

Pattern Recognition VI

Feature Selection & Extraction

Michal Haindl

Faculty of Information Technology, KTI
Czech Technical University in Prague
Institute of Information Theory and Automation
Czech Academy of Sciences
Prague, Czechia



<http://ro.utia.cz/>



MI/MIE-ROZ 2023-2024

© Haindl

MI-ROZ - 06

1/44

Outline

1 Feature Selection

- Probabilistic Distance Measures
- Probabilistic Dependence Measures
- Interclass Distance Measures
- Optimal Search Methods
- Branch-and-Bound Algorithm

2 Feature Extraction

- Karhunen-Loeve Expansion
- Linear Discriminant Analysis
- Sammon Projection



© Haindl

MI-ROZ - 06

2/44

Feature Selection / Extraction

motivation

• technical

- recognition problem dimensionality reduction ↘
- class separability increase ↗
- decreasing the overall measurement cost by eliminating redundant measurements ↘
- data compression (e.g., required communication channel capacity)
low orbit satellites circle planet 90 minutes $\approx$ 2 GB data - 10 minutes a communication window (only fraction of recorded data can be sent back)
- for a given amount of data,
number of features ↘ $\Rightarrow$ number of parameters ↘ $\wedge$
a performance estimate accuracy ↗
- information-rich features allow simplified classifiers
- classifier performance optimization - Hughes phenomenon

• physical

- physical measurement (e.g., RS soil moisture, vegetation cover)



© Haindl

MI-ROZ - 06

3/44

Feature Selection / Extraction

motivation

- technical - Hughes (peaking) phenomenon



• physical

- physical measurement (e.g., RS soil moisture, vegetation cover)
- scene-type invariant measurement (e.g. topography)
- data enhancement

• categorization

Goal classification vs. representation

Strategy selection vs. extraction

Approach filter vs. wrapper



© Haindl

MI-ROZ - 06

3/44

Feature Selection / Extraction

motivation

- technical
- physical
 - physical measurement (e.g., RS soil moisture, vegetation cover)
 - scene-type invariant measurement (e.g. topography)
 - data enhancement

• categorization

Goal classification vs. representation

Strategy selection vs. extraction

Approach filter vs. wrapper

Randomness non-deterministic vs. deterministic



Feature Selection / Extraction

motivation

- technical
- physical
 - physical measurement (e.g., RS soil moisture, vegetation cover)
 - scene-type invariant measurement (e.g. topography)
 - data enhancement
- categorization

Goal classification vs. representation

Strategy selection vs. extraction

Approach filter vs. wrapper

Randomness non-deterministic vs. deterministic

- ⊖ effective mathematical theory only for linear transformation Gaussian data



Feature Selection / Extraction

sensor $\rightarrow$ feature selector/extractor $\rightarrow$ classifier

- feature selection $\check{X} : \dim \check{X} < \dim X$
some information discarded

$$J(\check{X}) = \max_X J(X)$$

- feature extraction $\check{X} = f(X)$, $f : \tilde{I} \times I$, $\tilde{I} < I$
all information used, compression by mapping

$$J(\check{X}) = \max_f J(f(X))$$

feature learning - finding $f(\cdot)$



Feature Selection / Extraction

- feature selection $\check{X} : \dim \check{X} < \dim X$
some information discarded

$$J(\check{X}) = \max_{\check{X}} J(\check{X})$$

- feature extraction $\check{X} = f(X)$, $f : \tilde{I} \times I$, $\tilde{I} < I$
all information used, compression by mapping

$$J(\check{X}) = \max_f J(f(X))$$

feature learning - finding $f(\cdot)$



Feature Selection / Extraction

- feature **selection** $\ddot{X}$: $\dim \ddot{X} < \dim X$
some information discarded

$$J(\ddot{X}) = \max_{\ddot{X}} J(\dot{X})$$

- feature **extraction** $\ddot{X} = f(X)$, $f: \mathbb{R}^I \times I, \ddot{I} < I$
all information used, compression by mapping

$$J(\ddot{X}) = \max_f J(f(X))$$

feature learning - finding $f(\cdot)$



Feature Selection / Extraction

- feature **selection** $\ddot{X}$: $\dim \ddot{X} < \dim X$
some information discarded

$$J(\ddot{X}) = \max_{\ddot{X}} J(\dot{X})$$

- feature **extraction** $\ddot{X} = f(X)$, $f: \mathbb{R}^I \times I, \ddot{I} < I$
all information used, compression by mapping

$$J(\ddot{X}) = \max_f J(f(X))$$

feature learning - finding $f(\cdot)$

- ⊕ FSE - performance optimization
- ⊕ FS - measurement cost reduction
- ⊖ FSE - no direct relation with classification error
- ⊖ FE - no physical feature interpretation



Feature Selection

specification

- the feature evaluation criterion $J(X)$
- the dimensionality of the feature space I
- the optimization procedure
- the FE form of mapping $f(X)$ (extractor)

$J(X)$ defined in terms of unknown model characteristics
 $P(\omega_i), p(X|\omega_i) \rightarrow$ estimates

Filter $J(X)$ some univariate metric

Wrapper $J(X)$ internal cross-validation by some classification algorithm



Feature Selection

specification

- the feature evaluation criterion $J(X)$
- the dimensionality of the feature space I
- the optimization procedure
- the FE form of mapping $f(X)$ (extractor)

$J(X)$ defined in terms of unknown model characteristics
 $P(\omega_i), p(X|\omega_i) \rightarrow$ estimates

