Learning Similarity Functions from Qualitative Feedback

Weiwei Cheng and Eyke Hüllermeier
University of Marburg, Germany

Introduction

- Proper definition of similarity (distance) measures is crucial for CBR systems.

- The specification of local similarity measures, pertaining to individual properties (attributes) of a case, is often less difficult than their combination into a global measure.

Goal of this work:

- Using machine learning techniques to support elicitation of similarity measures (combination of local into global measures) on the basis of qualitative feedback.

Problem Setting

Local-global principle: The *global distance* is an aggregation of *local distances* $\delta_i : \mathbb{C} \times \mathbb{C} \to \mathbb{R}_+$:

$$\Delta(\mathbf{a}, \mathbf{b}) = \mathrm{AGG}\left(\delta_1(\mathbf{a}, \mathbf{b}), \delta_2(\mathbf{a}, \mathbf{b}), \ldots, \delta_d(\mathbf{a}, \mathbf{b})\right)$$

For now, we focus on a linear model:

$$\Delta(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{d} w_i \, \delta_i(\mathbf{a}, \mathbf{b})$$

with $w_i \geq 0$ (monotonicity).

- … easy to incorporate background knowledge
- … amenable to efficient learning scheme
- … non-linear extension via *kernelization*

Problem Setting cont.

- Learning the weights from qualitative feedback:
  $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in \mathbb{C}^3$ means "case **a** is more similar to **b** than to **c**".

- Given a query, a distance measure induces a linear order on cases:

$$\mathbf{q} = (q_1 \ldots q_d) \in \mathbb{C}$$

$$\mathbf{a} \succeq_{\mathbf{q},\Delta} \mathbf{b} \overset{\text{df}}{\Longleftrightarrow} \Delta(\mathbf{q}, \mathbf{a}) \leq \Delta(\mathbf{q}, \mathbf{b})$$

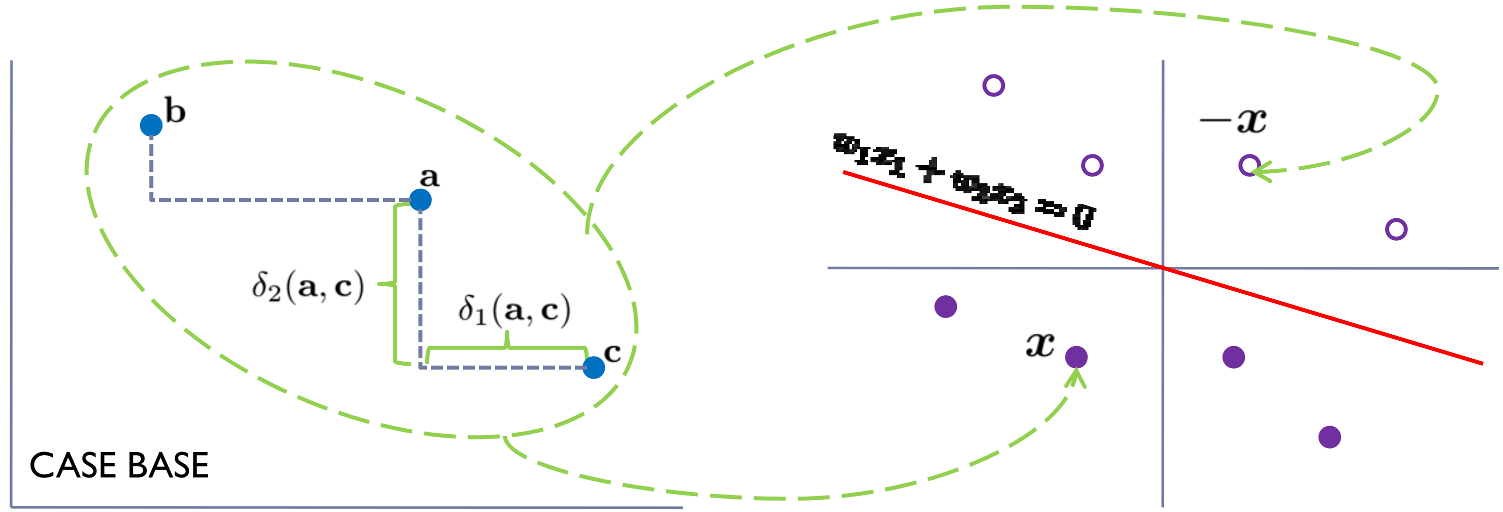
Notice: Often the *ordering of cases* is more important than the distance itself $\longrightarrow$ it is sufficient to find a Δ^{est}, such that

