad r_{ij} = E(x_i x_j).$$

- The covariance matrix **K** of a random vector **x** is

$$\mathbf{K} = E[(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^{\mathrm{T}}] = (k_{ij}), \quad k_{ij} = E[(x_i - m_i)(x_j - m_j)].$$

  - The diagonal elements k_{ii} are the variances of the vector components.

$$k_{ii} = \sigma_i^2 = Var(x_i) = E[(x_i - m_i)^2].$$

$$\begin{aligned}\mathbf{K} &= E[(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^{\mathrm{T}}] = E[\mathbf{x}\mathbf{x}^{\mathrm{T}} - \mathbf{x}\mathbf{m}^{\mathrm{T}} - \mathbf{m}\mathbf{x}^{\mathrm{T}} + \mathbf{m}\mathbf{m}^{\mathrm{T}}] \\ &= E[\mathbf{x}\mathbf{x}^{\mathrm{T}}] - E[\mathbf{x}]\mathbf{m}^{\mathrm{T}} - \mathbf{m}E[\mathbf{x}^{\mathrm{T}}] + \mathbf{m}\mathbf{m}^{\mathrm{T}} = E[\mathbf{x}\mathbf{x}^{\mathrm{T}}] - \mathbf{m}\mathbf{m}^{\mathrm{T}} = \mathbf{R} - \mathbf{m}\mathbf{m}^{\mathrm{T}}.\end{aligned}$$

$$\Rightarrow \quad \mathbf{R} = \mathbf{K} + \mathbf{m}\mathbf{m}^{\mathrm{T}}.$$

- Given a vector **y = A x**, $\mathbf{m}_{\mathbf{y}} = E[\mathbf{A}\mathbf{x}] = \mathbf{A}E[\mathbf{x}] = \mathbf{A}\mathbf{m}_{\mathbf{x}}.$

$$\mathbf{R}_{\mathbf{y}} = E[(\mathbf{A}\mathbf{x})(\mathbf{A}\mathbf{x})^T] = \mathbf{A}E[\mathbf{x}\mathbf{x}^T]\mathbf{A}^T = \mathbf{A}\mathbf{R}_{\mathbf{x}}\mathbf{A}^T. \qquad \mathbf{K}_{\mathbf{y}} = \mathbf{A}\mathbf{K}_{\mathbf{x}}\mathbf{A}^T.$$

  - If **A** is orthogonal, i.e. has orthonormal column vectors,

$$\left|\mathbf{R}_{\mathbf{y}}\right| = \left|\mathbf{R}_{\mathbf{x}}\right|, tr(\mathbf{R}_{\mathbf{y}}) = tr(\mathbf{R}_{\mathbf{x}}), \left|\mathbf{K}_{\mathbf{y}}\right| = \left|\mathbf{K}_{\mathbf{x}}\right|, tr(\mathbf{K}_{\mathbf{y}}) = tr(\mathbf{K}_{\mathbf{x}}).$$

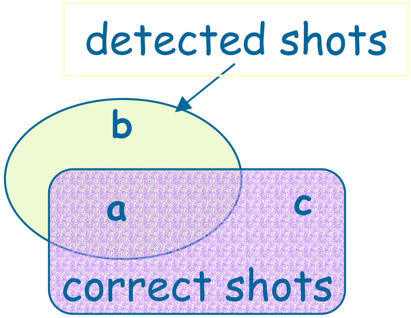
Independent Random Vectors

- If random vectors **x** and **y** are independent, they are uncorrelated.
  - The converse is generally not true.
  - Both are uncorrelated if $\begin{cases} \mathbf{R_{xy}} = E(\mathbf{x}\mathbf{y}^{\mathrm{T}}) = E(\mathbf{x})E(\mathbf{y}^{\mathrm{T}}) = \mathbf{m_x}\mathbf{m_y}^T, \\ \mathbf{K_{xy}} = E[(\mathbf{x}-\mathbf{m_x})(\mathbf{y}-\mathbf{m_y})^{\mathrm{T}}] = \mathbf{0}. \end{cases}$

- Gaussian Random Vectors:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \Rightarrow \quad f(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\sqrt{|\mathbf{K_x}|}} \exp[-\frac{(\mathbf{x}-\mathbf{m})\mathbf{K_x}^{-1}(\mathbf{x}-\mathbf{m})^T}{2}].$$

- Recall: a/(a+c), better for a smaller value of c.
  - ✓ The ratio of the number of shots **detected correctly** over the actual number of shots.
- Precision: a/(a+b), better for a smaller value of b.
  - ✓ The ratio of the number of shots detected correctly over the total number of **shots detected**.



Support Vector Machines (SVM)

- SVM is a novel kind of Neural Networks.
  - Multi-Layer-Perceptron (MLP): Classifier, regressor, etc.
    - Single-layer & Multi-layer with feed-forward connections.
    - Back propagation algorithm, maximum likelihood principle.
    - Training, self-structured: supervised, unsupervised.
    - The performance is justified by a loss function (say, MSE) over unseen samples of the test set.
      - The **expected** risk of the classifier on the test set [2] $\leq$ The **empirical** risk on the training set [0] + the **estimation** error [1].

