An Interactive Approach for CBIR Using a Network of Radial Basis Functions

by P. Muneesawang and L. Guan
IEEE Transactions on Multimedia, 2004

Joseph Lee

Contents

1. Background

- Content-based Image Retrieval (CBIR)
  - use colors, shapes, textures → rely on image itself
  - without image content → rely on metadata

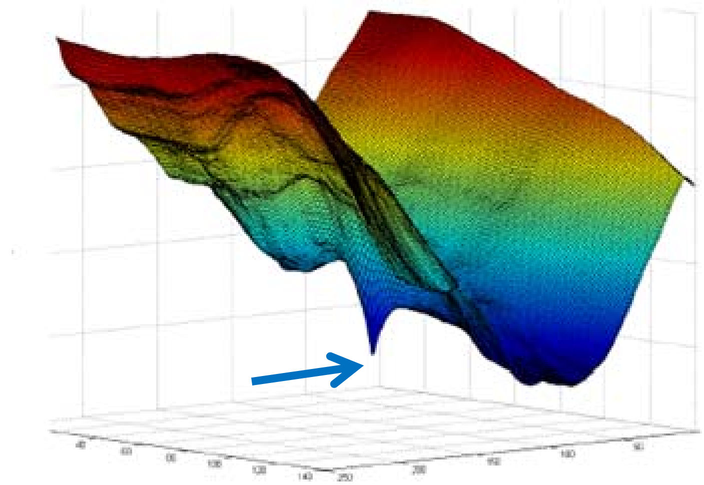
1. Background

- Content-based Image Retrieval (CBIR)
  - use colors, shapes, textures → rely on image itself
  - without image content → rely on metadata

- Relevance Feedback
  - User refines the result by marking images as *relevant* or *non-relevant* and repeats search.