$$\succeq_{\mathbf{q},\Delta^{est}} \approx \succeq_{\mathbf{q},\Delta}$$

The Learning Algorithm

- Basic idea: From distance learning to classification
- Extension 1: Incorporating monotonicity
- Extension 2: Ensemble learning
- Extension 3: Active learning

From Distance Learning to Classification



$$
\begin{aligned}
\text{Triplet } (\mathbf{a},\mathbf{b},\mathbf{c}) \quad &\Leftrightarrow \quad \Delta(\mathbf{a},\mathbf{b}) < \Delta(\mathbf{a},\mathbf{c}) \\
&\Leftrightarrow \quad \Delta(\mathbf{a},\mathbf{c}) - \Delta(\mathbf{a},\mathbf{b}) > 0 \\
&\Leftrightarrow \quad \langle \mathbf{w}, \boldsymbol{\delta}(\mathbf{a},\mathbf{c}) \rangle - \langle \mathbf{w}, \boldsymbol{\delta}(\mathbf{a},\mathbf{b}) \rangle > 0 \\
&\Leftrightarrow \quad \langle \mathbf{w}, \underbrace{\boldsymbol{\delta}(\mathbf{a},\mathbf{c}) - \boldsymbol{\delta}(\mathbf{a},\mathbf{b})}_{x \ \text{(d-dim. vector)}} \rangle > 0
\end{aligned}
$$

Monotonicity

Our model $\Delta(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{d} w_i\, \delta_i(\mathbf{a},\mathbf{b})$

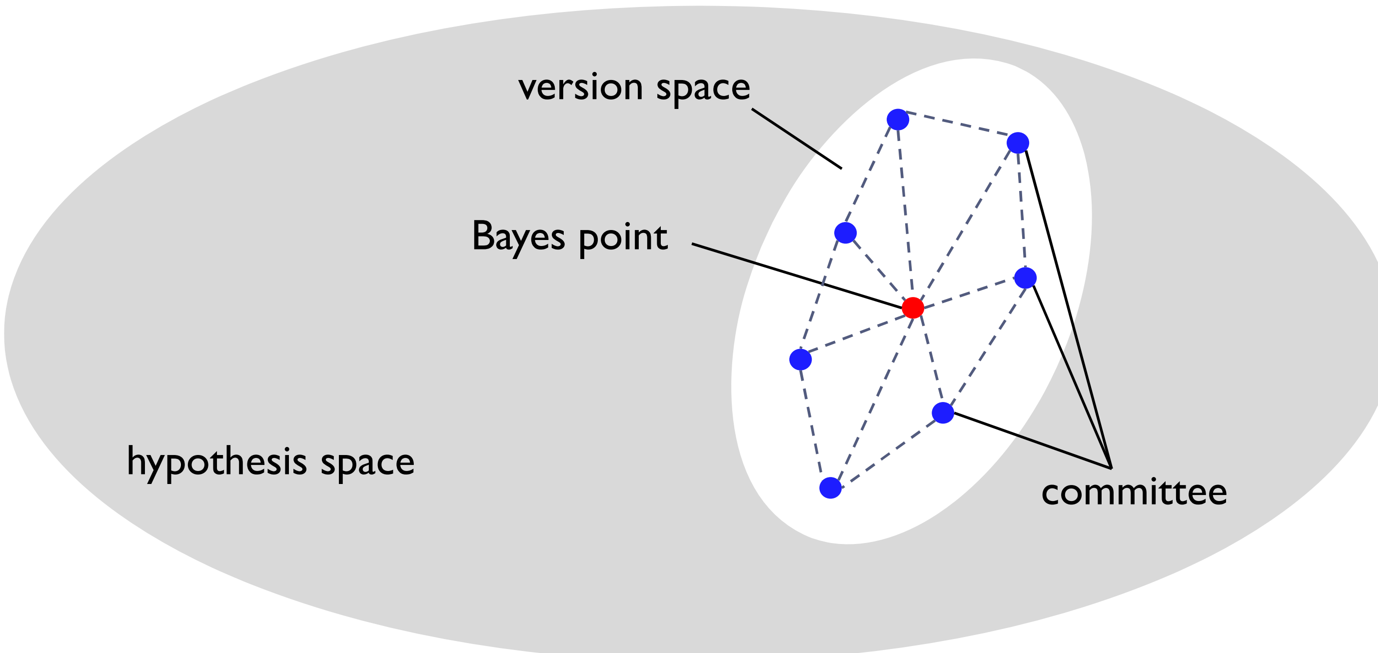
requires that $w_i \geq 0 \implies$ when a local distance increases, the global distance *cannot* decrease.