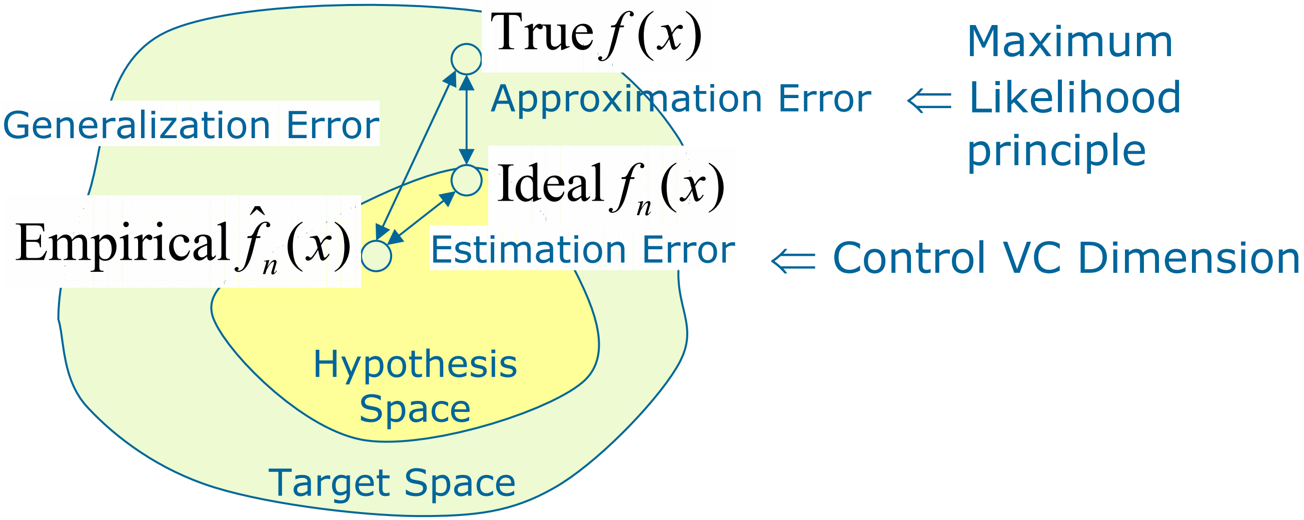
$$\text{Estimation error} \simeq \sqrt{\frac{h}{c}\log(1+2\frac{c}{h})}, \quad \begin{array}{l} c = |\text{Training Set}|, \\ h = \text{VC dimension of the classifer.} \end{array}$$

(Vapnik-Chervonenkis)= the maximal number of samples correctly classified in the training set.

[0] & [1] should be both min.

✓ Minimizing [0] alone do no good!!

Graphical Illustration



- In Modeling, the approximation error stems from the model mismatch.
  - The true f(x) may lie outside the hypothesis space.
- In Learning, the estimation error occurs due to the imperfect learning procedure.
  - The non-optimal model (empirically obtained) may be chosen.
- During the testing (evaluation), the generalization error is met.

- SVM minimizes the Expected risk by controlling VC dimension.
  - Learning becomes solving the problem of Quadratic Programming.

SVM = Optimal Hyperplane Algorithm

- Learning how to classify is estimating a function f: $R^n \rightarrow \pm 1$ over the training data set = $\{(x_i, y_i) \in R^n \times \pm 1 : i=1\ldots c)\}$
  - f will correctly classify other unseen example (x, y) under the same unknown probability distribution P(x, y). ⇒ namely, f(x)=y.
  - It is often assumed the data are i.i.d. (identically independent distributed).

$$\text{Hyperplanes}: w \cdot x + b = 0, \quad \begin{matrix} w \in R^n, \\ b \in R. \end{matrix} \quad \Leftrightarrow \quad \text{Decision Functions}: f(x) = \operatorname{sgn}(w \cdot x + b).$$

$$\Rightarrow \quad \exists \text{ a unique hyperplane}(w,b) \ni \max_{w,b} \min_i \left(\|x - x_i\| : x \in R^n, w \cdot x + b = 0, i = 1 \cdots c \right).$$

Maximize the separation margin.

⇒ Optimization problem: $\begin{cases} \min L(w) = \frac{1}{2}\|w\|^2 \\ y_i(w \cdot x_i + b) \geq 1, i = 1 \cdots c. \end{cases}$

Good separation

Correct

⇒ Solution = the saddle point of the Lagrangian: $\alpha_i \geq 0$

$$L(w,b,\alpha) = \tfrac{1}{2}\|w\|^2 - \sum_{i=1}^{c} \alpha_i [y_i (w \cdot x_i + b) - 1].$$

Minimized w.r.t. w & b, maximized w.r.t. α_i.

Solution

$$\begin{cases} \frac{\partial L(w,b,\alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{c} \alpha_i y_i = 0, \\ \frac{\partial L(w,b,\alpha)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{c} \alpha_i y_i x_i. \end{cases}$$

$\alpha_i \neq 0$ ⇒ Lying on the margin
Support vectors

The solution vector is a linear combination of a subset of the training patterns.

⇒ Support vectors summarize the information.

$$\Rightarrow \quad b = -\tfrac{1}{2} w \cdot (x_p + x_q), \alpha_p > 0, \alpha_q > 0, y_p = 1, y_q = -1, \text{for any SV } x_p, x_q.$$

- However, most classification problems are not linear separable.
  - Transform x_i to a high-dimension space to regain linear separation.

$$x \quad \Rightarrow \quad \Phi(x).$$

Φ(x) is hard to compute.

$$\text{Decision Functions}: f(x) = \text{sgn}(w \cdot x + b) = \text{sgn}\left(\sum_{i=1}^{c} \alpha_i y_i (x_i \cdot x) + b\right)$$

$$\Rightarrow \quad f(x) = \text{sgn}\left(\sum_{i=1}^{c} \alpha_i y_i (\Phi(x_i) \cdot \Phi(x)) + b\right).$$

The scalar (inner) product, $\Phi(x_i) \cdot \Phi(x)$, is easy to compute by a simple kernel.

As an example, the polynomial kernel $k(x,y) = \Phi(x) \cdot \Phi(y) = (x \cdot y)^d$.

✓ Matrices (K_{ij}) are positive definite, where $K_{ij} = k(x_i,x_j)$, i, j=1⋯c.

Dilemma

- Typically, the data will only be linearly separable in some, possibly very high dimensional space.
  - Separating the data exactly, particularly for a finite amount of data with noise, is favorable. However it will generalize badly.
  - In practice, it may be necessary to employ the non-separable approach (allow some classification error).

- To allow some overlapping between classes, the slack variables $\tau_i \geq 0$ is introduced.