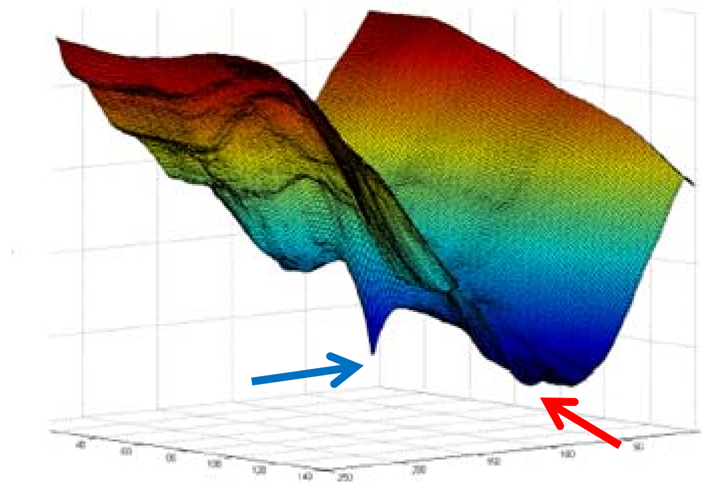
1. Background

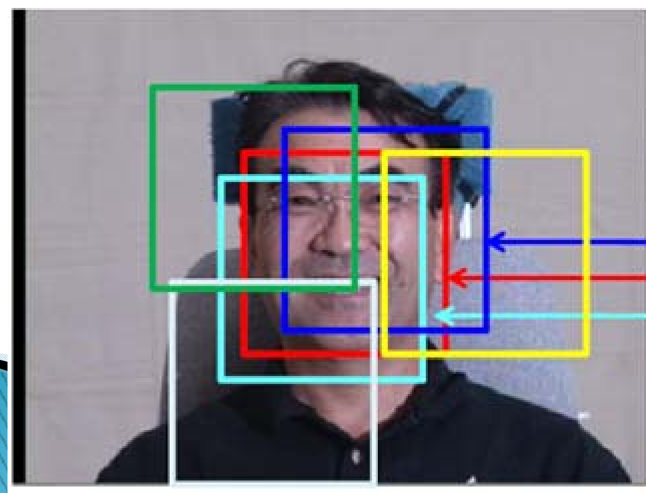
- Image Similarity

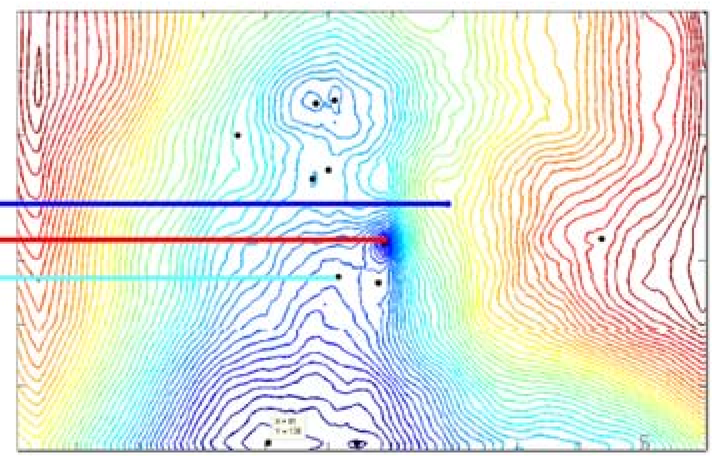


1. Background

- Image Similarity

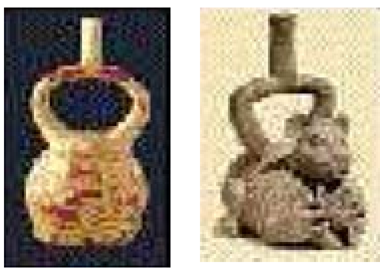




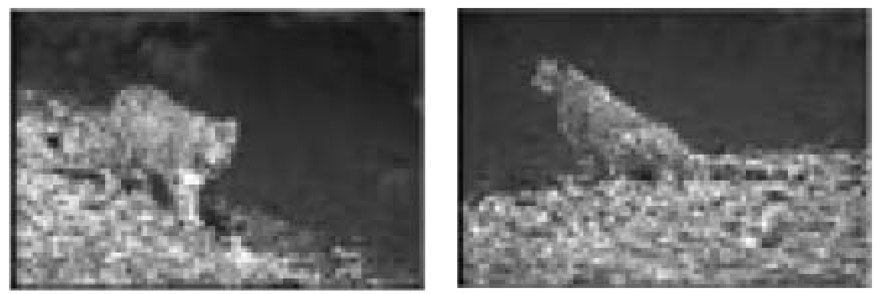


1. Background

- Examples of Image Similarity Problems



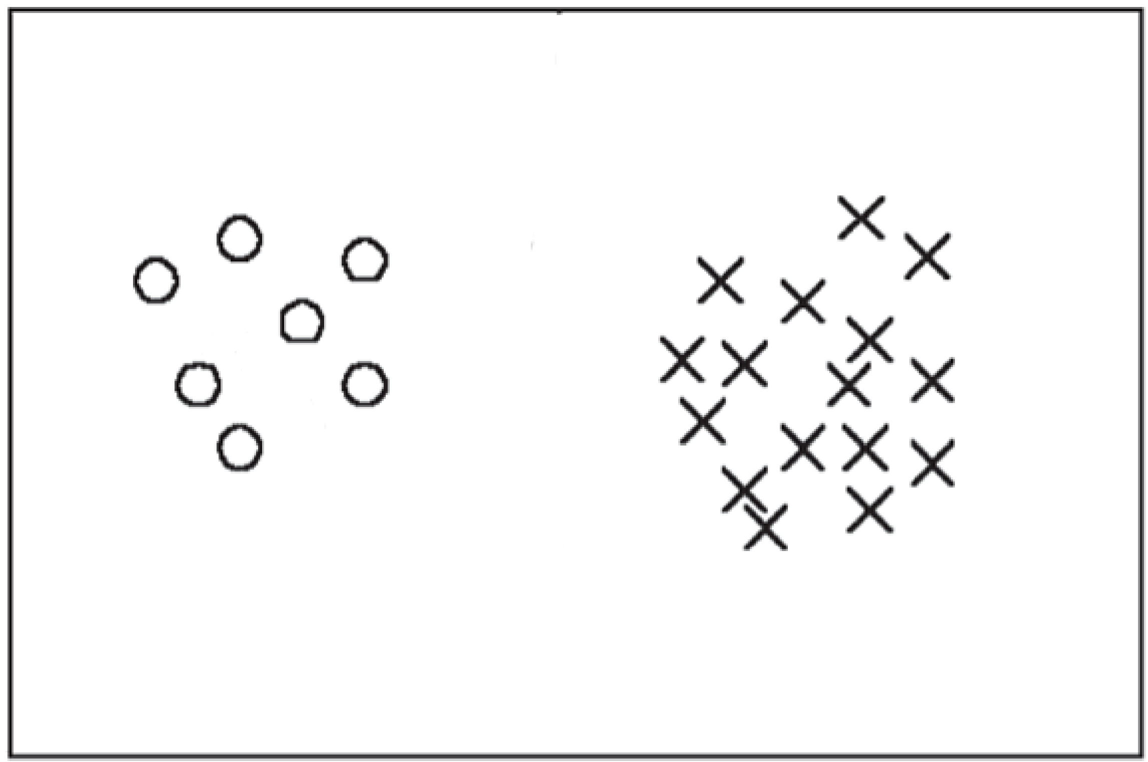
MARS-1



RBF

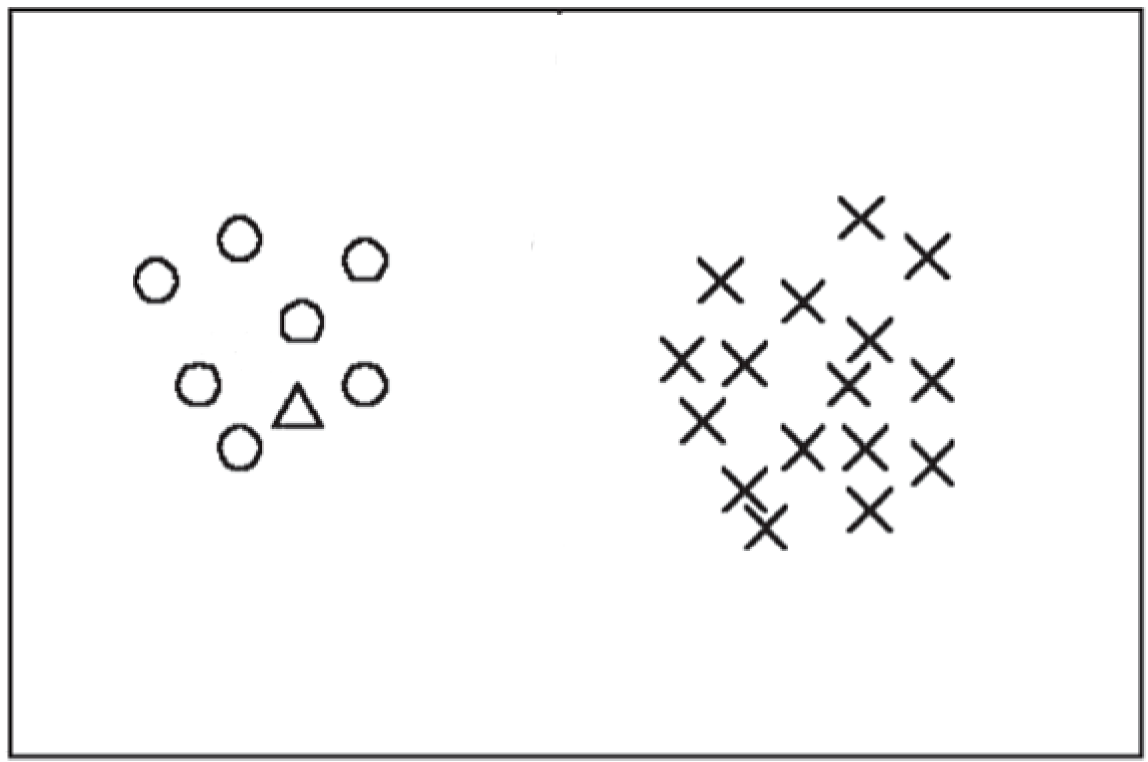
2. General Framework

- Ideal Case



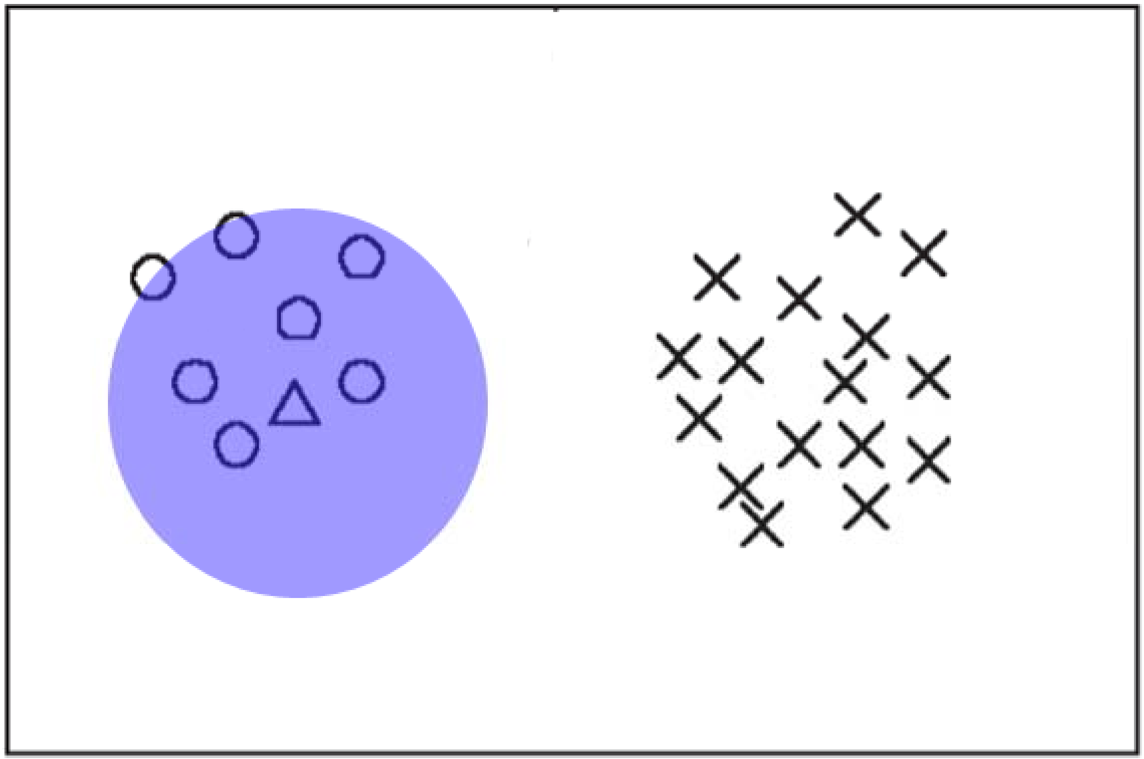
2. General Framework

- Ideal Case – User Query



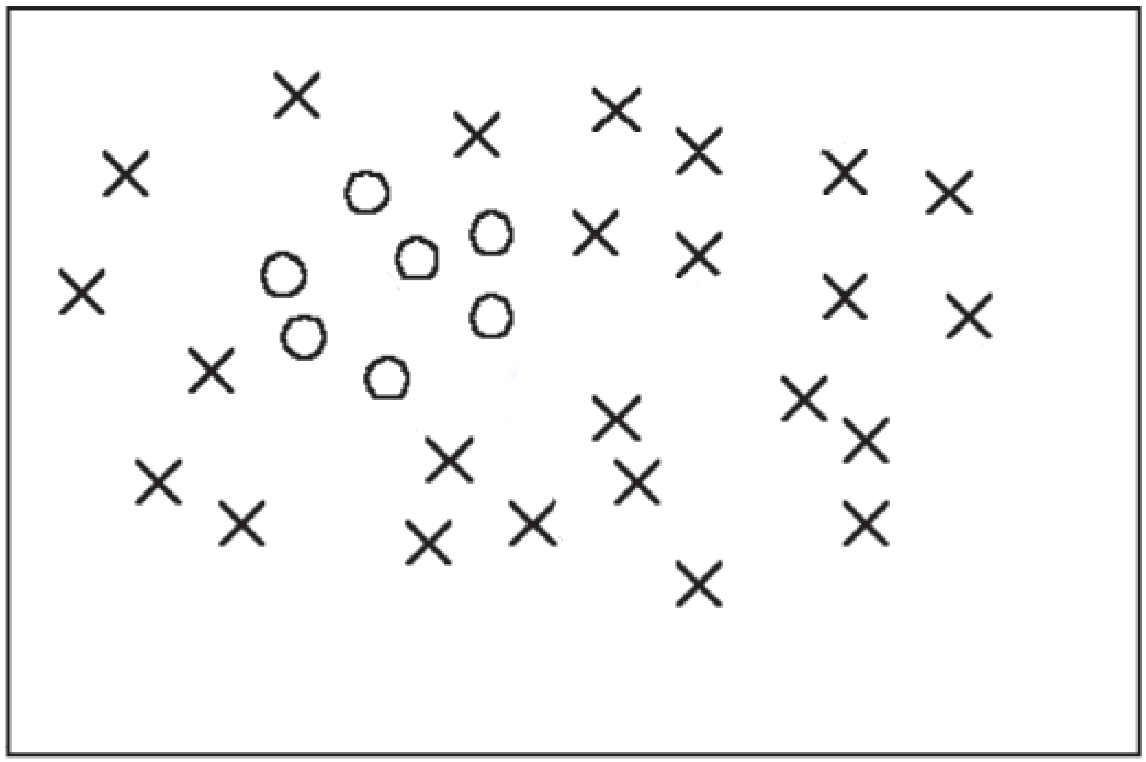
2. General Framework

- Ideal Case – Similarity Measure



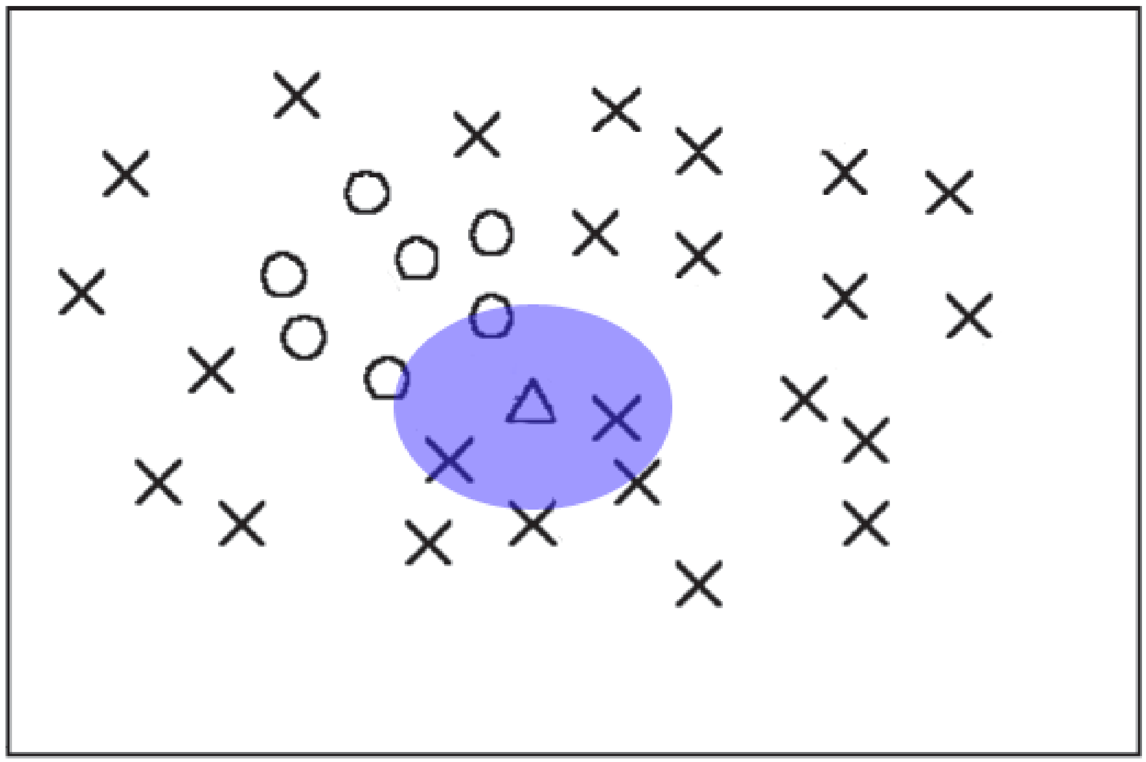
2. General Framework

- Real Case



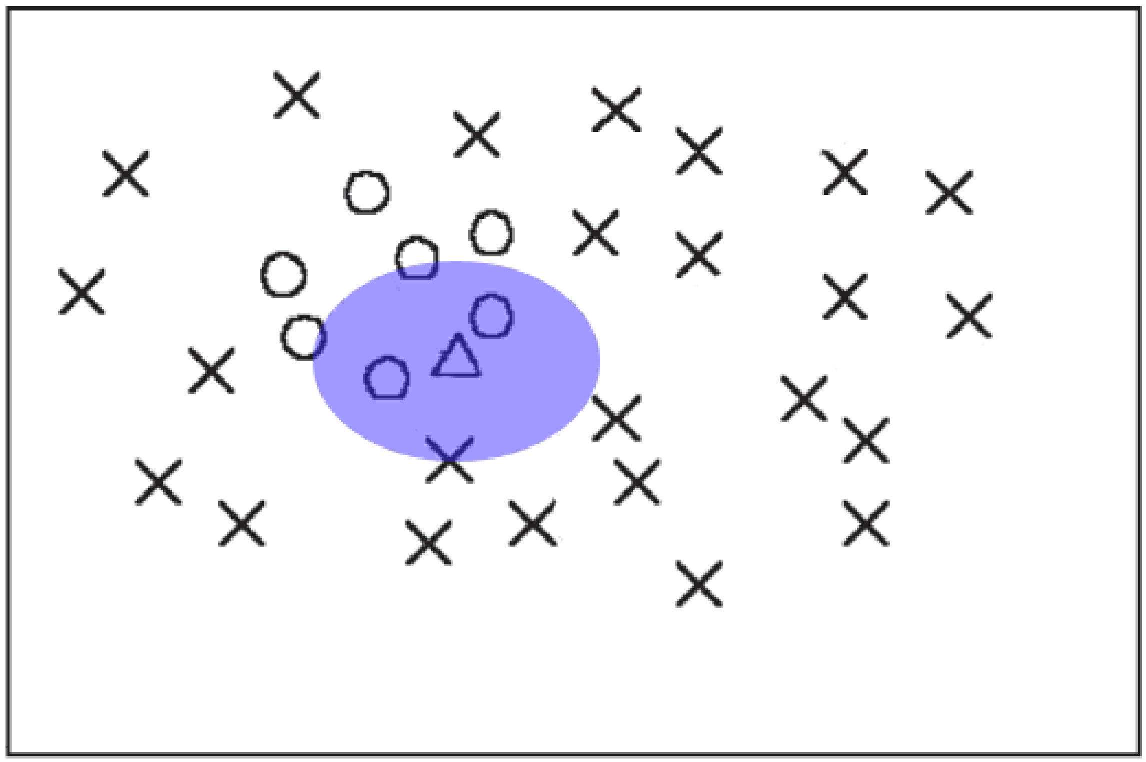
2. General Framework

- Real Case – Inaccurate Query



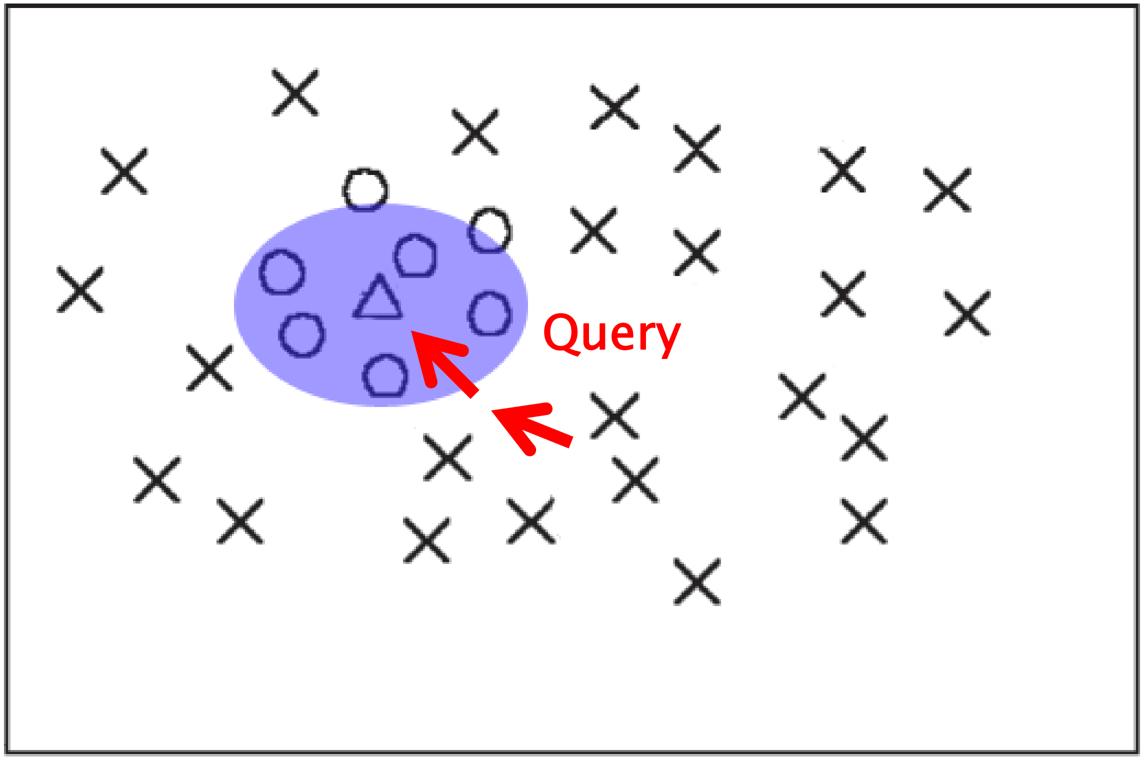
2. General Framework

- Real Case – Inaccurate Query



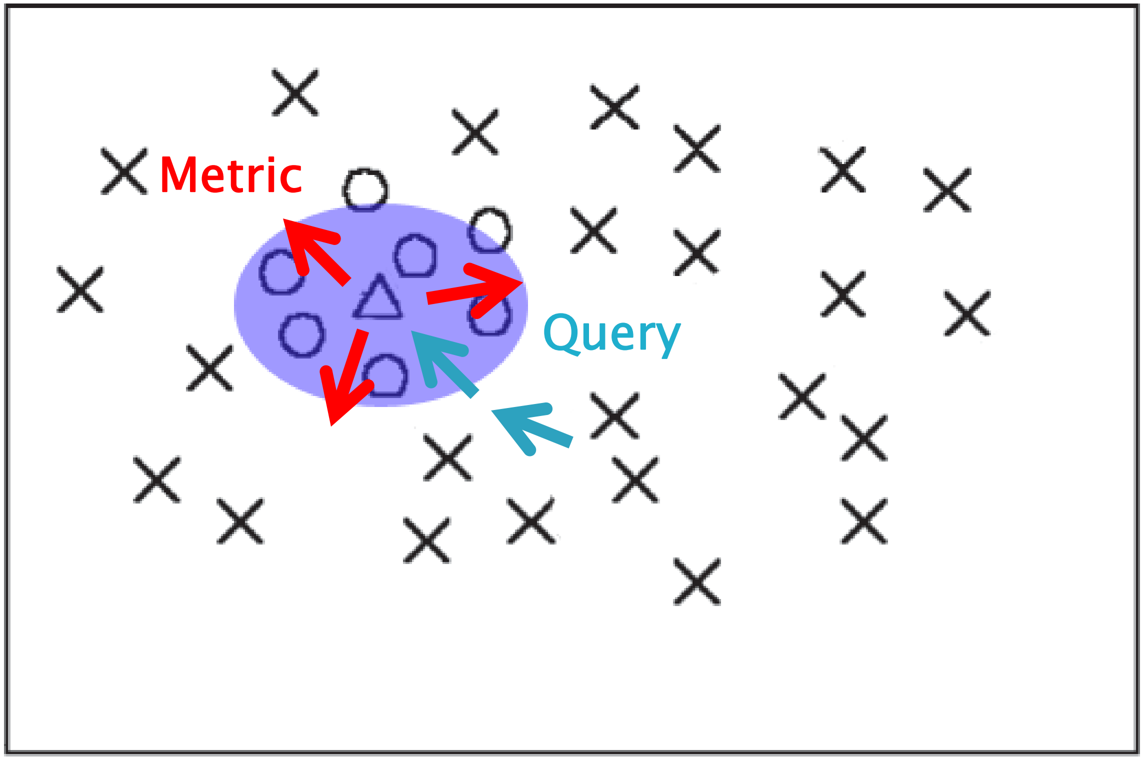
2. General Framework

- We should refine Query



2. General Framework

- We should refine Query & Metric



3. Previous Models

- Query Reformulation Model
  - Relevance feedback → learn **query representation**
- Adaptive Metric Model
  - Relevance feedback → learn **similarity function**

