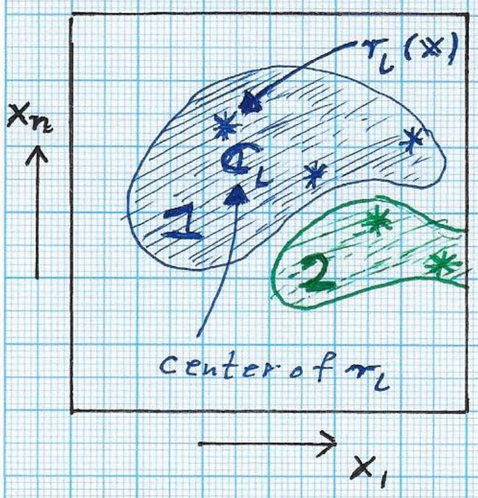
RBFs AND CLASSIFICATION – Cont'd.

→ Some RBF-related parameter optimizations:

Feature space



- For the computation of any BEST approximation one must attempt to optimize:
  - → no. of RBFs
  - → type of RBFs
  - → locations/centers of RBFs
  - → RBF-specific constants
  - → definition of finite compact supports of RBFs
  - → definition of metric tensors/distances for RBFs

Compute $f(x) = \sum_{l=1}^{L} w_l r_l(x)$

1) NUMBER: Start with a small no. of RBFs; optimize these initial RBFs; add RBFs if necessary (adaptively)

2) TYPE: i) Hardy's multiquadric or reciprocal multiquadric

$$r(x) = \left(C^2 + d^2(x) \right)^{+/- \frac{1}{2}}$$

(+ multiquadric, − reciprocal)

ii) Duchon's thin plate spline (TPS)

$$r(x) = d^2(x) \ln(d(x))$$

RBFs AND CLASSIFICATION – Cont'd.

iii) Gaussian function

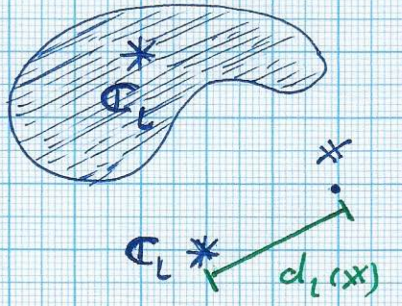
$$r(x) = e^{-C^2/d^2(x)}$$

iv) Polynomial

$$r(x) = \begin{cases} d(x) & \text{– linear} \\ d^3(x) & \text{– cubic} \end{cases}$$

here:
- C^2 positive constant (optimized)
- $d(x)$ (positive) distance between x and center of r

3) CENTERS: Each RBF $r_L(x)$ has a center c_L needed for "radial distance" $d_L(x)$ computation:

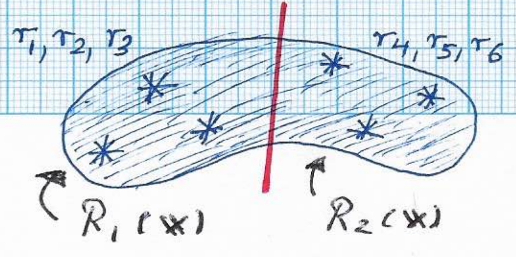


$$d_L(x) = \| x - c_L \| .$$

(An optimized quadratic form can be used to define $\|\cdot\|$.)

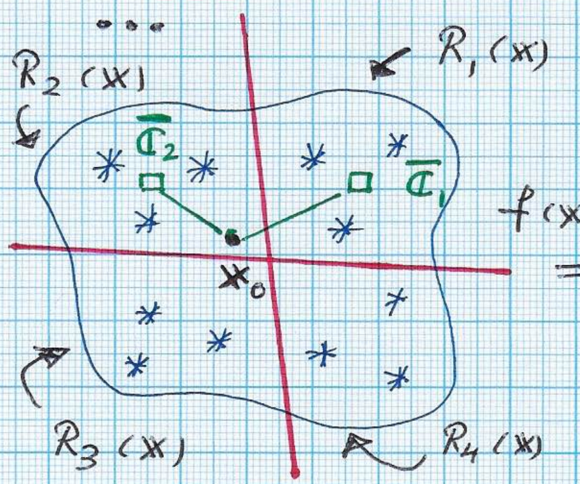
4) CONSTANTS: C^2 should be based on feature point distribution and values.

5) COMPACT SUPPORT: Compute multiple local data approximations using a "small" number of RBFs ⇒ Efficiency!



Ex: $R_1(x) = \sum_1^3 w_L r_L(x)$, $R_2(x) = \sum_4^6 w_L r_L(x)$

▪ RBFs AND CLASSIFICATION – Cont'd.