Filter $J(X)$ some univariate metric

Wrapper $J(X)$ internal cross-validation by some classification algorithm



Filter / Wrapper Methods

Filter

- $J(X)$ some univariate metric independent of the classification algorithm
- unimodal conditional densities can be assumed (probabilistic distance measures)
- ⊕ efficient & fast
- ⊕ generality - intrinsic data properties only (no tie to a classifier)
- ⊖ filter can miss a useful feature
- ⊖ tendency to select large subsets

Wrapper

Embedded

Hybrid



Filter / Wrapper Methods

Filter

Wrapper

- $J(X)$ cross-validation performed around a given classification algorithm (avoids overfitting)
- no information about conditional densities unimodality / complex structure
- ⊕ accuracy - better tuned to classifier and dataset interaction
- ⊖ computationally complex (cross-validation)
- ⊖ lack of generality - tied to the used classifier

Embedded

Hybrid



Filter / Wrapper Methods

Filter

Wrapper

Embedded

- feature selection is embedded within the classification algorithm (feature selection & learning)
 - Modified finite mixture with binary parameters $\Phi = \{\phi_1, \dots, \phi_I\}$
 - $\phi_i = 1$ for important features (own component), $\phi_i = 0$ unimportant features (shared density)
 - solution EM algorithm - $\hat{\theta}, \hat{\Phi}, \hat{p}$

Hybrid



Filter / Wrapper Methods

Filter

Wrapper

Embedded

Hybrid

- a combination of above approaches (good performance & numerical complexity)
 - 1 reduce number of features $\mathbf{Y}_d$ using $J_1(\cdot)$ for pre-filtering
 - 2 final features from $\mathbf{Y}_d$ using the main criterion $J_2(\cdot)$



Feature Selection Error Sources

- suboptimal criterion functions
- suboptimal search strategies
- pdf estimation errors (small sample size)
- numerical errors
- fitting errors



Optimal Feature Selection

ideal feature evaluation through classification error probability of the classifier being designed

$$P(\text{error}) = 1 - \int_S \max_{j=1,\dots,K} \{P(\omega_j)p(X|\omega_j)\} dX$$

ideal Bayesian rule cannot improve its performance by eliminating a feature (monotonicity)

nonideal Bayesian rule (assumptions not valid) can improve its performance

- ⊖ no known classifier exhibits monotonicity (except some linear ones)



Optimal Feature Selection

ideal feature evaluation through classification error probability of the classifier being designed

$$P(\text{error}) = 1 - \int_S \max_{j=1,\dots,K} \{P(\omega_j)p(X|\omega_j)\} dX$$

Open Feature Selection Problems

- ⊖ The search problem is NP hard, even moderate number of features requires suboptimal methods.
- ⊖ Given an optimal feature subset we are still unable to determine the usefulness (discriminatory power) of a particular feature.
- ⊖ If a FS criterion involves a classifier error rate (wrapper) → this classifier is optimized → the only computable error rate is an **apparent error** rate known to be very biased.



Feature Selection Approaches

- (Averaged) Probabilistic Distance Measures
- Probabilistic Dependence Measures
- Entropies item Interclass Distance Measures
- Feature-Set Search Algorithms
- Monte Carlo Techniques (simulated annealing), Genetic Algorithms



Probabilistic Distance Measures

$$K = 2$$

$$P(\text{error}) = \frac{1}{2} \left[1 - \int_S |P(\omega_1)p(X|\omega_1) - P(\omega_2)p(X|\omega_2)| dX \right]$$

$\max P(\text{error})$ if $p(X|\omega_i)$ completely **overlap**

similarly any measure between two pdf's

$$J(\ddot{X}) = \int_S f(P(\omega_i), p(X|\omega_i), i = 1, 2) dX$$

satisfying

- $J \geq 0$
- $J = 0$ for overlapping $p(X|\omega_i)$
- $J = \max$ for nonoverlapping $p(X|\omega_i)$

can be used for feature selection



Probabilistic Distance Measures

$$K = 2$$

$$P(\text{error}) = \frac{1}{2} \left[1 - \int_S |P(\omega_1)p(X|\omega_1) - P(\omega_2)p(X|\omega_2)| dX \right]$$

$\max P(\text{error})$ if $p(X|\omega_i)$ completely **overlap**

similarly any measure between two pdf's

$$J(\ddot{X}) = \int_S f(P(\omega_i), p(X|\omega_i), i = 1, 2) dX$$

satisfying

- $J \geq 0$
- $J = 0$ for overlapping $p(X|\omega_i)$
- $J = \max$ for nonoverlapping $p(X|\omega_i)$

can be used for feature selection



Probabilistic Distance Measures 2

if J can be expressed in the form of averaged f divergence i.e.,

$$J_F = \int f\left(\frac{P(\omega_1|X)}{P(\omega_2|X)}\right) P(\omega_2|X)p(X)dX$$

$f(s)$ convex function, $f_\infty = \lim_{s \rightarrow \infty} \frac{f(s)}{s}$ then

$$P(\text{error}) < \frac{f(0)P(\omega_2) + f_\infty P(\omega_1) - J_F}{f(0) + f_\infty - f(1)}$$

e.g., the averaged f divergence $\equiv$ the averaged Matusita distance $\bar{J}_T$

$$\bar{J}_T \rightarrow f(s) = (\sqrt{s}-1)^2, \quad J_F = \bar{J}_T^2, \quad f(0) = 1, \quad f(1) = 0, \quad f_\infty = 1$$

$$P(\text{error}) \leq \frac{1}{2}(1 - \bar{J}_T^2)$$



Probabilistic Distance Measures 2

if J can be expressed in the form of averaged f divergence i.e.,

$$J_F = \int f\left(\frac{P(\omega_1|X)}{P(\omega_2|X)}\right) P(\omega_2|X)p(X)dX$$