3. Previous Models

- Learning user perception

$$y_s = f(\mathbf{x})$$

3. Previous Models

- Learning user perception

$$y_s = f(\mathbf{x})$$

  - MARS-1 $\quad y_s = f_{\text{cosine}}(\mathbf{x}, \mathbf{x}_{\hat{q}})$ $\quad \boxed{\mathbf{x}_{\hat{q}} = \alpha \mathbf{x}_q + \gamma(\underset{l_i=1}{\text{mean}}\{\mathbf{x}_i\}) - \varepsilon(\underset{l_i=0}{\text{mean}}\{\mathbf{x}_i\})}$

3. Previous Models

- Learning user perception

$$y_s = f(\mathbf{x})$$

  - MARS-1 $\quad y_s = f_{\text{cosine}}(\mathbf{x}, \mathbf{x}_{\hat{q}}) \quad \boxed{\mathbf{x}_{\hat{q}} = \alpha \mathbf{x}_q + \gamma(\underset{l_i=1}{\text{mean}}\{\mathbf{x}_i\}) - \varepsilon(\underset{l_i=0}{\text{mean}}\{\mathbf{x}_i\})}$
  - MARS-2 $\quad y_s = f(\mathbf{x}, \mathbf{x}_q) \boxed{= (\mathbf{x} - \mathbf{x}_q)^T W (\mathbf{x} - \mathbf{x}_q)}$

3. Previous Models

- Learning user perception

$$y_s = f(\mathbf{x})$$

  - MARS-1 $\quad y_s = f_{\text{cosine}}(\mathbf{x}, \mathbf{x}_{\hat{q}}) \quad \boxed{\mathbf{x}_{\hat{q}} = \alpha \mathbf{x}_q + \gamma(\underset{l_i=1}{\text{mean}}\{\mathbf{x}_i\}) - \varepsilon(\underset{l_i=0}{\text{mean}}\{\mathbf{x}_i\})}$
  - MARS-2 $\quad y_s = f(\mathbf{x}, \mathbf{x}_q) \boxed{= (\mathbf{x} - \mathbf{x}_q)^T W (\mathbf{x} - \mathbf{x}_q)}$