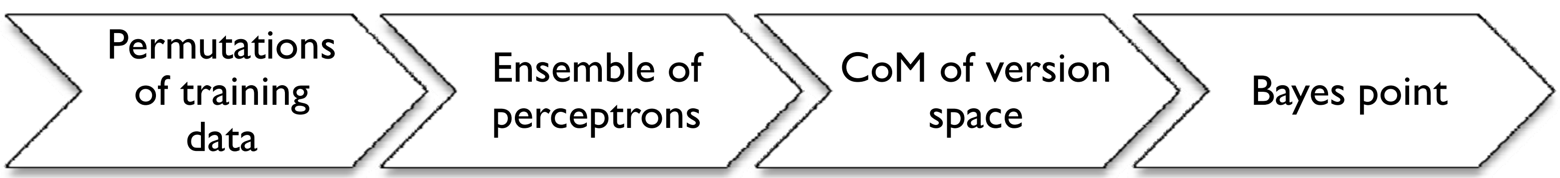
Our approach: (Noise-tolerant) Perceptron learning with a modified update rule:

$$w^{new} = \begin{cases} w^{old} + \Delta w^{old} & \text{if } w^{old} + \Delta w^{old} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The modified algorithm provably converges after a finite number of iterations.

Ensemble Learning





Active Learning

Goal:

Reducing the feedback effort of the user by choosing the most informative training data.

Our approach (a variation of QBC):

1. choose 2 most conflicting models $\arg\max_{\mathbf{w}_i, \mathbf{w}_j \in \mathbf{W}} \frac{\langle \mathbf{w}_i, \mathbf{w}_j \rangle}{\|\mathbf{w}_i\| \|\mathbf{w}_j\|}$
2. generate 2 rankings with these 2 models
3. get the first conflict pair of these rankings

Example:

$\mathbf{w}_2 \rightarrow$ ranking 1: $a\ b\ c\ d\ e$

$\mathbf{w}_2 \rightarrow$ ranking 2: $a\ b\ d\ e\ c$

$(\mathbf{q}, \mathbf{c}, \mathbf{d})$

Experimental Setting

Goal:

Investigating the efficacy of our approach and the effectiveness of the extensions:

1. incorporating monotonicity
2. ensemble learning
3. active learning

Data sets

	uni	iris	wine	yeast	nba
#features	6	4	13	24	15
#cases	200	150	178	2465	3924

Quality Measures

- Kendall's tau (a common rank correlation measure)

  … defined by number of rank inversions (normalized to [-1,+1]):

  $$\#\left\{(\mathbf{a},\mathbf{b}) \,|\, \pi(\mathbf{a}) < \pi(\mathbf{b}),\ \pi^{est}(\mathbf{a}) > \pi^{est}(\mathbf{b})\right\}$$

- Recall (a common retrieval measure)