$f(s)$ convex function, $f_\infty = \lim_{s \rightarrow \infty} \frac{f(s)}{s}$ then

$$P(\text{error}) < \frac{f(0)P(\omega_2) + f_\infty P(\omega_1) - J_F}{f(0) + f_\infty - f(1)}$$

e.g., the averaged f divergence $\equiv$ the averaged Matusita distance $\bar{J}_T$

$$\bar{J}_T \rightarrow f(s) = (\sqrt{s}-1)^2, \quad J_F = \bar{J}_T^2, \quad f(0) = 1, \quad f(1) = 0, \quad f_\infty = 1$$

$$P(\text{error}) \leq \frac{1}{2}(1 - \bar{J}_T^2)$$



The Most Common Probabilistic Distance Measures

- Chernoff $s \in (0, 1)$

$$J_C = -\ln \int p^s(X|\omega_1) p^{1-s}(X|\omega_2) dX$$

- Bhattacharyya

$$J_B = -\ln \int [p(X|\omega_1)p(X|\omega_2)]^{1/2} dX$$

- Matusita

$$J_T = \left\{ \int [\sqrt{p(X|\omega_1)} - \sqrt{p(X|\omega_2)}]^2 dX \right\}^{1/2}$$

- Divergence

$$J_D = \int [p(X|\omega_1) - p(X|\omega_2)] \ln \frac{p(X|\omega_1)}{p(X|\omega_2)} dX$$

- Patrick-Fisher

$$J_P = \left\{ \int [p(X|\omega_1) - p(X|\omega_2)]^2 dX \right\}^{1/2}$$



The Most Common Probabilistic Distance Measures 2

averaged probability distance measures

$$J_X \rightarrow \bar{J}_X \quad p(X|\omega_i) \rightarrow p(X|\omega_i)P(\omega_i),$$

i.e.,

$$\bar{J}_P = \left\{ \int [p(X|\omega_1)P(\omega_1) - p(X|\omega_2)P(\omega_2)]^2 dX \right\}^{1/2}$$

⊕ useful for Gaussian classes



Gaussian Density

- Chernoff $s \in (0, 1)$

$$J_C = \frac{1}{2}s(1-s)(\mu_2 - \mu_1)^T [(1-s)\Sigma_1 + s\Sigma_2]^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \frac{|(1-s)\Sigma_1 + s\Sigma_2|}{|\Sigma_1|^{1-s} |\Sigma_2|^s}$$

- Bhattacharyya

$$J_B = \frac{1}{4}(\mu_2 - \mu_1)^T [\Sigma_1 + \Sigma_2]^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \frac{|\frac{1}{2}(\Sigma_1 + \Sigma_2)|}{\sqrt{|\Sigma_1||\Sigma_2|}}$$

- Matusita

$$J_T = \{2[1 - \exp\{-J_B\}]\}^{1/2}$$



Gaussian Density 2

- Divergence

$$J_D = \frac{1}{2}(\mu_2 - \mu_1)^T [\Sigma_1^{-1} + \Sigma_2^{-1}] (\mu_2 - \mu_1) + \frac{1}{2} \text{tr} \{ \Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2I \}$$

- Patrick-Fisher

$$J_P = \left[(2\pi)' |2\Sigma_1| \right]^{-1/2} + \left[(2\pi)' |2\Sigma_2| \right]^{-1/2} - 2 \left[(2\pi)' |\Sigma_1 + \Sigma_2| \right]^{-1/2} \exp \left\{ -\frac{1}{2}(\mu_2 - \mu_1)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_2 - \mu_1) \right\}$$



Gaussian Density 2

- Mahalanobis (if $\Sigma_1 = \Sigma_2 = \Sigma$)

$$J_M = J_D = 8J_B = (\mu_2 - \mu_1)^T \Sigma^{-1} (\mu_2 - \mu_1)$$

- ⊕ recursive evaluation (averaged PDM too)

$K > 2$ e.g.,

$$J = \sum_{i=1}^K \sum_{j=i}^K P(\omega_i) P(\omega_j) J_{i,j}$$

- ⊖ not retain a close relationship with error probability



Probabilistic Dependence Measures

if $p(X|\omega_i) = p(X) \Rightarrow X, \omega_i$ independent $\Rightarrow$ no learning about ω_i from X max J

dependence between random variable X and a realization of ω_i is measured by the distance between $p(X), p(X|\omega_i)$

if one of $p(X|\omega_i) \rightarrow p(X)$
all probabilistic distance measures suit overall dependence



Probabilistic Dependence Measures 2

- Chernoff $s \in (0, 1)$

$$J_C = -\ln \int p^s(X|\omega_1) p^{1-s}(X) dX$$

- Bhattacharyya

$$J_B = -\ln \int [p(X|\omega_1) p(X)]^{1/2} dX$$

- Matusita

$$J_T = \left\{ \int [\sqrt{p(X|\omega_1)} - \sqrt{p(X)}]^2 dX \right\}^{1/2}$$

- Divergence

$$J_D = \int [p(X|\omega_1) - p(X)] \ln \frac{p(X|\omega_1)}{p(X)} dX$$

- Patrick-Fisher

$$J_P = \left\{ \int [p(X|\omega_1) - p(X)]^2 dX \right\}^{1/2}$$



Gaussian Probabilistic Dependence Measures

- Chernoff $s \in (0, 1)$

$$J_C = \frac{1}{2} s(1-s) (\mu - \mu_1)^T [(1-s)\Sigma_1 + s\Sigma]^{-1} (\mu - \mu_1) + \frac{1}{2} \ln \frac{|(1-s)\Sigma_1 + s\Sigma|}{|\Sigma_1|^{1-s} |\Sigma|^s}$$