  - OPT-RF $\quad y_s = f(\mathbf{x}, \mathbf{x}_q) \boxed{= (\mathbf{x} - \mathbf{x}_q)^T W (\mathbf{x} - \mathbf{x}_q)} \quad \boxed{\mathbf{x}_{\hat{q}} = \frac{\mathbf{X}^T \mathbf{v}}{\sum_{i=1}^{N} v_i}}$

3. Previous Models

- Learning user perception

$$y_s = f(\mathbf{x})$$

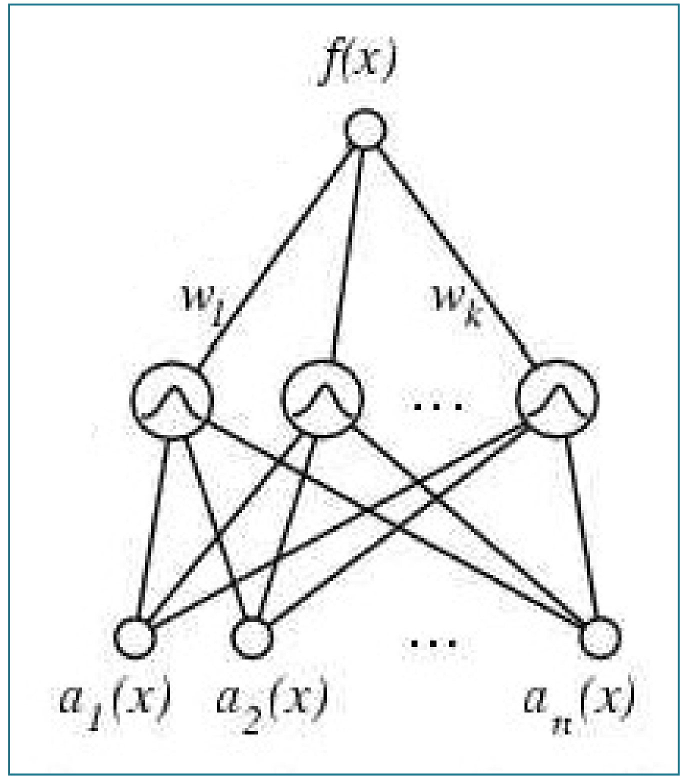
  - MARS-1 $y_s = f_{\text{cosine}}(\mathbf{x}, \mathbf{x}_{\hat{q}})$ $\boxed{\mathbf{x}_{\hat{q}} = \alpha \mathbf{x}_q + \gamma(\underset{l_i=1}{\text{mean}}\{\mathbf{x}_i\}) - \varepsilon(\underset{l_i=0}{\text{mean}}\{\mathbf{x}_i\})}$
  - MARS-2 $y_s = f(\mathbf{x}, \mathbf{x}_q) \boxed{= (\mathbf{x} - \mathbf{x}_q)^T W (\mathbf{x} - \mathbf{x}_q)}$
  - OPT-RF $y_s = f(\mathbf{x}, \mathbf{x}_q) \boxed{= (\mathbf{x} - \mathbf{x}_q)^T W (\mathbf{x} - \mathbf{x}_q)}$ $\boxed{\mathbf{x}_{\hat{q}} = \frac{\mathbf{X}^T \mathbf{v}}{\sum_{i=1}^{N} v_i}}$

- Assumption
  - same distance gives same degree of similarity

4. Proposed RBF Model

- CBIR
  - online learning
  - two-class problem

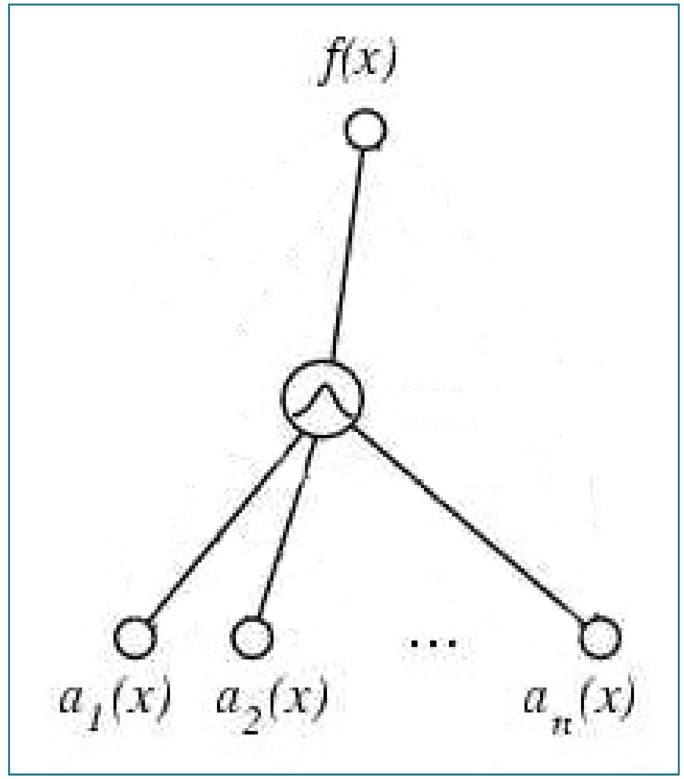
$$
\begin{aligned}
f(\mathbf{x}) &= \sum_{j=1}^{N} w_j G(\mathbf{x}, \mathbf{z}_j) \\
&= \sum_{j=1}^{N} w_j \exp\left(-\frac{1}{2\sigma_j^2}\sum_{i=1}^{P}(x_i - z_{ji})^2\right)
\end{aligned}
$$