$\Rightarrow$ Optimization problem: $\begin{cases} \min L(w,\tau) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{c}\tau_i, \quad C \text{ is some constant} \geq 0. \\ y_i(w\cdot x_i + b) \geq 1 - \tau_i, \quad i = 1\cdots c. \qquad \tau_i \geq 0 \end{cases}$

$$\Rightarrow \quad L(w,b,\alpha,\tau,\beta) = \tfrac{1}{2}\|w\|^2 - \sum_{i=1}^{c}\alpha_i[y_i(w\cdot x_i + b) - 1 + \tau_i] - \sum_{i=1}^{c}\beta_i\tau_i.$$

Saddle points

$$C \geq \alpha_i \geq 0 \quad \sum_{i=1}^{c}\alpha_i y_i = 0, \quad w = \sum_{i=1}^{c}\alpha_i y_i x_i, \quad \alpha_i + \beta_i = C.$$

(Any SV x_i has τ_i=0.)

$$\Rightarrow \quad \max_{\alpha} W(\alpha) = \sum_{i=1}^{c}\alpha_i - \tfrac{1}{2}\sum_{i=1}^{c}\sum_{j=1}^{c}\alpha_i\alpha_j y_i y_j (x_i\cdot x_j).$$

Non-linear Separation

- Using the kernel,

$$\Rightarrow \quad \max_{\alpha} W(\vec{\alpha}) = \sum_{i=1}^{c} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{c} \sum_{j=1}^{c} \alpha_i \alpha_j y_i y_j k(x_i, x_j).$$

- Define the matrix Q, $Q_{ij} = y_i y_j k(\vec{x}_i, \vec{x}_j). \Rightarrow \quad W(\vec{\alpha}) = \vec{\alpha}^T \vec{1} - \tfrac{1}{2} \vec{\alpha}^T Q \vec{\alpha}$

$$= \vec{\alpha}^T (\vec{1} - \tfrac{1}{2} Q \vec{\alpha}).$$

- Decomposition: break the entire training set into smaller ones.
  - Select the working (active) subset.
    - Other α_i are fixed in the current iteration.
  - Shrink the problem.
    - There are much less SVs than c.
    - Many SVs have α_i = C.
    - Caching and incremental updates of the gradient & the termination criteria.

Generalized Discriminant Analysis (GDA)

- GDA is the eigenvalue problem resolution for nonlinear discriminant analysis.
  - It is similar in functionality to SVM.
- The input set X has m vectors, $x_1 \cdots x_m$, belong to n classes, $X_1 \cdots X_n$.
  - The cardinality of the subset X_i is m_i. ⇒ $X = \bigcup_{i=1}^{n} X_i, \quad \sum_{i=1}^{n} m_i = m.$
  - The covariance matrix C of all x_i: $C = \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T.$

Suppose $x \Rightarrow \Phi(x).$ $C \Rightarrow V = \frac{1}{m}\sum_{i=1}^{m} \Phi(x_i)\Phi^T(x_i).$

Center $\Phi(x_i)$ in the transform space: $\tilde{\Phi}(x_i) = \Phi(x_i) - \frac{1}{m}\sum_{k=1}^{m} \Phi(x_k).$

- The **inter-class** inertia B is the covariance matrix of the class centers.

$$B = \frac{1}{m}\sum_{i=1}^{n} m_i \bar{\Phi}_i \bar{\Phi}_i^T, \quad \bar{\Phi}_i = \frac{1}{m_i}\sum_{k=1}^{m_i} \Phi(x_{i,k}).$$

The kth vector of Class i is $x_{i,k}$.

Likewise, $V = \frac{1}{m}\sum_{i=1}^{m} \Phi(x_i)\Phi^T(x_i) = \frac{1}{m}\sum_{i=1}^{n}\sum_{k=1}^{m_i} \Phi(x_{i,k})\Phi^T(x_{i,k}).$ **total inertia**

Formulation

- Using Kernel function: $k(x_i, x_j) = k_{i,j} = \Phi^T(x_i)\Phi(x_j).$

  For classes p & q, $\left(k_{i,j}\right)_{p,q} = \Phi^T(x_{p,i})\Phi(x_{q,j}).$

- Define a mxm matrix K: $K = \left(K_{p,q}\right)_{p,q=1\cdots n}, \quad K_{p,q} = \left(k_{i,j}\right)_{i=1\cdots m_p, j=1\cdots m_q} = K^T_{q,p}.$

  A mxm block diagonal matrix W: $W = \left(W_t\right)_{t=1\cdots n}, \quad W_t = \left(\frac{1}{m_t}\right)_{m_t \times m_t}.$

- The classical criteria for class separability is defined by the quotient between the inter-class inertia and the intra-classes inertia.
  - Its maximization is equivalent to the eigenvalue resolution.

- Assume the classes follow a multivariate Gaussian distribution, and each observation can be assigned to the class having the maximum posterior probability using the **Mahalanobis distance**.

Eigenvalue Resolution

- Given two symmetric matrices A & B with the same size, and B^{-1} exists,
  - The quotient $\dfrac{v^T Av}{v^T Bv}$ is maximal for eigenvector v of $B^{-1}A$ associated to the large eigenvalue λ.

Since

$$\frac{(v^T Bv)(2Av)-(v^T Av)(2Bv)}{(v^T Bv)^2}=0 \Rightarrow B^{-1}Av=\left(\frac{v^T Av}{v^T Bv}\right)v.$$

$$\Rightarrow \left(\frac{v^T Av}{v^T Bv}\right): \text{eigenvalue}, v: \text{eigenvector of } B^{-1}A.$$

Therefore, the quotient $\dfrac{B}{V}$ of both inertia's in the problem is maximized:

$$\begin{matrix}\lambda Vv = Bv \\ \lambda v = V^{-1}Bv\end{matrix} \quad \Rightarrow \left(\frac{v^T Bv}{v^T Vv}\right)=\lambda: \text{the largest eigenvalue}, v: \text{eigenvector of } V^{-1}B.$$

$$\begin{cases} B=\frac{1}{m}\sum_{i=1}^{n} m_i \bar{\Phi}_i \bar{\Phi}_i^T, \quad \bar{\Phi}_i=\frac{1}{m_i}\sum_{k=1}^{m_i}\Phi(x_{i,k}). \\ V=\frac{1}{m}\sum_{i=1}^{m}\Phi(x_i)\Phi^T(x_i)=\frac{1}{m}\sum_{i=1}^{n}\sum_{k=1}^{m_i}\Phi(x_{i,k})\Phi^T(x_{i,k}). \end{cases}$$

$$v=\sum_{i=1}^{n}\sum_{k=1}^{m_i}\alpha_{i,k}\Phi(x_{i,k}).$$

Linear combination

Formulation

$$\alpha = \left(\alpha_i\right)_{i=1\cdots n}, \quad \alpha_i = \left(\alpha_{i,k}\right)_{k=1\cdots m_i} \quad \Rightarrow \quad \left(\frac{v^T Bv}{v^T Vv}\right) = \lambda = \left(\frac{\alpha^T KWK\alpha}{\alpha^T KK\alpha}\right).$$