⇒ must blend/combine Local approximations (i.e., functions!) using "proper" weight functions $W(x)$:

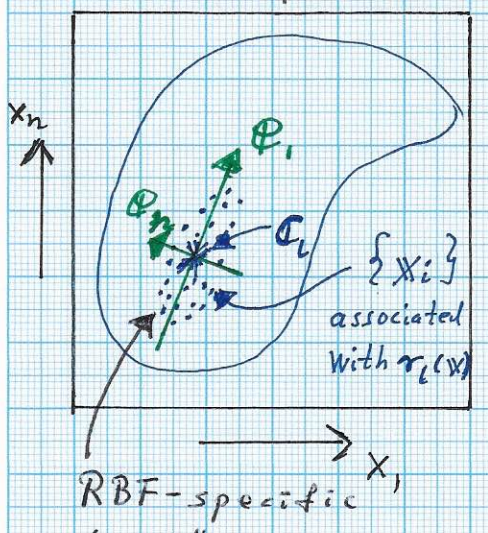
• e.g., $\overline{c}_I$, $\overline{c}_J$ centroids of RBF center point sets of $R_J(x)$ and $R_I(x)$; compute distances d_I and d_J

$$d_I = \| x_0 - \overline{c}_I \|$$

$$d_J = \| x_0 - \overline{c}_J \|$$

$$\Rightarrow f(x_0) = \left(\frac{1}{d_I} R_I(x_0) + \frac{1}{d_J} R_J(x_0) \right) \Big/ \left(\frac{1}{d_I} + \frac{1}{d_J} \right)$$

6) DISTANCE: A distance measure / metric for computing distance values between a point $x = (x_1, \ldots, x_n)^T$ should ideally be "affinely invariant": The locations and distribution of the feature points in $\{x_i\}$, with c_l as center of $r_l(x)$, define a covariance matrix with – generally – n real eigenvalues λ_i and mutually orthogonal eigenvectors e_i, $i = 1 \ldots n$.

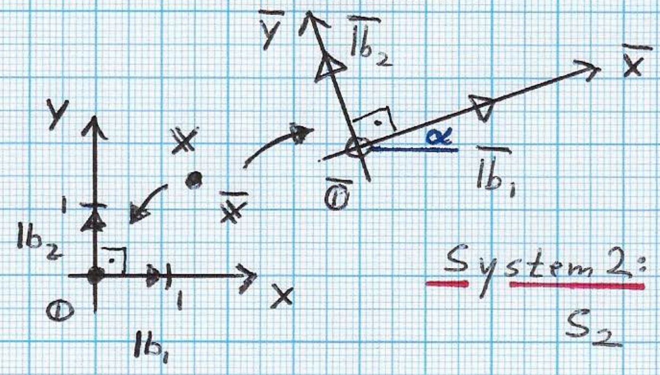


RBF-specific Local "coordinate system" (to be optimized) for "ideal" distance $d(x)$

Stratovan

▪ RBFs AND CLASSIFICATION – Cont'd.

... 6) **DISTANCE:** 2D motivating example...



System 2: S_2

System 1: S_1

- given: $\mathbb{x}$ expressed relative to S_1
- wanted: $\bar{\mathbb{x}}$ expressed relative to S_2
- Mapping of S_1 to S_2: $\mathbb{o} \mapsto \bar{\mathbb{o}}$, $\mathbb{b}_i \mapsto \bar{\mathbb{b}}_i$
- three elementary maps: scale, rotate, translate

- Matrix notation for elementary maps:

$$\text{Scale } S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{Rotate } R = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

homogeneous notation

$$\text{Translate } T = \begin{pmatrix} 1 & 0 & \bar{o}_x \\ 0 & 1 & \bar{o}_y \\ 0 & 0 & 1 \end{pmatrix}$$

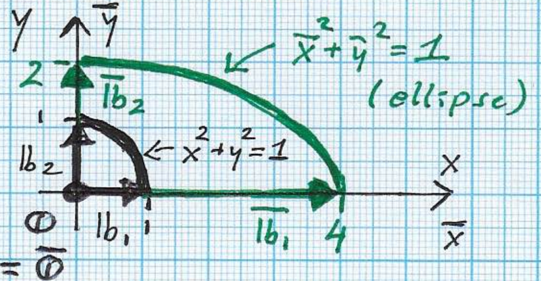
- "Theorem": Apply the inverse operations/matrices in reverse order to $\mathbb{x}_{S_1} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ to obtain $\mathbb{x}_{S_2} = \begin{pmatrix} \bar{x} \\ \bar{y} \\ 1 \end{pmatrix}$:

$$\begin{pmatrix} \bar{x} \\ \bar{y} \\ 1 \end{pmatrix} = \begin{pmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\alpha & s\alpha & 0 \\ -s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\bar{o}_x \\ 0 & 1 & -\bar{o}_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\Longleftrightarrow \underline{\underline{\bar{\mathbb{x}} = S^{-1} R^{-1} T^{-1} \mathbb{x}}}$$

▪ RBFs AND CLASSIFICATION – Cont'd.