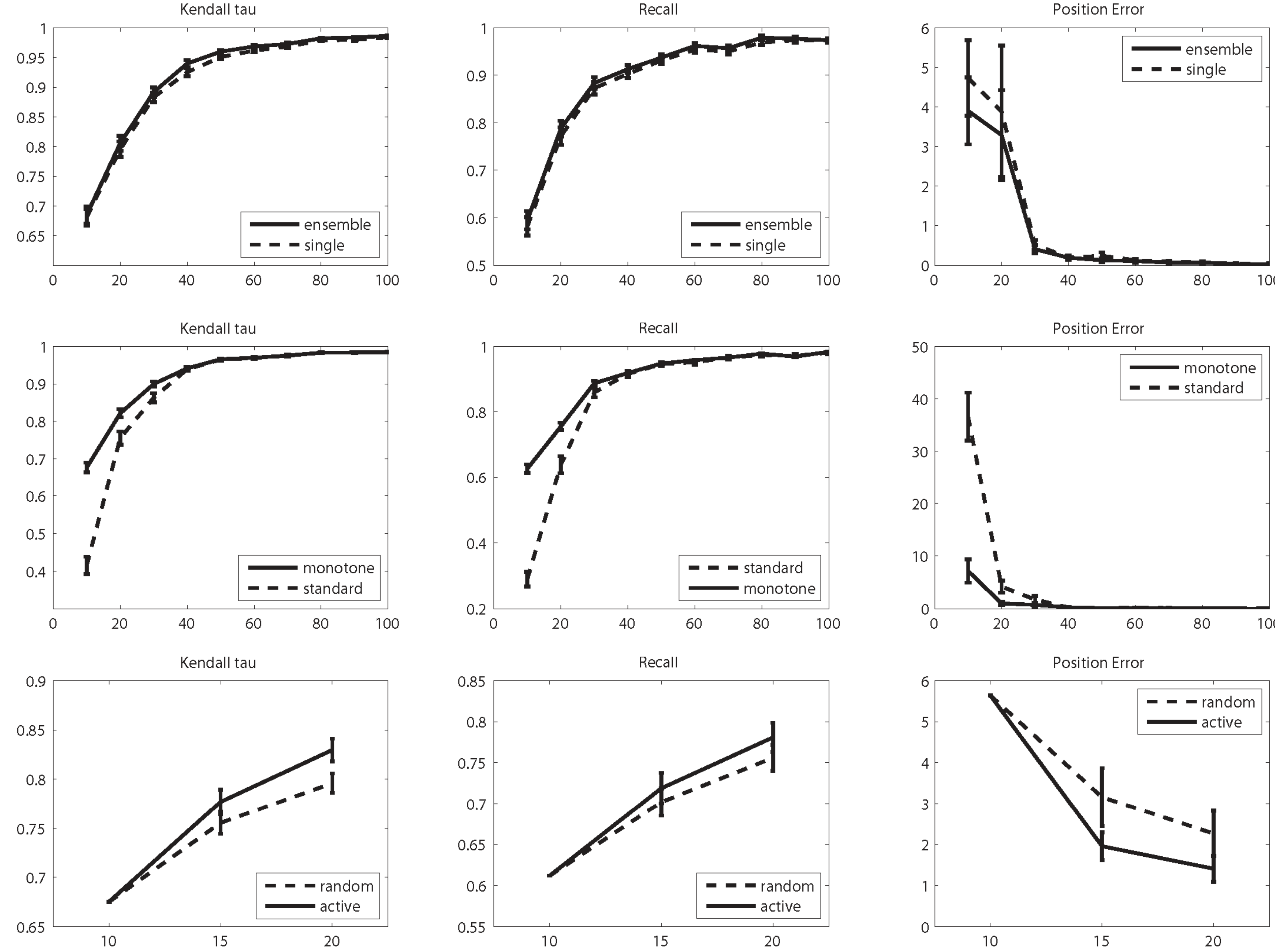
  … defined as number of predicted among true top-k cases (k=10):

  $$\frac{\#(\mathcal{K} \cap \mathcal{K}^{est})}{k}$$

- Position error

  … defined by the position of true topmost case (minus 1):

  $$\pi^{est}\left(\pi^{-1}(1)\right) - 1$$


Kendall tau
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
20
40
60
80
100
ensemble
single
Recall
1
0.9
0.8
0.7
0.6
0.5
0
20
40
60
80
100
ensemble
single
Position Error
6
5
4
3
2
1
0
0
20
40
60
80
100
ensemble
single
Kendall tau
1
0.9
0.8
0.7
0.6
0.5
0.4
0
20
40
60
80
100
monotone
standard
Recall
1
0.8
0.6
0.4
0.2
0
20
40
60
80
100
standard
monotone
Position Error
50
40
30
20
10
0
0
20
40
60
80
100
monotone
standard
Kendall tau
0.9
0.85
0.8
0.75
0.7
0.65
10
15
20
random
active
Recall
0.85
0.8
0.75
0.7
0.65
0.6
0.55
10
15
20
random
active
Position Error
6
5
4
3
2
1
0
10
15
20
random
active