4. Proposed RBF Model

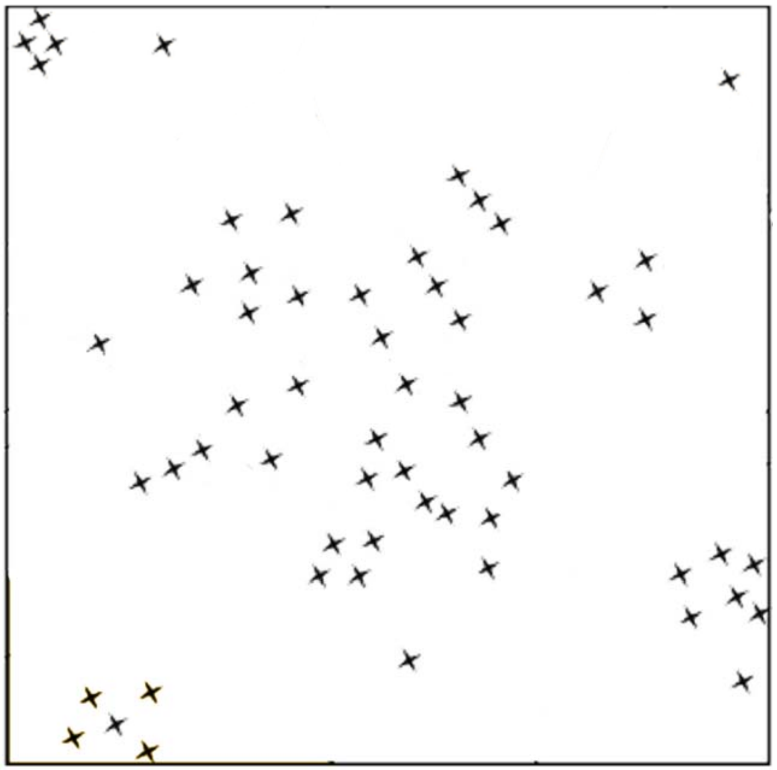
- CBIR
  - online learning
  - two-class problem

$$f(\mathbf{x}) = \sum_{i=1}^{P} G_i(x_i, z_i)$$
$$= \sum_{i=1}^{P} \exp\left(-\frac{(x_i - z_i)^2}{2\sigma_i^2}\right)$$