... 6) **DISTANCE:** Ex: Scaling basis vectors



S_1: $\mathbb{0}$, $\mathbb{b}_1$, $\mathbb{b}_2$

S_2: $\bar{\mathbb{0}} = \mathbb{0}$, $\overline{\mathbb{b}}_1 = \begin{pmatrix}4\\0\end{pmatrix}$, $\overline{\mathbb{b}}_2 = \begin{pmatrix}0\\2\end{pmatrix}$

$\Rightarrow$ Scale $S = \begin{pmatrix}4&0&0\\0&2&0\\0&0&1\end{pmatrix}$

$$S^{-1} = \begin{pmatrix}1/4&0&0\\0&1/2&0\\0&0&1\end{pmatrix}$$

$$\Rightarrow \bar{\mathbb{x}} = S^{-1}\mathbb{x} = \begin{pmatrix}1/4&0&0\\0&1/2&0\\0&0&1\end{pmatrix}\begin{pmatrix}x\\y\\1\end{pmatrix} = \begin{pmatrix}x/4\\y/2\\1\end{pmatrix}$$

$\Rightarrow$ "Unit contour" in S_2: $\bar{x}^2 + \bar{y}^2 = 1$

$$\Leftrightarrow \left(\tfrac{x}{4}\right)^2 + \left(\tfrac{y}{2}\right)^2 = 1$$

! CIRCULAR unit contour $\Rightarrow$ ELLIPTICAL unit contour

$\Rightarrow$ Definition of squared length of the position vector associated with point $\bar{\mathbb{x}}$, relative to origin $\bar{\mathbb{0}}$:

$$\|\bar{\mathbb{x}}\|^2 = \langle \bar{\mathbb{x}}, \bar{\mathbb{x}} \rangle = (\bar{x}, \bar{y})\begin{pmatrix}\lambda_1&0\\0&\lambda_2\end{pmatrix}\begin{pmatrix}\bar{x}\\\bar{y}\end{pmatrix}$$
$$= \bar{\mathbb{x}}^T \Lambda \bar{\mathbb{x}}$$

$\Rightarrow$ points $\bar{\mathbb{x}}$ on unit contour satisfy

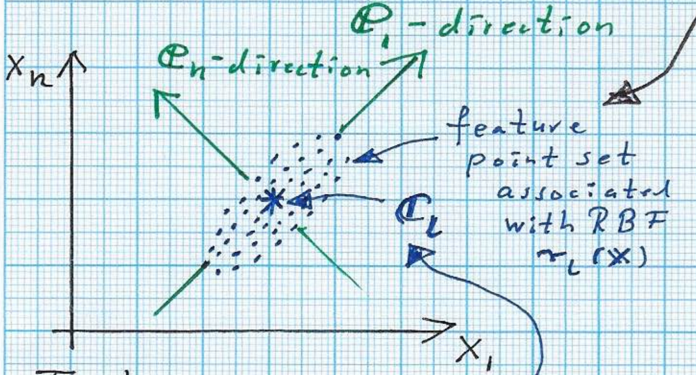
$$\|\bar{\mathbb{x}}\|^2 = 1 \Leftrightarrow \lambda_1 \bar{x}^2 + \lambda_2 \bar{y}^2 = 1$$

"Lengths of axes given by $\sqrt{\ }$ of eigenvalues"

$$\lambda_1 \left(\tfrac{x}{4}\right)^2 + \lambda_2 \left(\tfrac{y}{2}\right)^2 = 1$$
$$\tfrac{\lambda_1}{16} x^2 + \tfrac{\lambda_2}{4} y^2 = 1$$
$$\Rightarrow \lambda_1 = 16,\ \lambda_2 = 4$$

▪ RBFs AND CLASSIFICATION – Cont'd.

… 6) DISTANCE:



Feature space

• subtract $\mathbb{c}_L$ from feature points in this set prior to PCA

Feature point set $\{\mathbb{x}_i^L\}$ associated with $r_L(\mathbb{x})$ can be subjected to a principal component analysis (PCA) / covariance matrix analysis.