- Bhattacharyya

$$J_B = \frac{1}{4} (\mu - \mu_1)^T [\Sigma_1 + \Sigma]^{-1} (\mu - \mu_1) + \frac{1}{2} \ln \frac{|\frac{1}{2}(\Sigma_1 + \Sigma)|}{\sqrt{|\Sigma_1| |\Sigma|}}$$

- Matusita

$$J_T = \{2[1 - \exp\{-J_B\}]\}^{1/2}$$



Gaussian Probabilistic Dependence Measures

- Divergence

$$J_D = \frac{1}{2}(\mu - \mu_1)^T [\Sigma_1^{-1} + \Sigma^{-1}](\mu - \mu_1) + \frac{1}{2} \text{tr} \{ \Sigma_1^{-1} \Sigma + \Sigma^{-1} \Sigma_1 - 2I \}$$

- Patrick-Fisher

$$J_P = \left[(2\pi)' |2\Sigma_1| \right]^{-\frac{1}{2}} + \left[(2\pi)' |2\Sigma| \right]^{-\frac{1}{2}} - 2 \left[(2\pi)' |\Sigma_1 + \Sigma| \right]^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2}(\mu - \mu_1)^T (\Sigma_1 + \Sigma)^{-1} (\mu - \mu_1) \right\}$$



Entropy Measures

observe X and compute $P(\omega_i | X)$ to determine an information gain
if $P(\omega_i | X) = P(\omega_j | X) \quad \forall j \neq i$ then minimal information gain and
max. entropy (uncertainty)

average generalized entropy of degree α

$$J_E^\alpha = \int (2^{1-\alpha} - 1)^{-1} \left[\sum_{i=1}^K P^\alpha(\omega_i | X) - 1 \right] p(X) dX$$

- Shannon $\alpha = 1 \quad J_S(X^*) = \min_X J_S(X)$

$$J_S = - \int \sum_{i=1}^K P(\omega_i | X) \log_2 [P(\omega_i | X)] p(X) dX$$



Entropy Measures 2

- Bayesian distance $\alpha = 2$ (Quadratic entropy $\min J_C$) $\max J_Q$

$$J_Q = \int \sum_{i=1}^K P^2(\omega_i | X) p(X) dX$$

- ⊕ useful for multiclass problem, can be estimated using k-NNR



Interclass Distance Measures

$$J_\delta = \frac{1}{2} \sum_{i=1}^K P(\omega_i) \sum_{j=1}^K P(\omega_j) \frac{1}{\text{card}(T_i) \text{card}(T_j)} \sum_{k=1}^{\text{card}(T_i)} \sum_{l=1}^{\text{card}(T_j)} \delta(X_{i,k}, X_{j,l})$$

$\delta(X_{i,k}, X_{j,l})$ a metric measuring distance between k th pattern from ω_i and l th pattern from ω_j

- City block

$$\delta_C(X_i, X_k) = \sum_{j=1}^l |X_{i,j} - X_{k,j}|$$

- Euclidean

$$\delta_E(X_i, X_k) = \left[\sum_{j=1}^l (X_{i,j} - X_{k,j})^2 \right]$$



Interclass Distance Measures 2

$$\downarrow \mathcal{S}_W = \sum_{j=1}^K P(\omega_j) \Sigma_j$$

$$\uparrow \mathcal{S}_B = \sum_{j=1}^K P(\omega_j) (\mu_j - \mu)(\mu_j - \mu)^T$$

$$\mu = \sum_{j=1}^K P(\omega_j) \mu_j$$

measures derived from J_δ & Euclidean measure

$$J_1 = \text{tr}\{\mathcal{S}_W + \mathcal{S}_B\} = \text{tr}\{\mathcal{S}_T\}$$

$$J_2 = \frac{\text{tr} \mathcal{S}_B}{\text{tr} \mathcal{S}_W}$$

$$J_3 = \text{tr}\{\mathcal{S}_W^{-1} \mathcal{S}_B\}$$



Interclass Distance Measures 2

measures derived from J_δ & Euclidean measure

$$J_1 = \text{tr}\{\mathcal{S}_W + \mathcal{S}_B\} = \text{tr}\{\mathcal{S}_T\}$$

$$J_2 = \frac{\text{tr} \mathcal{S}_B}{\text{tr} \mathcal{S}_W}$$

$$J_3 = \text{tr}\{\mathcal{S}_W^{-1} \mathcal{S}_B\}$$

$$J_4 = \frac{|\mathcal{S}_W + \mathcal{S}_B|}{|\mathcal{S}_W|}$$

- ⊕ do not involve pdf → nonparametric applications
- ⊕ can be calculated recursively
- ⊖ heuristic notion, very loose relation with error probability



Optimal Search Methods - Direct Search

for a given $\ddot{l} < l$ direct search → evaluation of effectiveness of

$$\binom{l}{\ddot{l}} = \frac{l!}{(l-\ddot{l})! \ddot{l}!}$$



Optimal Search Methods - Direct Search

for a given $\ddot{l} < l$ direct search → evaluation of effectiveness of

$$\binom{l}{\ddot{l}} = \frac{l!}{(l-\ddot{l})! \ddot{l}!}$$

NASA Earth Observer 1 Hyperion - $l = 242$ spectral channels,
 $\ddot{l} = 10 \rightarrow \binom{242}{10} \approx 1,5 \times 10^{17}$

- ⊕ branch and bound, Monte Carlo methods, genetic algorithms
- ⊖ combinatorial problem excessive even for moderate $l, \ddot{l}$