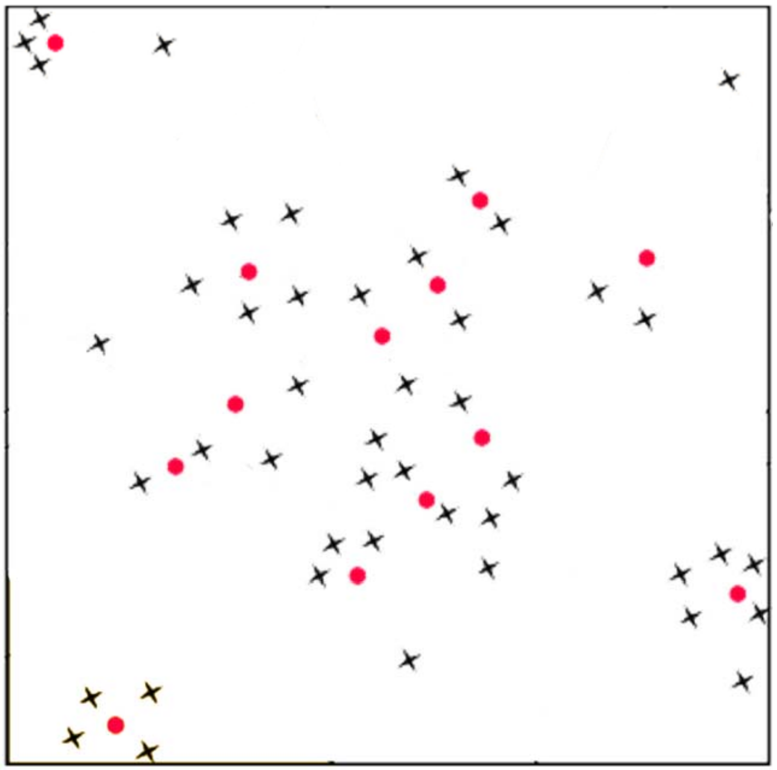
5. Learning Strategy

- Example Vectors



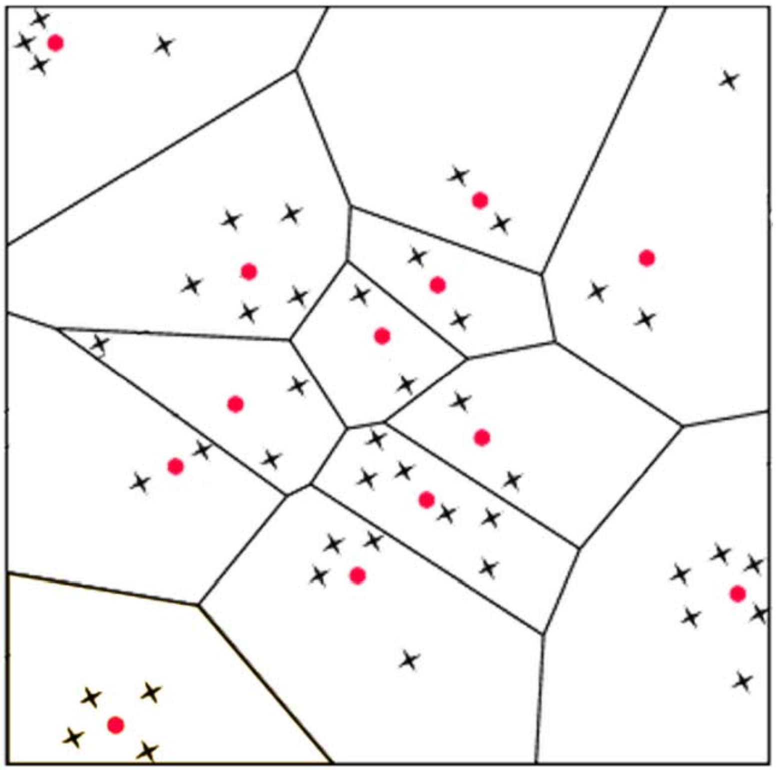
5. Learning Strategy

- Voronoi Vectors



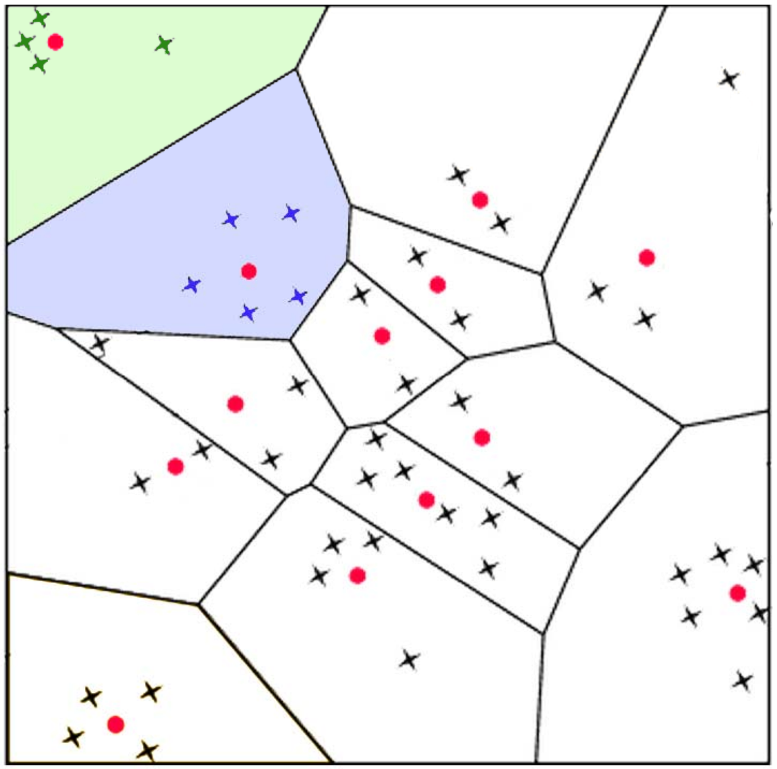
5. Learning Strategy

- Voronoi Cells



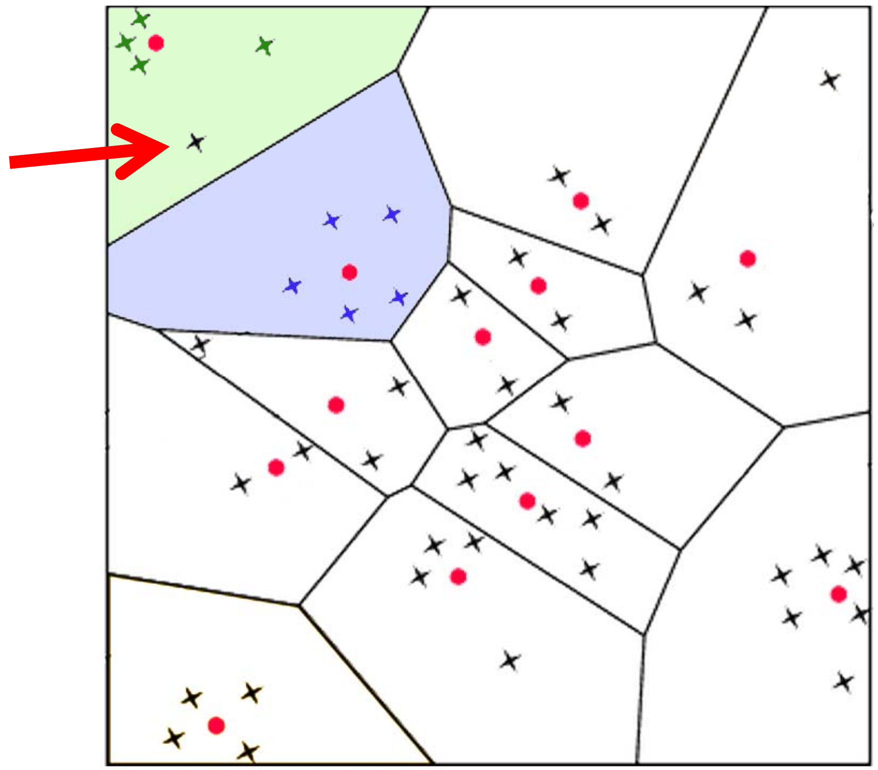
5. Learning Strategy

- Learning Vector Quantization



5. Learning Strategy

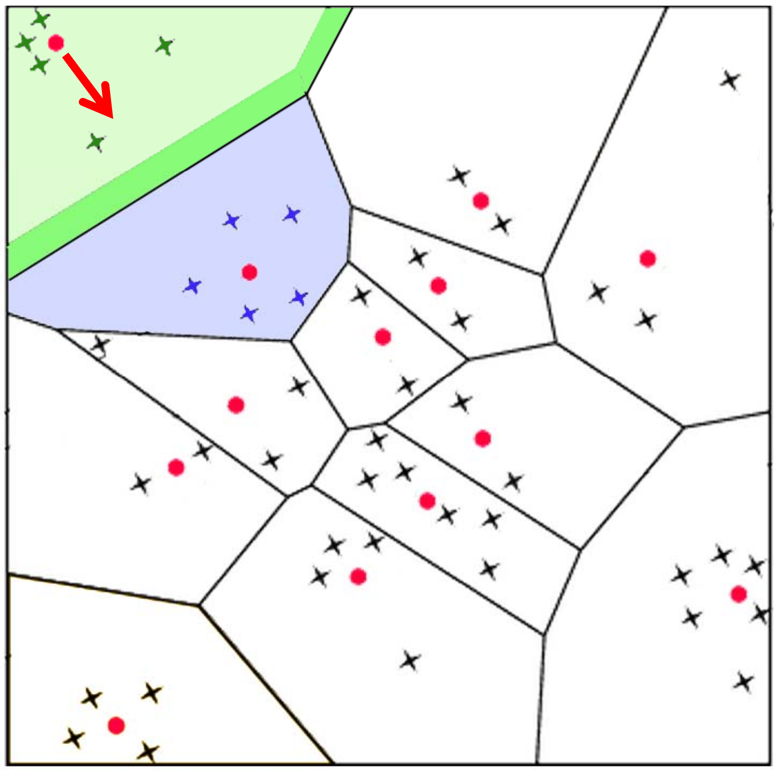
- Learning Vector Quantization



5. Learning Strategy

- Learning Vector Quantization

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

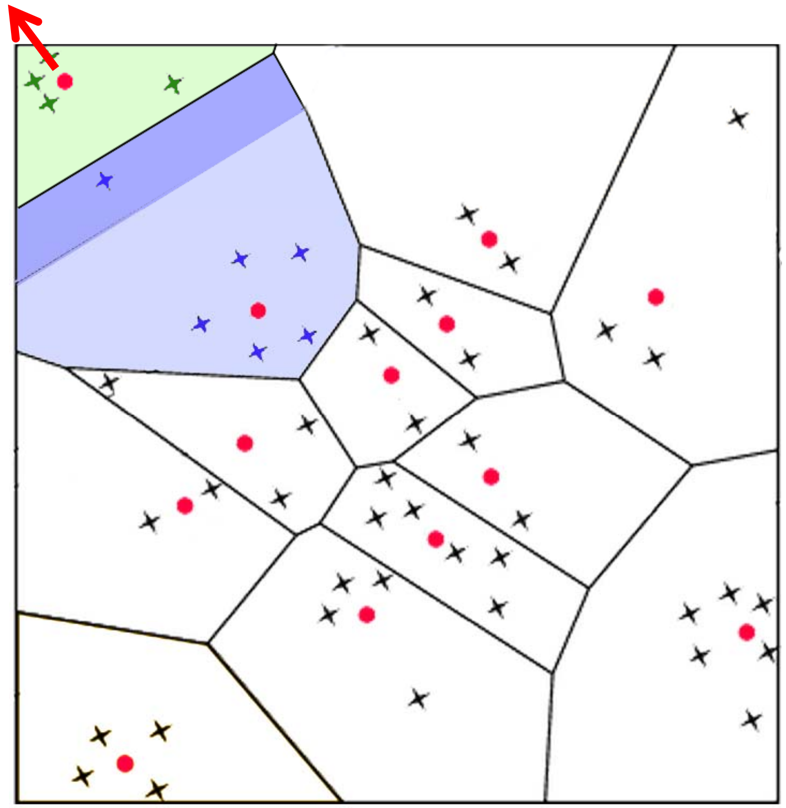


5. Learning Strategy

- Learning Vector Quantization

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) - \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

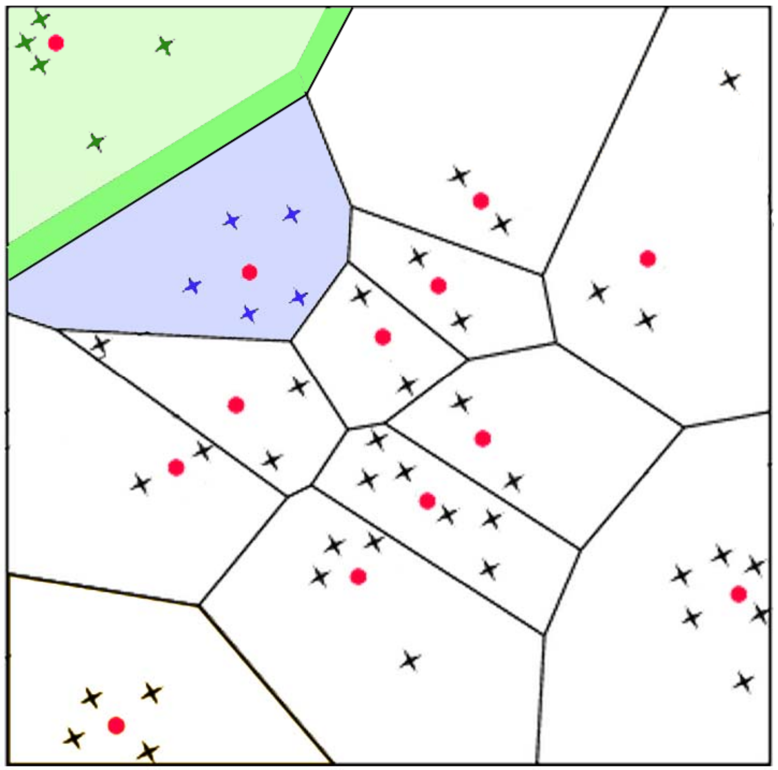


5. Learning Strategy

- Modified LVQ (Model 1)

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) - \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$



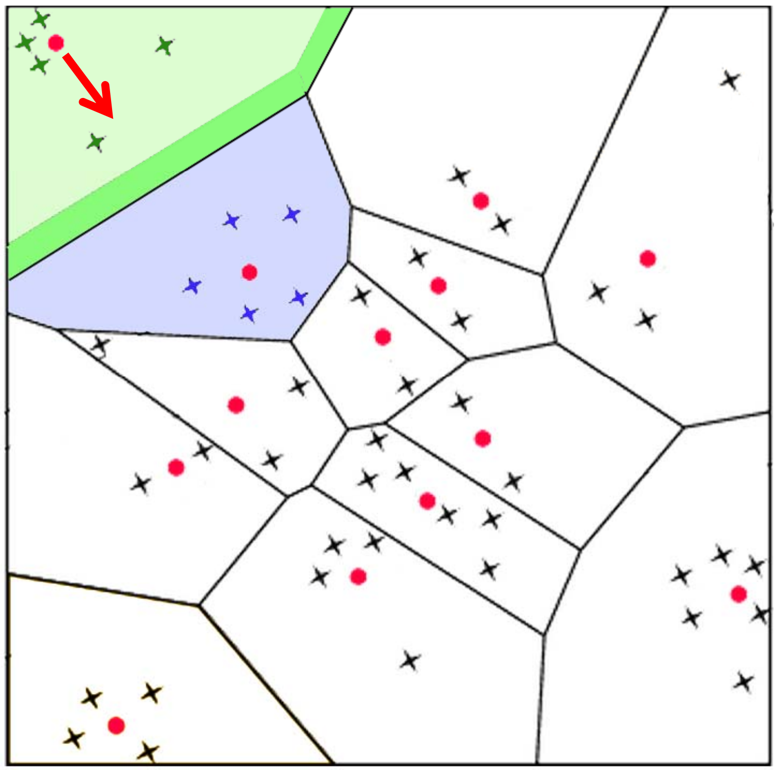
5. Learning Strategy

- Modified LVQ (Model 1)