Extension to Nonlinear Models

- Actually, we only need linearity in the coefficients, not in the local distances. Therefore, some generalizations are easily possible, such as

$$\Delta(\mathbf{a},\mathbf{b}) = \sum_{i=1}^{d} w_i \cdot \delta_i(\mathbf{a},\mathbf{b}) + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} \, \delta_i(\mathbf{a},\mathbf{b})\delta_j(\mathbf{a},\mathbf{b})$$

- More generally, with $\boldsymbol{\delta}(\mathbf{a},\mathbf{b}) = (\,\delta_1(\mathbf{a},\mathbf{b}) \ldots \delta_d(\mathbf{a},\mathbf{b})\,)$:

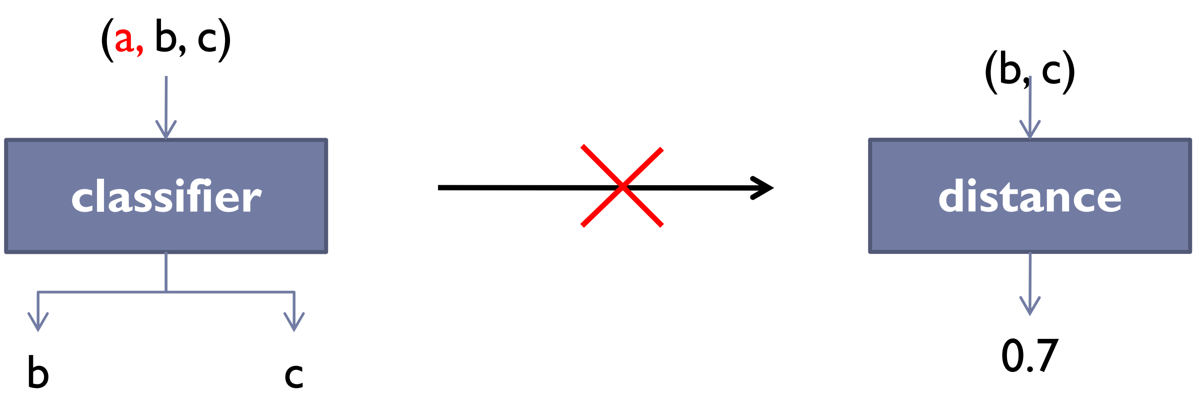
$$\Delta(\mathbf{a},\mathbf{b}) = \langle \mathbf{v}, \phi(\boldsymbol{\delta}(\mathbf{a},\mathbf{b}))\rangle = \sum_{\ell=1}^{k} v_\ell \cdot \phi_\ell(\boldsymbol{\delta}(\mathbf{a},\mathbf{b}))$$

Extensions

- Special case of a kernel function leads to kernelization:

$$\begin{aligned}\Delta(\mathbf{a},\mathbf{b}) &= \sum_{l=1}^{k} v_l\,\phi_l(\boldsymbol{\delta}(\mathbf{a},\mathbf{b})) \\ &= \sum_i \alpha_i\big(K(\boldsymbol{\delta}(\mathbf{a},\mathbf{b}),\boldsymbol{\delta}(\mathbf{a}_i,\mathbf{c}_i)) - K(\boldsymbol{\delta}(\mathbf{a},\mathbf{b}),\boldsymbol{\delta}(\mathbf{a}_i,\mathbf{b}_i))\big)\end{aligned}$$

- Nonlinear classification and sorting



Conclusions

- Learning to combine local distance measures into a global measure.
- Only assuming qualitative feedback of the type "a is more similar to b than to c".
- Reduction of distance learning to classification.