- Proof:

$$\boxed{\begin{aligned} &K = \left(K_{p,q}\right)_{p,q=1\cdots n}, \\ &K_{p,q} = \left(k_{i,j}\right)_{i=1\cdots m_p,\, j=1\cdots m_q} = K^T_{q,p}. \end{aligned}}$$

$$\lambda Vv = Bv \quad \Rightarrow \quad \lambda \Phi^T(x_{r,s})Vv = \Phi^T(x_{r,s})Bv.$$

$$\boxed{\left(k_{i,j}\right)_{p,q} = \Phi^T(x_{p,i})\Phi(x_{q,j}).}$$

$$\begin{aligned} Vv &= \tfrac{1}{m}\sum\nolimits_{p=1}^{n}\sum\nolimits_{i=1}^{m_p}\Phi(x_{p,i})\Phi^T(x_{p,i}) \times \sum\nolimits_{q=1}^{n}\sum\nolimits_{k=1}^{m_q}\alpha_{q,k}\Phi(x_{q,k}) \\ &= \tfrac{1}{m}\sum\nolimits_{q=1}^{n}\sum\nolimits_{k=1}^{m_q}\alpha_{q,k}\sum\nolimits_{p=1}^{n}\sum\nolimits_{i=1}^{m_p}\Phi(x_{p,i})\Phi^T(x_{p,i})\Phi(x_{q,k}). \end{aligned}$$

$$\lambda\Phi^T(x_{r,s})Vv = \tfrac{\lambda}{m}\sum\nolimits_{q=1}^{n}\sum\nolimits_{k=1}^{m_q}\alpha_{q,k}\sum\nolimits_{p=1}^{n}\sum\nolimits_{i=1}^{m_p}[\Phi^T(x_{r,s})\Phi(x_{p,i})][\Phi^T(x_{p,i})\Phi(x_{q,k})].$$

$$\Rightarrow \quad \lambda[\Phi^T(x_{1,m_1}),\cdots,\Phi^T(x_{m_1,m_1}),\cdots,\Phi^T(x_{1,m_n}),\cdots,\Phi^T(x_{m_n,m_n})]Vv = \tfrac{\lambda}{m}KK\alpha.$$

$$\begin{aligned} Bv &= \tfrac{1}{m}\sum\nolimits_{p=1}^{n} m_p\left[\tfrac{1}{m_p}\sum\nolimits_{i=1}^{m_p}\Phi(x_{p,i})\right]\left[\tfrac{1}{m_p}\sum\nolimits_{i=1}^{m_p}\Phi(x_{p,i})\right]^T \times \sum\nolimits_{q=1}^{n}\sum\nolimits_{k=1}^{m_q}\alpha_{q,k}\Phi(x_{q,k}) \\ &= \tfrac{1}{m}\sum\nolimits_{q=1}^{n}\sum\nolimits_{k=1}^{m_q}\alpha_{q,k}\sum\nolimits_{p=1}^{n}\left[\sum\nolimits_{i=1}^{m_p}\Phi(x_{p,i})\right]\left[\tfrac{1}{m_p}\right]\left[\sum\nolimits_{i=1}^{m_p}\Phi^T(x_{p,i})\Phi(x_{q,k})\right]. \end{aligned}$$

$$\Rightarrow \quad [\Phi^T(x_{1,m_1}),\cdots,\Phi^T(x_{m_1,m_1}),\cdots,\Phi^T(x_{1,m_n}),\cdots,\Phi^T(x_{m_n,m_n})]Bv = \tfrac{1}{m}KWK\alpha.$$

Eigenvalue Resolution

- By the eigenvectors decomposition of K, $K = P\Gamma P^T$
  - P contains the normalized eigenvectors, say v.
    - P is orthonormal since K is symmetric.
  - Γ is the diagonal matrix with non-zero eigenvalues.

$$\left(\frac{v^T B v}{v^T V v}\right) = \lambda = \left(\frac{\alpha^T KWK\alpha}{\alpha^T KK\alpha}\right) = \frac{\alpha^T \left(P\Gamma P^T\right) W \left(P\Gamma P^T\right)\alpha}{\alpha^T \left(P\Gamma P^T\right)\left(P\Gamma P^T\right)\alpha} = \frac{\left(\Gamma P^T\alpha\right)^T P^T W P \left(\Gamma P^T \alpha\right)}{\left(\Gamma P^T\alpha\right)^T P^T P\left(\Gamma P^T\alpha\right)}.$$

$$\beta = \Gamma P^T \alpha \quad \Rightarrow \quad \lambda P^T P\beta = \lambda\beta = P^T W P \beta \quad \Rightarrow \quad \alpha = P\Gamma^{-1}\beta.$$

Also, $1 = v^T v = \sum_{p=1}^{n} \sum_{k=1}^{m_p} \alpha_{p,k}\Phi^T(x_{p,k}) \sum_{q=1}^{n} \sum_{k=1}^{m_q} \alpha_{q,k}\Phi(x_{q,k})$

$$= \sum_{p=1}^{n} \sum_{q=1}^{n} \alpha_p^T K_{p,q}\alpha_q = \alpha^T K\alpha \quad \Rightarrow \quad \alpha \text{ should be normalized by } \sqrt{\alpha^T K \alpha}.$$

Given a test vector z, the projections can be computed as

$$v^T z = \sum_{p=1}^{n}\sum_{k=1}^{m_p}\alpha_{p,k}\Phi^T(x_{p,k})\, z = \sum_{p=1}^{n}\sum_{k=1}^{m_p}\alpha_{p,k} k(x_{p,k}, z).$$

Summary

- GDA procedure is summarized in the following steps:

1. Compute K and W. $\left(k_{i,j}\right)_{p,q} = \Phi^T(x_{p,i})\Phi(x_{q,j}).$

$$K = \left(K_{p,q}\right)_{p,q=1\cdots n}, K_{p,q} = \left(k_{i,j}\right)_{i=1\cdots m_p, j=1\cdots m_q}. \quad W = \left(W_t\right)_{t=1\cdots n}, \quad W_t = \left(\tfrac{1}{m_t}\right)_{m_t \times m_t}.$$

2. Decompose K using eigenvectors decompositions.

$$K = P\Gamma P^T$$

3. Compute eigenvectors β and eigenvalues of the system.

$$\beta = \Gamma P^T \alpha \quad \Rightarrow \quad \lambda P^T P\beta = \underline{\lambda\beta = P^T W P \beta} \quad \Rightarrow \quad \alpha = P\Gamma^{-1}\beta.$$

4. Compute eigenvectors v using α and normalize them.

$$v = \sum_{p=1}^{n}\sum_{k=1}^{m_p} \alpha_{p,k}\Phi(x_{p,k}). \qquad \alpha \text{ should be normalized by } \sqrt{\alpha^T K \alpha}.$$

5. Compute projections of test points onto the eigenvectors v.

$$v^T z = \sum_{p=1}^{n}\sum_{k=1}^{m_p} \alpha_{p,k}\Phi^T(x_{p,k})\, z = \sum_{p=1}^{n}\sum_{k=1}^{m_p} \alpha_{p,k} k(x_{p,k}, z).$$

Kernel Functions

- Various kernel functions can be used:
  - Gaussian kernel, RBF-kernel: $k(x,y)=\exp\left(\frac{-\|x-y\|^{d}}{2\sigma^{2}}\right).$
  - Polynomial kernel: $k(x,y)=(x\cdot y)^{d}.$

$$d=2 \quad \Rightarrow \quad (x_1^2,\cdots,x_t^2,x_1x_2,\cdots,x_ix_j,\cdots):\ \frac{t(t-1)}{2}\text{terms for } x\in R^{t}.$$

- Threshold values are learned and chosen.
  - The number of classes minus one is the number of thresholds chosen for classification.

Biased Discriminant Transform (BDT)

- MM information retrieval relies on the descriptors (or feature vectors), a set of real numbers.
  - Effectiveness of the representation in descriptors.
  - Selection of similarity metric.

- Difference between Traditional and MM DB:
  - Binary “Hit-or-Miss” decision using keywords in traditional DB.
    - The occurrences of the keywords or their synonyms, or
    - Rule-based ranking. etc.
  - In MMDB, the feature space is R^n (continuous).
    - Inherently, it is a nearest neighbor or a top-k ranking problem.

Why On-Line Learning?

- "Consensus" interpretation on MM contents:
  - Among all the users
  - Among all the times
    - ➢ The correct answer should match the context of conversation.
    - "The bat slipped from his hand." shows different meaning in the context of a baseball game or a cave exploring.
    - Medical image DB may define specific functionalities to perform off-line pre-clustering.
- On-learning is indispensable.
  - The system need to communicate with the user to perceive the specific goal of the queries.
    - In CBIR, a user is required to offer the feature-weighting scheme.
    - In "Relevance Feedback", a user is kept in the loop to tell the relevance of an image or video. (NO R/W of textual description)

Supervised Classification Problem

- One descriptor is assumed to represent the MM object.
  - By it, the media type becomes transparent to the system.
  - The object can be an whole image, image block, segmented region, shorts, frames, or a key frame.
  - ✓ A point is associated with the descriptor in the feature space.

- Relevance feedback: supervised classification problem.
  - Learning Speed: the number of iterations.
  - Training Size: the number of samples, i.e. their population.
    - Class density, positive/negative samples, etc.
  - Top-k returns: not a binary decision.
    - Binary classification (two-class) may not be optimal.
  - ➢ Initial results are returned; returns/evaluation are iterated.
    - ➢ The goal is to learn the discriminating **subspace**.

Variants of Relevance Feedback

- Objectives:
  - A user may look for a particular object or a similar one.
- Feedbacks:
  - A user may give back the positive feedback, negative, or both.
  - The degree of relevance for each result may be returned.
  - Partial likeness: it is like A in color, like B in shape, etc.
- Multiple Descriptors per Sample:
  - A mixed model can be used for refinement (intersection, union) to emphasize the local features.
- Class distribution:
  - Two or more target classes may be assumed.
    - Gaussian: two; Kernel-based: more for non-linearity.
- Data Organization:
  - A hierarchical tree structure may slow learning in real-time.
- Focus:
  - To learn a linear transformation, consider the correlations of feature components, estimate the class density, etc.

Fisher & Multiple Discriminant Analyses

- The consensus is to find the features to best cluster & separate the positive examples from the negative.
- Traditional approaches:
  - Two-class assumption (FDA): to find a lower dimensional space in which the ratio of between-class scatter over within-class scatter is maximized.

$$\mathbf{W} = \arg_{\mathbf{W}} \max \frac{\left|\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{b}}\mathbf{W}\right|}{\left|\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{W}}\mathbf{W}\right|}.$$

Large inter

$$\mathbf{S}_{\mathbf{b}} = (\mathbf{m}_{\mathbf{x}} - \mathbf{m})(\mathbf{m}_{\mathbf{x}} - \mathbf{m})^T + (\mathbf{m}_{\mathbf{y}} - \mathbf{m})(\mathbf{m}_{\mathbf{y}} - \mathbf{m})^T.$$

$$\mathbf{S}_{\mathbf{W}} = \sum_{i=1}^{N_x} (\mathbf{x}_{\mathbf{i}} - \mathbf{m}_{\mathbf{x}})(\mathbf{x}_{\mathbf{i}} - \mathbf{m}_{\mathbf{x}})^T + \sum_{i=1}^{N_y} (\mathbf{y}_{\mathbf{i}} - \mathbf{m}_{\mathbf{y}})(\mathbf{y}_{\mathbf{i}} - \mathbf{m}_{\mathbf{y}})^T.$$

Small intra

$\mathbf{x}_\mathbf{i}$: positive; $\mathbf{y}_\mathbf{i}$: negative.