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) - \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_q(t+1) = \mathbf{z}_q(t) + \alpha_R(\bar{\mathbf{x}}' - \mathbf{z}_q(t)) - \underline{\alpha_N(\bar{\mathbf{x}}'' - \mathbf{z}_q(t))}$$



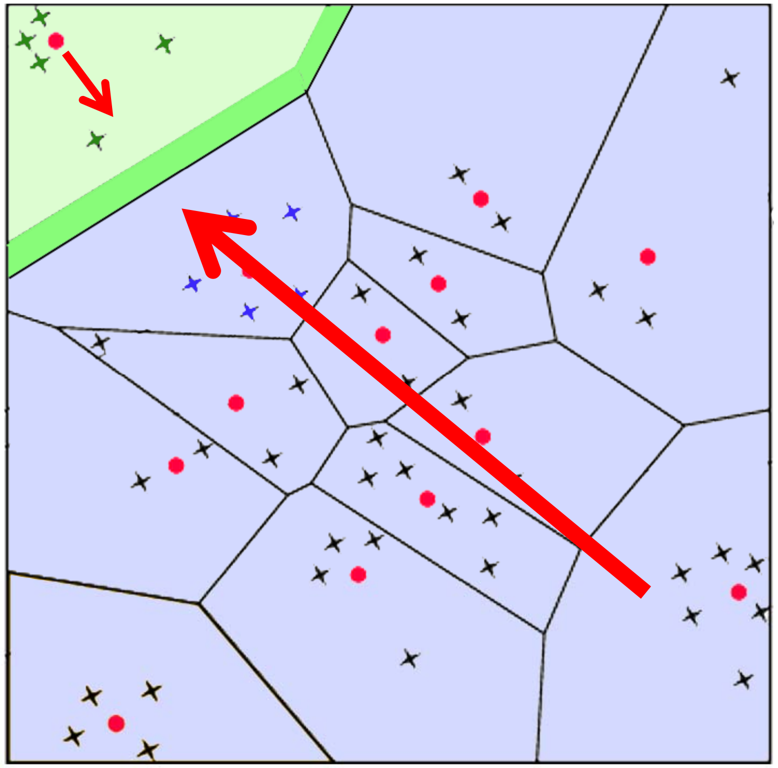
5. Learning Strategy

- Modified LVQ (Model 1)

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) - \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_q(t+1) = \mathbf{z}_q(t) + \alpha_R(\bar{\mathbf{x}}' - \mathbf{z}_q(t)) - \underline{\alpha_N(\bar{\mathbf{x}}'' - \mathbf{z}_q(t))}$$



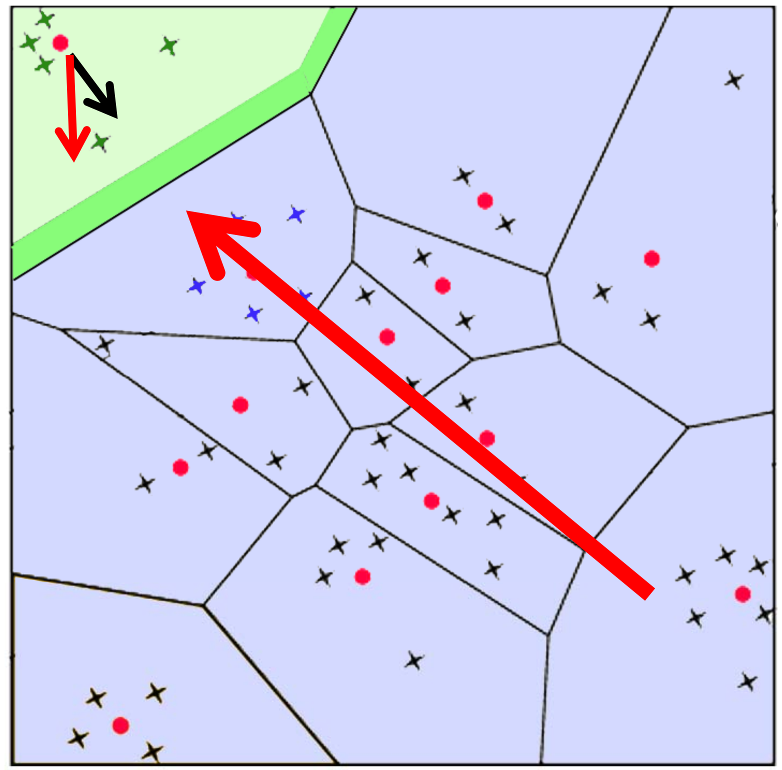
5. Learning Strategy

- Modified LVQ (Model 1)

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) - \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_q(t+1) = \mathbf{z}_q(t) + \alpha_R(\bar{\mathbf{x}}' - \mathbf{z}_q(t)) - \underline{\alpha_N(\bar{\mathbf{x}}'' - \mathbf{z}_q(t))}$$



5. Learning Strategy

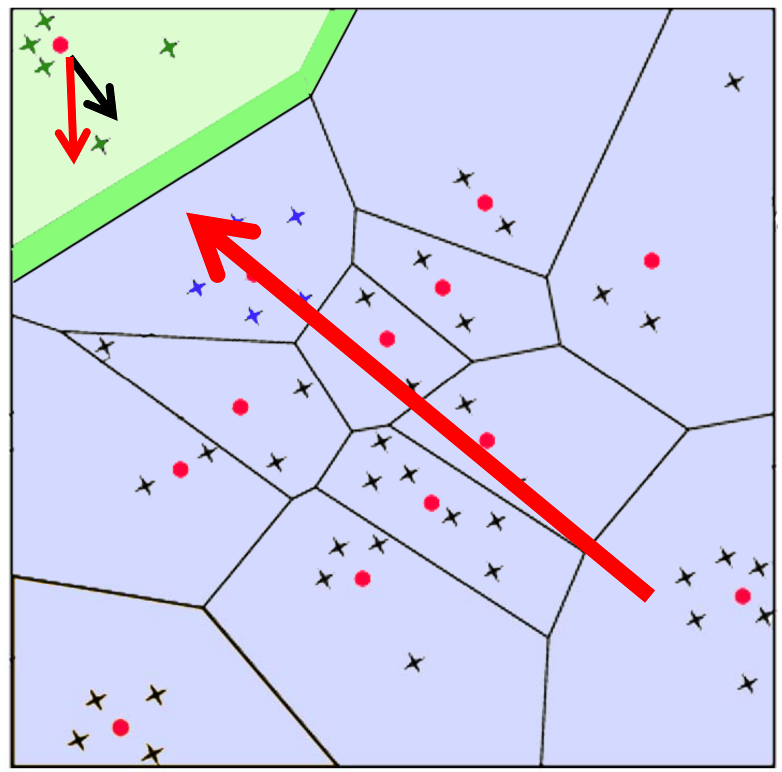
- Modified LVQ (Model 2)

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) - \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]$$

$$\mathbf{z}_q(t+1) = \mathbf{z}_q(t) + \alpha_R(\bar{\mathbf{x}}' - \mathbf{z}_q(t)) - \underline{\alpha_N(\bar{\mathbf{x}}'' - \mathbf{z}_q(t))}$$

$$\mathbf{z}_q(t+1) = \bar{\mathbf{x}}' - \underline{\alpha_N(\bar{\mathbf{x}}'' - \mathbf{z}_q(t))}$$

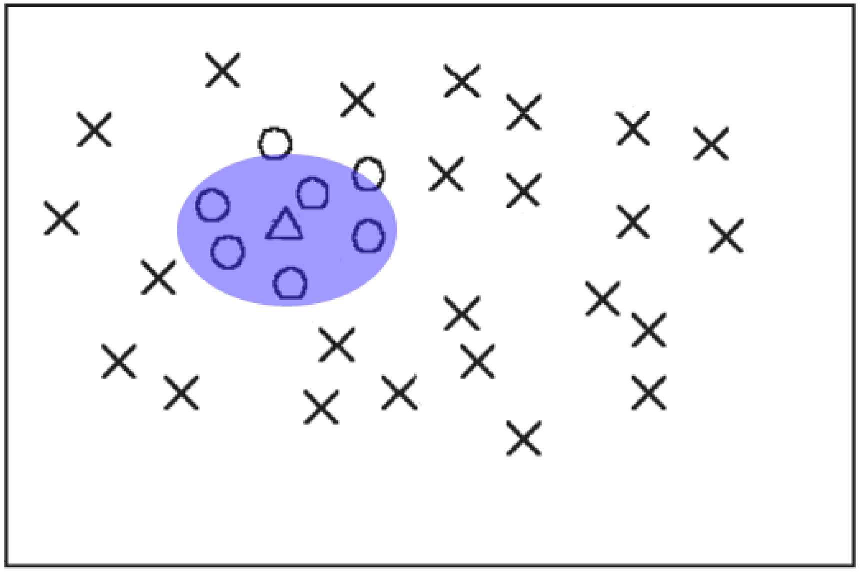


5. Learning Strategy

- Effects of Positive and Negative Learning
  - Relevant samples → common interest
  - Non-relevant samples → specific interest