Optimal Search Methods - Direct Search

exception $K = 2$, $X \in \mathcal{N}$, $\Sigma_i = \text{diag}$ the Mahalanobis distance is

$$J_M = \sum_{j=1}^{\tilde{l}} \frac{(\mu_{1,j} - \mu_{2,j})^2}{\sigma_{j,j}}$$

i.e. contribution of class separability of one measurement independent on other measurements $\Rightarrow$ optimal subset of features = $\tilde{l}$ individually best measurements

- 1 compute $J_M(x_j)$ $j = 1, \dots, l$
- 2 rank measurements in the decreasing magnitude of the criterion

$$J_M(x_1) > J_M(x_2) > \dots > J_M(x_l)$$

- 3 the best feature subset is defined by the first $\tilde{l}$ x_j

⊖ features cannot be in general selected based on their individual effectiveness even for independent features



Mutual Correlation Method

$$X_i = [{}^i x_1, \dots, {}^i x_d] \quad i = 1, \dots, n$$

K classes

$$r_{x_i, x_j} = \frac{\sum_k {}^k x_i {}^k x_j - n \bar{x}_i \bar{x}_j}{\sqrt{(\sum_k {}^k x_i^2 - n \bar{x}_i^2)(\sum_k {}^k x_j^2 - n \bar{x}_j^2)}}$$

Haindl, Michal - Somol, Petr - Ververidis, Dimitrios - Kotropoulos, Constantine, Lecture Notes in Computer

Science 4225, Springer, pp. 569 - 577, 2006.



Mutual Correlation Method

$$X_i = [{}^i x_1, \dots, {}^i x_d] \quad i = 1, \dots, n$$

K classes

$$r_{x_i, x_j} = \frac{\sum_k {}^k x_i {}^k x_j - n \bar{x}_i \bar{x}_j}{\sqrt{(\sum_k {}^k x_i^2 - n \bar{x}_i^2)(\sum_k {}^k x_j^2 - n \bar{x}_j^2)}}$$

If x_i and x_j independent $\Rightarrow r_{x_i, x_j} = 0$

Haindl, Michal - Somol, Petr - Ververidis, Dimitrios - Kotropoulos, Constantine, Lecture Notes in Computer

Science 4225, Springer, pp. 569 - 577, 2006.



Mutual Correlation Method

$$r_{j,\delta} = \frac{1}{\delta} \sum_{i=1, i \neq j}^{\delta} |r_{x_i, x_j}|$$

remove

$$\alpha = \arg \max_j r_{j,\delta}$$

correction

$$r_{j,\delta-1} = \frac{\delta r_{j,\delta} - |r_{x_\alpha, x_j}|}{\delta - 1}$$



Mutual Correlation Method

$$r_{j,\delta} = \frac{1}{\delta} \sum_{i=1, i \neq j}^{\delta} |r_{x_i, x_j}|$$

remove

$$\alpha = \arg \max_j r_{j,\delta}$$

correction

$$r_{j,\delta-1} = \frac{\delta r_{j,\delta} - |r_{x_\alpha, x_j}|}{\delta - 1}$$



Mutual Correlation Method

$$r_{j,\delta} = \frac{1}{\delta} \sum_{i=1, i \neq j}^{\delta} |r_{x_i, x_j}|$$

remove

$$\alpha = \arg \max_j r_{j,\delta}$$

correction

$$r_{j,\delta-1} = \frac{\delta r_{j,\delta} - |r_{x_\alpha, x_j}|}{\delta - 1}$$



Mutual Correlation Method - Algorithm

- 1 Initialize $\delta = d - 1$.
- 2 Discard feature x_α with the largest average mutual correlation.
- 3 Decrement $\delta = \delta - 1$, if $\delta < \ddot{l}$ return the resulting $\ddot{l}$ dimensional feature set and stop. Otherwise,
- 4 Recalculate the average correlations.
- 5 Go to step 2.

Haindl, Michal - Somol, Petr - Ververidis, Dimitrios - Kotropoulos, Constantine, Lecture Notes in Computer

Science 4225, Springer, pp. 569 - 577, 2006.



Branch-and-Bound Algorithm

assumption - can be used if a feature selection criterion satisfies the monotonicity property

monotonicity property - for nested feature sets X_j related

$$X_1 \subset X_2 \subset \dots \subset X_l$$

the criterion function $J(X_j)$ satisfies

$$J(X_1) \leq J(X_2) \leq \dots \leq J(X_l)$$

e.g. $K = 2$

$$J(X) = (\mu_1 - \mu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\mu_1 - \mu_2)$$



Branch-and-Bound Algorithm

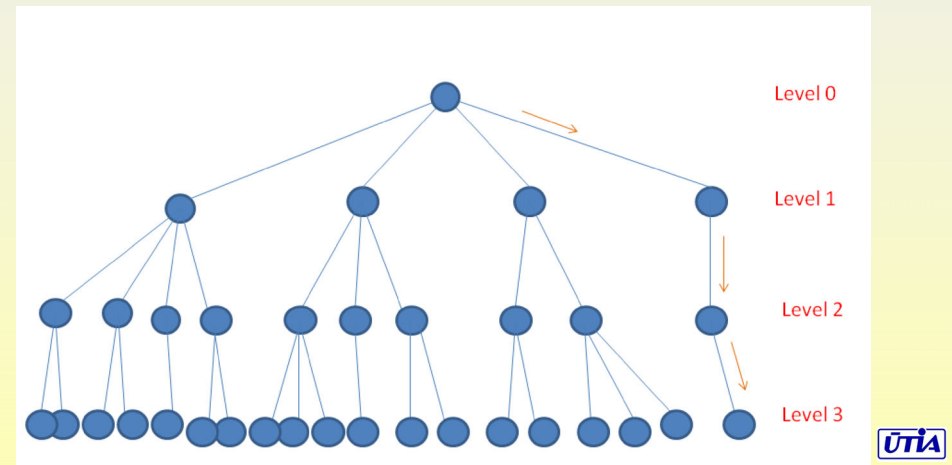
top down search with backtracking

- 1 All possible discarded features are represented in a tree the tree is scanned in top-down and right to left manner. The tree has $l - \bar{l}$ levels and $\binom{l}{i}$ number of branches.
- 2 If $J_{node} < J^*$ discard the whole branch, if $J_{node} > J^*$ continue along the branch.
- 3 If $J_{terminal\ node} > J^*$ update J^* .