  - Two-class assumption (MDA):

$$\mathbf{S}_{\mathbf{b}} = (\mathbf{m}_{\mathbf{x}} - \mathbf{m})(\mathbf{m}_{\mathbf{x}} - \mathbf{m})^T + \sum_{i=1}^{N_y} (\mathbf{y}_{\mathbf{i}} - \mathbf{m})(\mathbf{y}_{\mathbf{i}} - \mathbf{m})^T.$$

$$\mathbf{S}_{\mathbf{W}} = \sum_{i=1}^{N_x} (\mathbf{x}_{\mathbf{i}} - \mathbf{m}_{\mathbf{x}})(\mathbf{x}_{\mathbf{i}} - \mathbf{m}_{\mathbf{x}})^T.$$

Biased Discriminant Analysis (BDA)

- (1+x)-class assumption:
  - The user is only interested in one class, while there are an unknown number of other classes.
    - “All happy families are alike, each unhappy family is unhappy in its own fashion” - Leo Tolstoy's Anna Karenina.
    - All positive examples are alike in a way; each negative example is negative in its own way.

$$\mathbf{W} = \arg_{\mathbf{W}} \max \frac{\left|\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{y}}\mathbf{W}\right|}{\left|\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{x}}\mathbf{W}\right|}.$$

$$\mathbf{S}_{\mathbf{y}} = \sum_{i=1}^{N_y} (\mathbf{y}_{\mathbf{i}} - \mathbf{m}_{\mathbf{x}})(\mathbf{y}_{\mathbf{i}} - \mathbf{m}_{\mathbf{x}})^T.$$

$$\mathbf{S}_{\mathbf{x}} = \sum_{i=1}^{N_x} (\mathbf{x}_{\mathbf{i}} - \mathbf{m}_{\mathbf{x}})(\mathbf{x}_{\mathbf{i}} - \mathbf{m}_{\mathbf{x}})^T.$$

  - Regularization and Discounting Factors:
    - Sample-based estimates may be severely biased for small number of training examples.

$$\mathbf{S}_{\mathbf{x}}^{r} = (1-\mu)\mathbf{S}_{\mathbf{x}} + \frac{\mu}{n} tr[\mathbf{S}_{\mathbf{x}}]\mathbf{I}. \qquad \mathbf{S}_{\mathbf{y}}^{d} = (1-\gamma)\mathbf{S}_{\mathbf{y}} + \frac{\gamma}{n} tr[\mathbf{S}_{\mathbf{y}}]\mathbf{I}.$$

n=dim(original space).

Kernel-based BDA (KBDA)

- For non-linearity in the data, a non-linear mapping $\Phi:\mathbf{x}\rightarrow\Phi(\mathbf{x})$ is used to restore linearity in the transform space.
  - The evaluation of kernel K = (k$_{ij}$), where $k_{ij} = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$.

$$\mathbf{W} = \arg_{\mathbf{W}} \max \frac{\left|\mathbf{W}^T\mathbf{S}_{\mathbf{y}}^{\Phi}\mathbf{W}\right|}{\left|\mathbf{W}^T\mathbf{S}_{\mathbf{x}}^{\Phi}\mathbf{W}\right|}.$$

$$\mathbf{S}_{\mathbf{y}}^{\Phi} = \sum_{i=1}^{N_y}(\Phi(\mathbf{y_i})-\mathbf{m}_{\mathbf{x}}^{\Phi})(\Phi(\mathbf{y_i})-\mathbf{m}_{\mathbf{x}}^{\Phi})^T.$$

$$\mathbf{S}_{\mathbf{x}}^{\Phi} = \sum_{i=1}^{N_x}(\Phi(\mathbf{x_i})-\mathbf{m}_{\mathbf{x}}^{\Phi})(\Phi(\mathbf{x_i})-\mathbf{m}_{\mathbf{x}}^{\Phi})^T.$$

  - Let **w** is the eigenvector associated with the largest eigenvalue for **W**.

$$\mathbf{w} = \sum_{i=1}^{N_x}\alpha_i\Phi(\mathbf{x_i}) + \sum_{j=1}^{N_y}\alpha_{i+N_x}\Phi(\mathbf{y_i}) = \mathbf{\Phi\alpha}.$$

$$\mathbf{K}_{\mathbf{y_i}} = \mathbf{\Phi}^T\Phi(\mathbf{y_i}) = \Big(\underset{\underset{\mathrm{N\times N_y}}{\uparrow}}{\mathbf{K_y}}\Big)_{:,j},$$

$$\mathbf{K_{mx}} = \mathbf{\Phi}^T\mathbf{m}_{\mathbf{x}}^{\Phi},$$

$$\mathbf{I}_{\mathbf{N_x}}^{\mathbf{y}} = \frac{\mathbf{1}}{\mathbf{N_x}}(\mathbf{1})_{\mathbf{N_x\times N_y}}.$$

$$\mathbf{w}^T\mathbf{S}_{\mathbf{y}}^{\Phi}\mathbf{w} = \boldsymbol{\alpha}^T\mathbf{\Phi}^T\left[\sum_{i=1}^{N_y}(\Phi(\mathbf{y_i})-\mathbf{m}_{\mathbf{x}}^{\Phi})(\Phi(\mathbf{y_i})-\mathbf{m}_{\mathbf{x}}^{\Phi})^T\right]\mathbf{\Phi}\boldsymbol{\alpha}$$

$$= \boldsymbol{\alpha}^T\left[\sum_{i=1}^{N_y}(\mathbf{K}_{\mathbf{y_i}}-\mathbf{K_{mx}})(\mathbf{K}_{\mathbf{y_i}}-\mathbf{K_{mx}})^T\right]\boldsymbol{\alpha}$$

$$= \boldsymbol{\alpha}^T\left[(\mathbf{K_y}-\mathbf{K_x}\mathbf{I}_{\mathbf{N_x}}^{\mathbf{y}})(\mathbf{K_y}-\mathbf{K_x}\mathbf{I}_{\mathbf{N_x}}^{\mathbf{y}})^T\right]\boldsymbol{\alpha}.$$