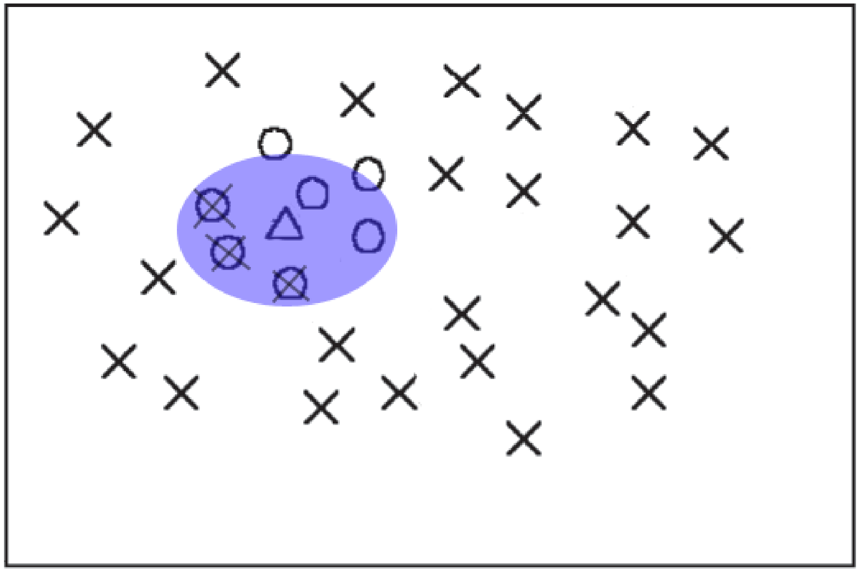
5. Learning Strategy

- Effects of Positive and Negative Learning
  - Relevant samples → common interest
  - Non-relevant samples → specific interest



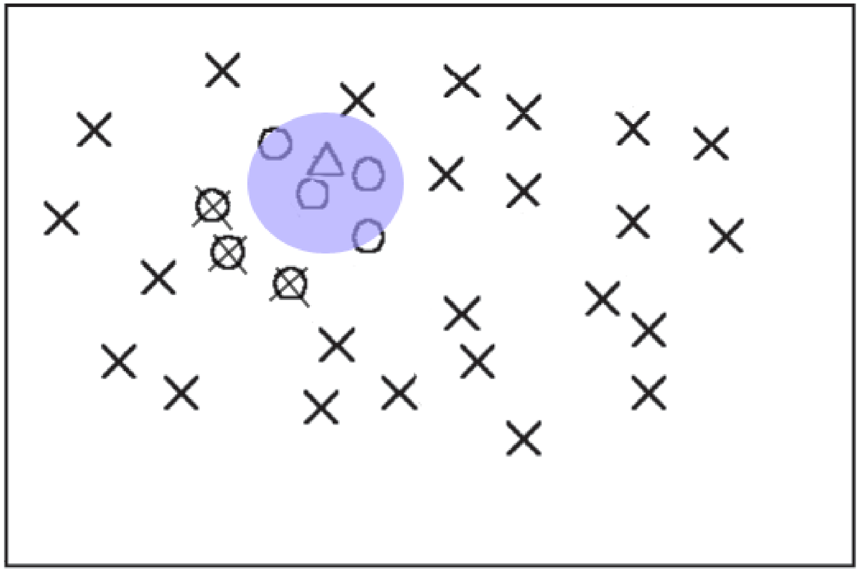
5. Learning Strategy

- **Effects of Positive and Negative Learning**
  - Relevant samples → common interest
  - Non-relevant samples → specific interest



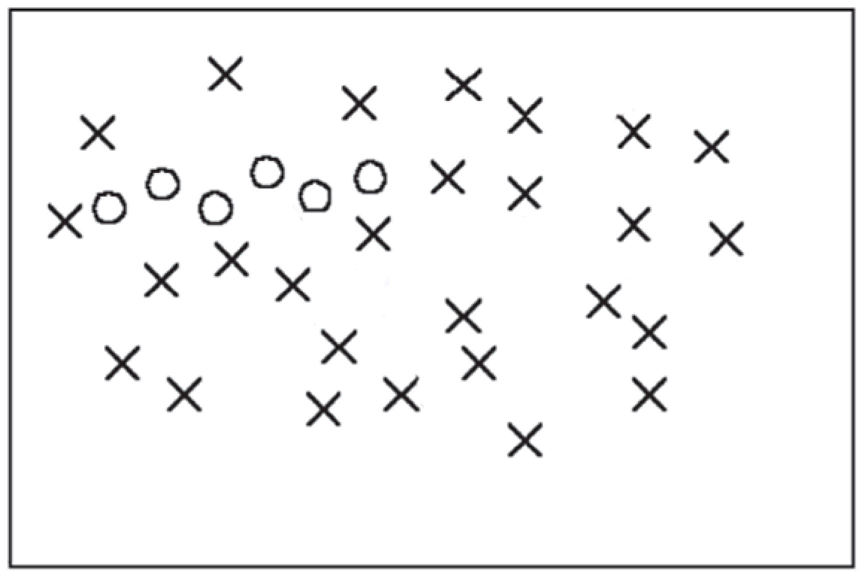
5. Learning Strategy

- Effects of Positive and Negative Learning
  - Relevant samples → common interest
  - Non-relevant samples → specific interest



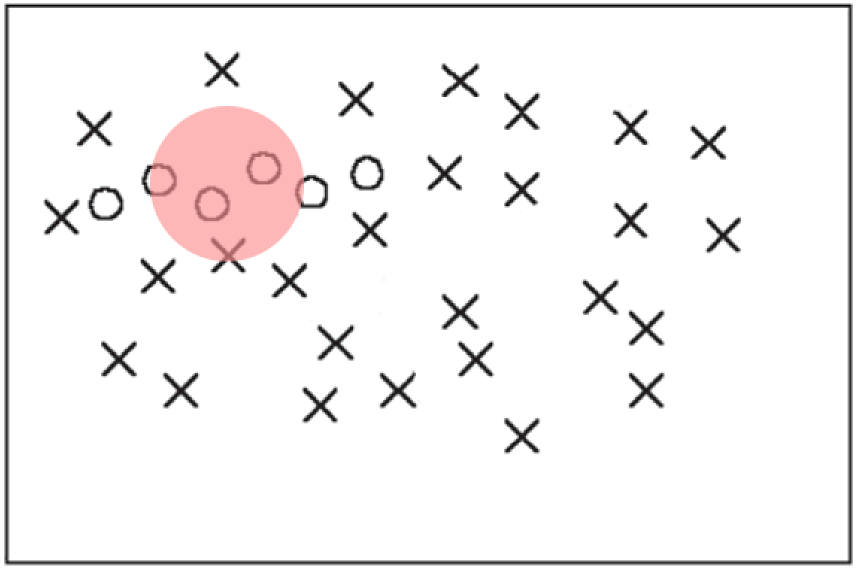
5. Learning Strategy

- Selection of RBF Width



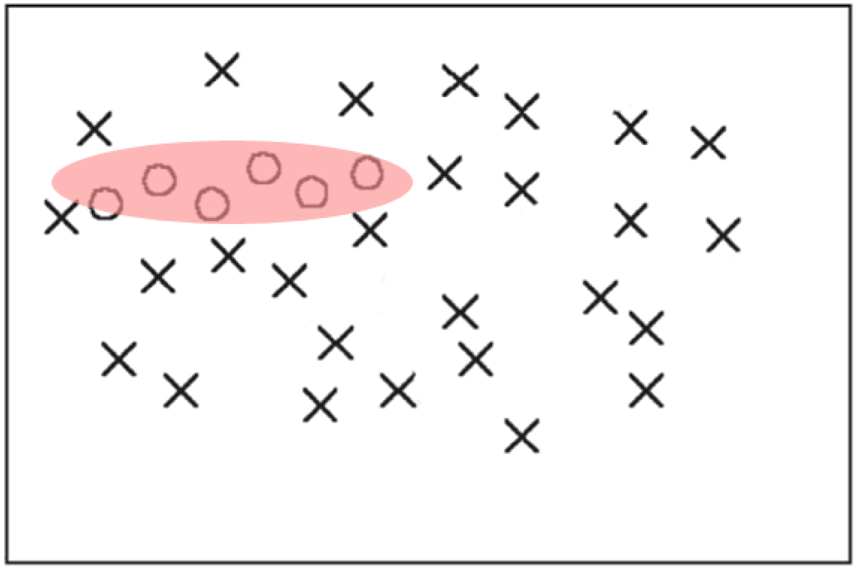
5. Learning Strategy

- Selection of RBF Width



5. Learning Strategy

- Selection of RBF Width
  - $\sigma_i = \eta \max_m |x'_{mi} - z_i|$
  - $\sigma_i = \exp(\beta \cdot \mathrm{Std}_i)$



6. Experimental Results

Method	t=0	t=1	t=2	t=3	CPU time (second per iteration)
RBF	44.82	$\boxed{79.82}$	88.75	91.79	2.34
MARS-2	44.82	60.18	61.61	61.96	1.26
OPT-RF	44.82	72.14	$\boxed{79.64}$	80.54	1.27
Simple CBIR	44.82	-	-	-	0.90

Average Precision Rate (%)

6. Conclusion

- Non-linear model for similarity evaluation
- Learning from positive and negative samples

References

- http://videolectures.net/minh_hoai_nguyen/
- http://www.mqasem.net/vectorquantization/vq.html