Branch-and-Bound Algorithm 2

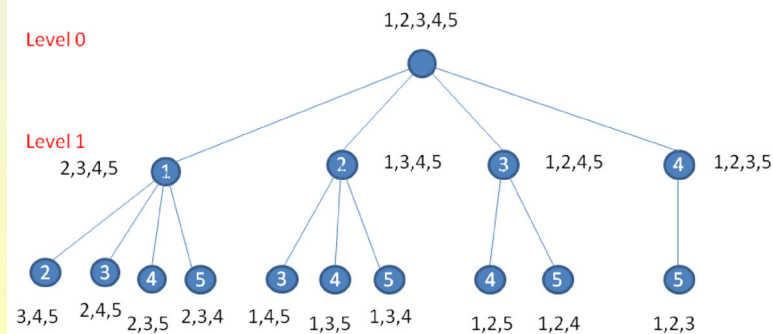
- ⊕ finds an optimal feature subset if monotonicity condition holds
- ⊕ effective search organization
- ⊖ prohibitive for large l



Branch-and-Bound Algorithm 2

- ⊕ finds an optimal feature subset if monotonicity condition holds
- ⊕ effective search organization
- ⊖ prohibitive for large l

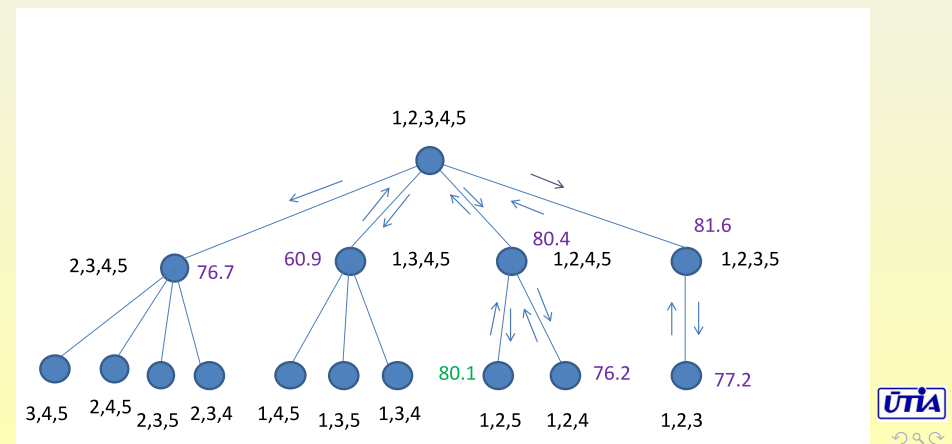
Example 3 from 5 $\binom{5}{3} = 10$



Branch-and-Bound Algorithm 2

- ⊕ finds an optimal feature subset if monotonicity condition holds
- ⊕ effective search organization
- ⊖ prohibitive for large l

Example 3 from 5 $\binom{5}{3} = 10 \rightarrow 7$



Sequential Forward / Backward Selection

- ⊖ suboptimal if branch-and-bound computationally unfeasible

Sequential Forward Selection (SFS) - bottom-up process
suppose k features were selected from X to form X_k , the $k + 1$ feature selected from remaining X so that

$$J(X_{k+1}) = \max J(X_k \cup y_j) \quad y_j \in X - X_k$$

$$X_0 \equiv \emptyset$$

- ⊖ not possible to remove a superfluous feature (as a result of including other measurement)



Sequential Forward / Backward Selection

- ⊖ suboptimal if branch-and-bound computationally unfeasible

Sequential Forward Selection (SFS) - bottom-up process
suppose k features were selected from X to form X_k , the $k + 1$ feature selected from remaining X so that

$$J(X_{k+1}) = \max J(X_k \cup y_j) \quad y_j \in X - X_k$$

$$X_0 \equiv \emptyset$$

- ⊖ not possible to remove a superfluous feature (as a result of including other measurement)



Sequential Forward / Backward Selection 2

Sequential Backward Selection (SBS) - top-down process
suppose k features were removed from the set of measurements X to form X_{l-k} , the $k + 1$ feature to be eliminated is chosen so that

$$J(X_{l-k-1}) = \max J(X_{l-k} - y_j) \quad y_j \in X_{l-k}$$

$$X_l \equiv X$$

- ⊖ once a measurement is discarded no revision possible
- ⊖ computationally more complex than SFS $l, l - 1, \dots, \tilde{l}$
- ⊕ continues monitoring of the amount of information loss