KBDA (cont'd)

$$\mathbf{K}_{\mathbf{x}} = N \times N_x,$$

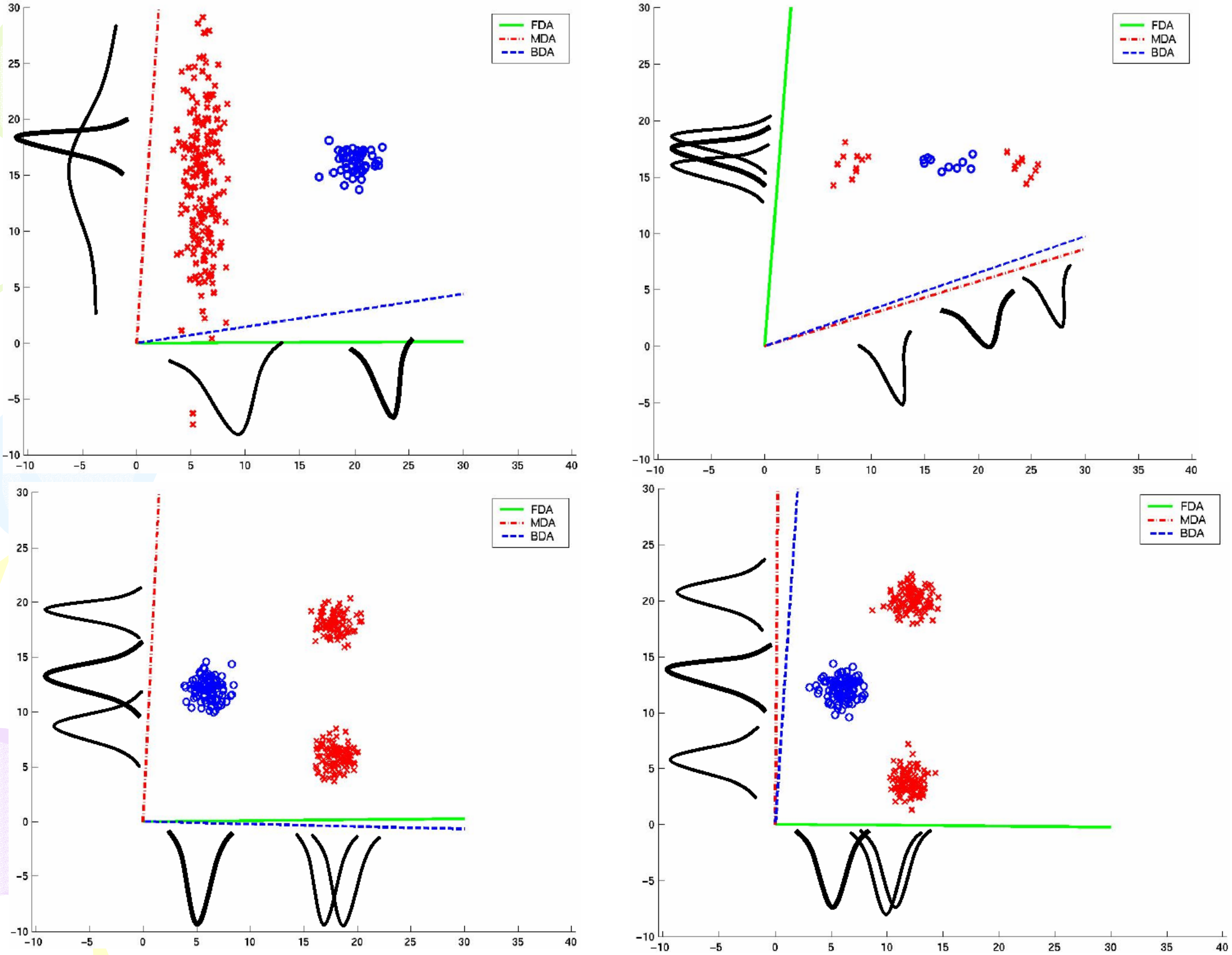
$$\mathbf{I}^{\mathbf{x}}_{\mathbf{N}_{\mathbf{x}}} = \frac{1}{\mathbf{N}_{\mathbf{x}}}(\mathbf{1})_{\mathbf{N}_{\mathbf{x}} \times \mathbf{N}_{\mathbf{x}}} .$$