$\Rightarrow$ • real positive eigenvalues $\lambda_1, \ldots, \lambda_n$

• mutually orthogonal eigenvectors $\mathbb{e}_1, \ldots, \mathbb{e}_n$

$\Rightarrow$ PCA produces an orthogonal basis defined by eigenvectors $\mathbb{e}_1, \ldots, \mathbb{e}_n$ with eigenvalues $\lambda_1, \ldots, \lambda_2$ (and local origin $\mathbb{c}_L$). A feature point $\mathbb{x} = (x_1, \ldots, x_n)^T$ can therefore be represented with respect to this "PCA system" with local coordinates $\overline{\mathbb{x}} = (\overline{x}_1, \ldots, \overline{x}_n)^T$. ONE CAN NOW DETERMINE THE DISTANCE OF $\overline{\mathbb{x}}$ from center $\mathbb{c}_L$ by computing $\|\overline{\mathbb{x}}\| = \sqrt{\langle \overline{\mathbb{x}}, \overline{\mathbb{x}} \rangle}$.

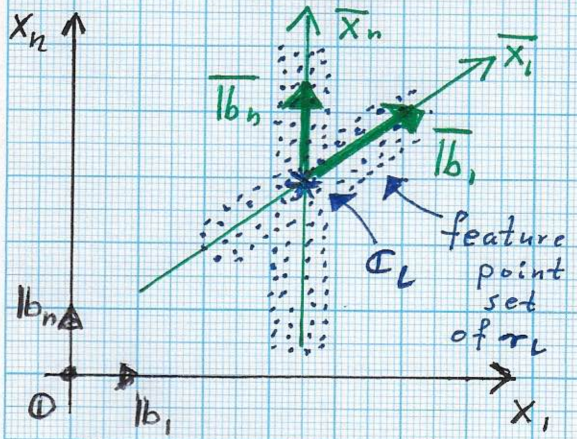
$$\Rightarrow \|\overline{\mathbb{x}}\|^2 = \overline{\mathbb{x}}^T \overline{\mathbb{x}} = (\overline{x}_1, \ldots, \overline{x}_n) \begin{pmatrix} \overline{x}_1 \\ \vdots \\ \overline{x}_n \end{pmatrix}.$$

$\Rightarrow$ More generally, one can use a "metric tensor" to define $\|\overline{\mathbb{x}}\|^2$: $\|\overline{\mathbb{x}}\|^2 = \overline{\mathbb{x}}^T Q \, \overline{\mathbb{x}}$. Q must be an n-by-n positive definite matrix (to be optimized).

~ BH

RBFs AND CLASSIFICATION – Cont'd.

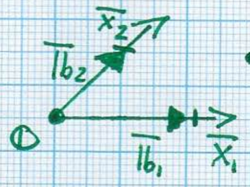
... 6) DISTANCE: • Independent component analysis (ICA) is a "generalization" of PCA and can be used to data-inherent non-orthogonal basis vectors to more properly capture distributions.



Feature space

• Computation of an "ICA basis" uses an 'expensive' optimization — but the basis allows one to represent data more adaptively.

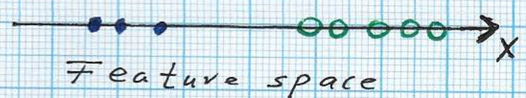
• Ex. (with $\mathbb{c}_L = \mathbb{0}$):



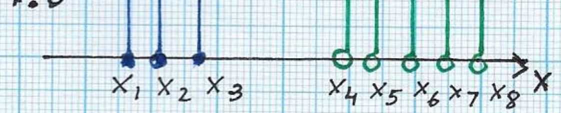
$\overline{\mathbb{x}} = \binom{1}{1}$

$\|\overline{\mathbb{x}}\|^2 = \langle \overline{\mathbb{x}}, \overline{\mathbb{x}} \rangle = 2$

• SIMPLE example: 1D feature space, 2 classes



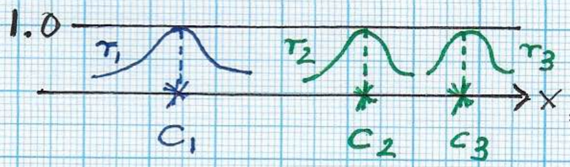
→ class 1: •, class 2: o



→ 2 classifier functions f_1, f_2, defined by conditions

$f_1(x_i) = 1,\ i = 1 \ldots 3,\quad f_2(x_i) = 1,\ i = 4 \ldots 8$

$f_1(x_i) = 0,\ i = 4 \ldots 8,\quad f_2(x_i) = 0,\ i = 1 \ldots 3$



→ using 3 RBFs centered at c_i, e.g., invers multiquadrics $(1 + d_i^2(x))^{-\frac{1}{2}}$

$$\Rightarrow f_1(x_j) = \sum_{i=1}^{3} w_i^1 \left(1 + d_i^2(x_j)\right)^{-1/2} = \begin{cases} 1, & j = 1 \ldots 3 \\ 0, & j = 4 \ldots 8 \end{cases} ;\ f_2(x_j) = \ldots$$

$\Rightarrow$ compute BEST approximations $\Rightarrow \mathbb{F} = (f_1, f_2)$ classifier

~ BH