Monte Carlo Methods

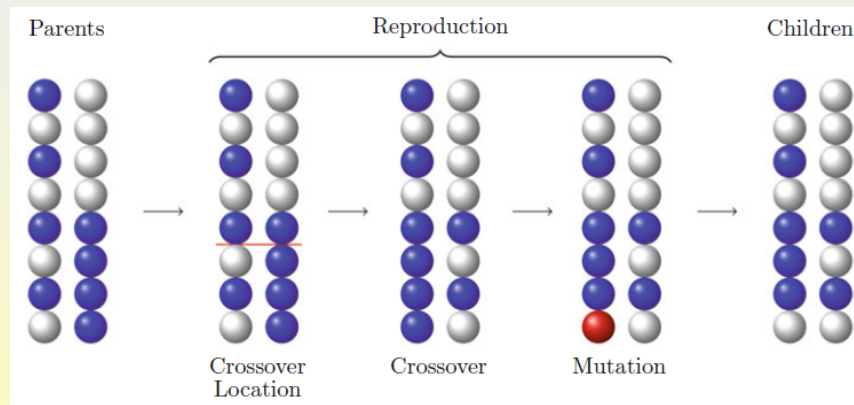
Simulated annealing

$\min J(X)$ (stochastic hill-climbing)

- 1 select an annealing schedule (T_i), select initial solution X_i
- 2 $X_i \rightarrow X_{i+1}$, $\Delta J = J(X_{i+1}) - J(X_i)$
- 3 accept change with the probability $P = \exp\{-\frac{\Delta J}{T_i}\}$



Genetic Algorithms



Genetic Algorithms

- 1 Define the stopping criterion, fitness function, population, probability of mutation p
- 2 Generate the starting population of m binary chromosomes, each of length $\bar{l}$
- 3 repeat
- 4 for each chromosome do - Train a model and compute each chromosome's fitness
- 5 for reproduction $k = 1, \dots, m/2$ do
- 6 Select two chromosomes based on the fitness function
- 7 Crossover: Randomly select a loci
- 8 Mutation: Randomly change binary value of each gene in each new child chromosome with probability p
- 9 end
- 10 until stopping criterion is met



Feature Extraction Approaches

- Probabilistic Distance Measures
- Karhunen-Loeve Expansion
- Linear Discriminant Analysis
- Sammon Projection



Feature Extraction

information compression by mapping

- nonlinear e.g. X generated by AR $\rightarrow$ parameter space
 - ⊖ computationally unfeasible, analytically untractable
- linear $\dot{X} = A^T X$



Feature Extraction

information compression by mapping

- nonlinear e.g. X generated by AR $\rightarrow$ parameter space
 $\ominus$ computationally unfeasible, analytically untractable
- linear $\ddot{X} = A^T X$



Parametric Measures

PDM & linear transformation

$$\ddot{\mu}_i = A^T \mu_i$$

$$\ddot{\Sigma}_i = A^T \Sigma_i A$$

Chernoff $s \in (0, 1)$

$$J_C(A) = \frac{1}{2} s(1-s)(\mu_2 - \mu_1)^T A [(1-s)A^T \Sigma_1 A + sA^T \Sigma_2 A]^{-1} A^T (\mu_2 - \mu_1) \\ + \frac{1}{2} \ln \frac{|(1-s)A^T \Sigma_1 A + sA^T \Sigma_2 A|}{|A^T \Sigma_1 A|^{1-s} |A^T \Sigma_2 A|^s}$$

optimal solution - numerical search in the gradient direction $J'_C(A)$

$$\text{if } \Sigma_1 = \Sigma_2 = \Sigma \Rightarrow A^* = \Sigma^{-1}(\mu_2 - \mu_1)$$

$$\text{if } \mu_2 = \mu_1 \Rightarrow A^* \text{ matrix of ranked eigenvectors of } \Sigma_2^{-1} \Sigma_1$$



Karhunen-Loeve Expansion

a projection of X onto the K-L coordinate system

& subsequent approximation of X with $\ddot{X}, \ddot{l} < l$

$u_j, j = 1, \dots, \infty$ - the complete set orthonormal basis vectors, i.e.

$$u_j^T u_i = \delta_{ij}$$

$$X = \sum_{j=1}^{\infty} x_j u_j$$

$$\ddot{X} = \sum_{j=1}^{\ddot{l}} x_j u_j$$

$$\{u_1, \dots, u_j\} :$$

$$\epsilon_{\min} = E\{(X - \ddot{X})^T (X - \ddot{X})\}$$

$$\epsilon = E\left\{\sum_{j=\ddot{l}+1}^{\infty} x_j^2\right\} = \sum_{j=\ddot{l}+1}^{\infty} u_j^T E\{XX^T\} u_j = \sum_{j=\ddot{l}+1}^{\infty} u_j^T \Phi u_j$$

Φ symmetric, PD $\ddot{l} \times \ddot{l} \Rightarrow$ only $\ddot{l}$ independent solutions, i.e.



Karhunen-Loeve Expansion 2

$$\ddot{X} = \sum_{j=1}^{\ddot{l}} x_j u_j = A^T X$$

u_j, λ_j are eigenvectors, eigenvalues of Φ

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{\ddot{l}} \geq \dots \geq \lambda_l$$

$$\epsilon = \sum_{j=\ddot{l}+1}^{\infty} u_j^T \lambda_j u_j = \sum_{j=\ddot{l}+1}^l \lambda_j$$

$$\epsilon_{\min} \rightarrow \lambda_{\ddot{l}+1}, \dots, \lambda_l \text{ minimal}$$

$$\text{Mahalanobis distance } (\ddot{X}, \ddot{\mu}) \quad J_M = \sum_{j=1}^{\ddot{l}} \frac{(\ddot{x}_j - \ddot{\mu}_{i,j})^2}{\lambda_j}$$