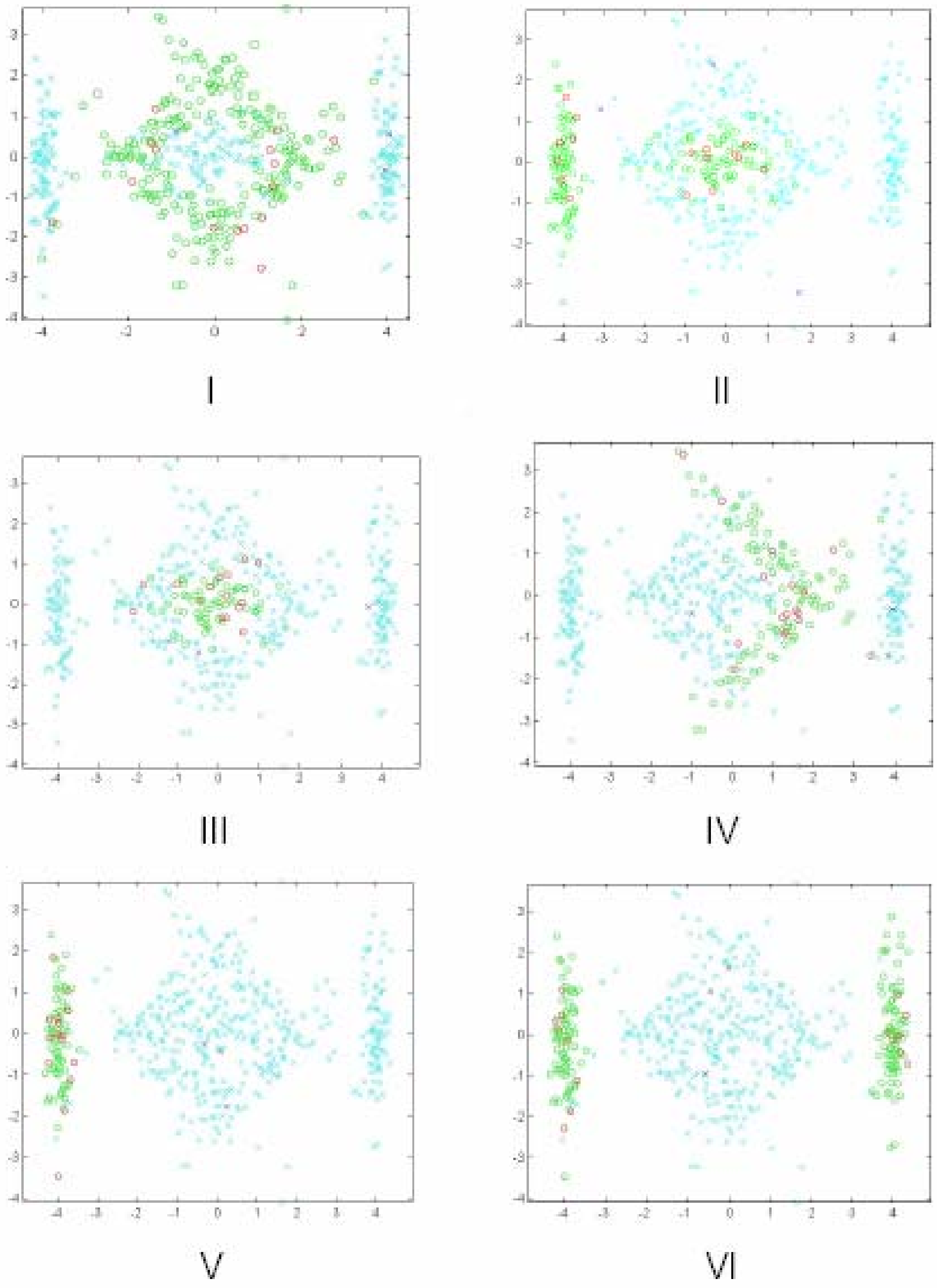
$$\begin{aligned}\mathbf{w}^T \mathbf{S}^{\Phi}_{\mathbf{x}} \mathbf{w} &= \boldsymbol{\alpha}^T \left[(\mathbf{K}_{\mathbf{x}} - \mathbf{K}_{\mathbf{x}} \mathbf{I}^{\mathbf{x}}_{\mathbf{N}_{\mathbf{x}}})(\mathbf{K}_{\mathbf{x}} - \mathbf{K}_{\mathbf{x}} \mathbf{I}^{\mathbf{x}}_{\mathbf{N}_{\mathbf{x}}})^T \right] \boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^T \mathbf{K}_{\mathbf{x}} \left[(\mathbf{I} - \mathbf{I}^{\mathbf{x}}_{\mathbf{N}_{\mathbf{x}}})(\mathbf{I} - \mathbf{I}^{\mathbf{x}}_{\mathbf{N}_{\mathbf{x}}})^T \right] \mathbf{K}^T_{\mathbf{x}} \boldsymbol{\alpha} \\ &= \boldsymbol{\alpha}^T \mathbf{K}_{\mathbf{x}} (\mathbf{I} - \mathbf{I}^{\mathbf{x}}_{\mathbf{N}_{\mathbf{x}}})^2 \mathbf{K}^T_{\mathbf{x}} \boldsymbol{\alpha}.\end{aligned}$$

- Solve to get α : the eigenvector with the largest eigenvalue.
- Given a new pattern **z**, find its projection onto **w** by

$$\mathbf{w}^T \Phi(\mathbf{z}) = \sum_{i=1}^{N_x} \alpha_i k(\mathbf{x}_{\mathbf{i}}, \mathbf{z}) + \sum_{j=1}^{N_y} \alpha_{i+N_x} k(\mathbf{y}_{\mathbf{i}}, \mathbf{z}).$$

- In this new space, the nearest neighbors of the positive centroid are returned in each iteration.
  - Combined with the subsequent feedbacks, the new nearest neighbors are output.


FDA
MDA
BDA
30
25
20
15
10
5
0
−5
−10
−10
−5
0
5
10
15
20
25
30
35
40



RBF-Kernel:

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|^2 / (2\,\sigma^2)\right) ;$$

Primal optimization problem:

$$\text{minimize} \qquad \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to} \qquad y_i \cdot ((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1, \qquad i = 1, \ldots, \ell.$$

Decision function:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} y_i \alpha_i \cdot (\mathbf{x} \cdot \mathbf{x}_i) + b\right)$$

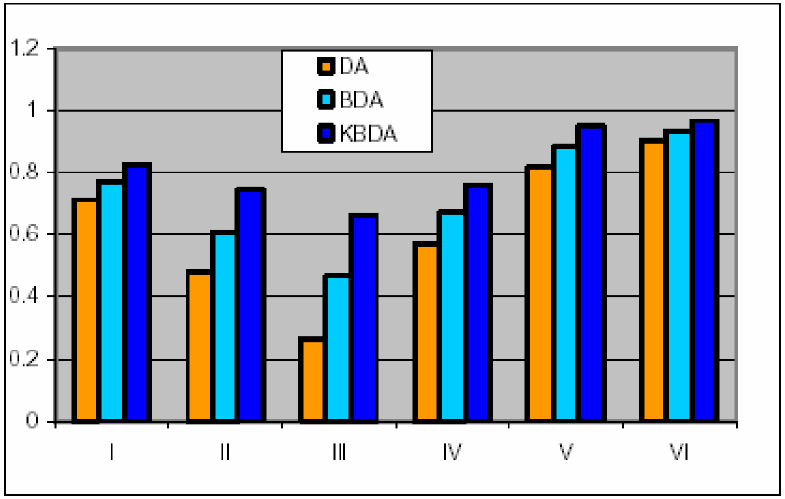


Figure 5 Test results on synthetic training data: six different configurations of non-linearity. The circles are positive examples and the crosses negative. A simulated query process is used for training sample selection, i.e., the 20 nearest neighbors of a randomly selected positive point are used as training samples. The bar diagram shows the averaged hit rate in top 20 returns.