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\tilde{x}_i$ mutually uncorrelated ($E\{\tilde{x}_i\tilde{x}_j\} = \lambda_j\delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \tilde{x}_i$ independent
- ⊕ $\tilde{\Sigma} = A\Sigma A^T$ if Σ is the K-L generation matrix then $\tilde{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_J]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \tilde{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊕ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\tilde{x}_i$ mutually uncorrelated ($E\{\tilde{x}_i\tilde{x}_j\} = \lambda_j\delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \tilde{x}_i$ independent
- ⊕ $\tilde{\Sigma} = A\Sigma A^T$ if Σ is the K-L generation matrix then $\tilde{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_J]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \tilde{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊕ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\tilde{x}_i$ mutually uncorrelated ($E\{\tilde{x}_i\tilde{x}_j\} = \lambda_j\delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \tilde{x}_i$ independent
- ⊕ $\tilde{\Sigma} = A\Sigma A^T$ if Σ is the K-L generation matrix then $\tilde{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_J]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \tilde{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊕ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\tilde{x}_i$ mutually uncorrelated ($E\{\tilde{x}_i\tilde{x}_j\} = \lambda_j\delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \tilde{x}_i$ independent
- ⊕ $\tilde{\Sigma} = A\Sigma A^T$ if Σ is the K-L generation matrix then $\tilde{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_J]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \tilde{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊕ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\ddot{x}_i$ mutually uncorrelated ($E\{\ddot{x}_i \ddot{x}_j\} = \lambda_j \delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \ddot{x}_i$ independent
- ⊕ $\ddot{\Sigma} = A \Sigma A^T$ if Σ is the K-L generation matrix then $\ddot{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_j]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \ddot{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊕ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\ddot{x}_i$ mutually uncorrelated ($E\{\ddot{x}_i \ddot{x}_j\} = \lambda_j \delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \ddot{x}_i$ independent
- ⊕ $\ddot{\Sigma} = A \Sigma A^T$ if Σ is the K-L generation matrix then $\ddot{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_j]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \ddot{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊕ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\ddot{x}_i$ mutually uncorrelated ($E\{\ddot{x}_i \ddot{x}_j\} = \lambda_j \delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \ddot{x}_i$ independent
- ⊕ $\ddot{\Sigma} = A \Sigma A^T$ if Σ is the K-L generation matrix then $\ddot{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_j]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \ddot{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊕ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\ddot{x}_i$ mutually uncorrelated ($E\{\ddot{x}_i \ddot{x}_j\} = \lambda_j \delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \ddot{x}_i$ independent
- ⊕ $\ddot{\Sigma} = A \Sigma A^T$ if Σ is the K-L generation matrix then $\ddot{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_j]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \ddot{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊕ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\ddot{x}_i$ mutually uncorrelated ($E\{\ddot{x}_i \ddot{x}_j\} = \lambda_j \delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \ddot{x}_i$ independent
- ⊕ $\ddot{\Sigma} = A \Sigma A^T$ if Σ is the K-L generation matrix then $\ddot{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_j]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \ddot{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊖ weak relationship with error probability



Properties of the Karhunen-Loeve Expansion

- ⊕ K-L approximation guarantees minimal mean square error $\epsilon \forall$ approx.
- ⊕ $\ddot{x}_i$ mutually uncorrelated ($E\{\ddot{x}_i \ddot{x}_j\} = \lambda_j \delta_{i,j}$) $\rightarrow$ normalised Euclidean classifier, if $X \in \mathcal{N} \Rightarrow \ddot{x}_i$ independent
- ⊕ $\ddot{\Sigma} = A \Sigma A^T$ if Σ is the K-L generation matrix then $\ddot{\Sigma} = \text{diag}[\lambda_1, \dots, \lambda_j]$
- ⊕ information is compressed into a small number of K-L axes
- ⊕ correlation coefficient $\rho(x_i, \ddot{x}_j) = u_j(i) \sqrt{\frac{\lambda_j}{\Sigma_{i,i}}}$
- ⊕ keep overall data dispersion (average signal energy is invariant to K-L)
- ⊕ new axes rotate to the direction of the largest dispersion
- ⊕ minimizes the representation entropy
- ⊕ data visualization, visual enhancement
- ⊖ weak relationship with error probability



Linear Discriminant Analysis

$$\ddot{X} = A^T X$$

projection vectors which best discriminant between K classes

$$A^* = \arg \max_A \frac{|A^T S_B A|}{A^T S_W A}$$

$$A^* = [u_1, \dots, u_{K-1}]$$

$$S_W^{-1} S_B u_j = \lambda_j u_j \quad \forall j$$

u_j, λ_j are eigenvectors, eigenvalues of $S_W^{-1} S_B$



Sammon Projection

$$X_D \rightarrow Y_d \quad D \gg d$$

Sammon's error:

$$E = \frac{1}{\sum_{i < j} d_{ij}^*} \sum_{i < j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}$$

$$d_{ij}^* = \text{dist}[X_i, X_j] \quad i, j = 1, \dots, n$$

$$d_{ij} = \text{dist}[Y_i, Y_j] \quad i, j = 1, \dots, n$$

- trying to preserve the structure of inter-point distances in high-dimensional space in the lower-dimension projection



Sammon Projection

$$X_D \rightarrow Y_d \quad D \gg d$$

Sammon's error:

$$E = \frac{1}{\sum_{i < j} d_{ij}^*} \sum_{i < j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}$$

- 1 $Y_i^{(0)}$ random / orthogonal projection of D dimensional X_i into d dimensional space spanned by the d original coordinates with the largest variances
- 2 $Y_i^{(t)} \quad t = 1, \dots \quad \forall i$ such that $E^{(t)} < E^{(t-1)}$



Sammon Projection

$$X_D \rightarrow Y_d \quad D \gg d$$

Sammon's error:

$$E = \frac{1}{\sum_{i < j} d_{ij}^*} \sum_{i < j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}$$

- ⊕ exploratory data analysis
- ⊖ numerical minimization (e.g. gradient descent),
- ⊖ $\frac{n(n-1)}{2}$ interdistance matrix has to be stored,
- ⊖ resulting mapping is a table not a function,

