Fuzzy Set Theory in Computer Vision: **Example 7**

Derek T. Anderson and James M. Keller

FUZZ-IEEE, July 2017

Linear Convex Sum

In most works, we assume that a kernel $\mathcal{K}$ is composed by a weighted combination of pre-computed kernel matrices, i.e.,

$$\mathcal{K} = \sum_{k=1}^{m} \sigma_k K_k,$$

where there are m kernels and σ_k is the weight applied to the kth kernel. The domain of σ is very important and many MKL implementations only work for a single domain. For example, $\Delta_2 = \{\sigma \in \mathcal{R}^m : \|\sigma\|_2 = 1, \sigma_k \geq 0\}$ is the ℓ_2-norm MKL.

Geometric way to think about LCS

$$\mathcal{K}_{ij} = \langle \phi_\sigma(\mathbf{x}_i), \phi_\sigma(\mathbf{x}_j) \rangle = \sum_{k=1}^{m} \sigma_k \left(K_k\right)_{ij} = \begin{pmatrix} \sqrt{\sigma_1}\phi_i^1 \\ \dots \\ \sqrt{\sigma_m}\phi_i^m \end{pmatrix}^t \begin{pmatrix} \sqrt{\sigma_1}\phi_j^1 \\ \dots \\ \sqrt{\sigma_m}\phi_j^m \end{pmatrix}$$

FI-based genetic algorithm (FIGA)

1. Compute kernel matrices $K_k = [\kappa_k(\mathbf{x}_i, \mathbf{x}_j)]^{n \times n}$, $k = 1, \ldots, m$;

FI-based genetic algorithm (FIGA)

1. Compute kernel matrices $K_k = [\kappa_k(\mathbf{x}_i, \mathbf{x}_j)]^{n \times n}$, $k = 1, \ldots, m$;
2. Train a base-learner (e.g., SVM) on each kernel K_k and record the classification accuracy η_k, $k = 1, \ldots, m$;

FI-based genetic algorithm (FIGA)

1. Compute kernel matrices $K_k = [\kappa_k(\mathbf{x}_i, \mathbf{x}_j)]^{n \times n}$, $k = 1, \ldots, m$;
2. Train a base-learner (e.g., SVM) on each kernel K_k and record the classification accuracy η_k, $k = 1, \ldots, m$;
3. Collect sorting indices π, such that $\eta_{\pi(1)} \geq \eta_{\pi(2)} \geq \ldots \geq \eta_{\pi(k)}$;

FI-based genetic algorithm (FIGA)

1. Compute kernel matrices $K_k = [\kappa_k(\mathbf{x}_i, \mathbf{x}_j)]^{n \times n}$, $k = 1, \ldots, m$;
2. Train a base-learner (e.g., SVM) on each kernel K_k and record the classification accuracy η_k, $k = 1, \ldots, m$;
3. Collect sorting indices π, such that $\eta_{\pi(1)} \geq \eta_{\pi(2)} \geq \ldots \geq \eta_{\pi(k)}$;
4. Use a GA to learn the FM g, such that the classification accuracy of a learner (e.g., SVM) on $\mathcal{K}$ is maximized

GAMKLp

- Fixed sort sadly reduces FIGA to LCS

GAMKLp

- Fixed sort sadly reduces FIGA to LCS
- GAMKL_p uses a GA to learn the weights $\sigma \in \Delta_p$

GAMKLp

- Fixed sort sadly reduces FIGA to LCS
- GAMKL_p uses a GA to learn the weights $\sigma \in \Delta_p$
- When $p = 1$, we have shown that this is equivalent to FIGA

GAMKLp

- Fixed sort sadly reduces FIGA to LCS
- GAMKL_p uses a GA to learn the weights $\sigma \in \Delta_p$
- When $p = 1$, we have shown that this is equivalent to FIGA
- The genes of the GAMKL_p algorithm are the values of the m weights, i.e., $\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$

GAMKLp

- Fixed sort sadly reduces FIGA to LCS
- GAMKL_p uses a GA to learn the weights $\sigma \in \Delta_p$
- When $p = 1$, we have shown that this is equivalent to FIGA
- The genes of the GAMKL_p algorithm are the values of the m weights, i.e., $\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$
- To ensure the GAMKL_p genes lie in the ℓ_p-norm domain Δ_p, all candidate genes $\tilde{\sigma}$ are ℓ_p-norm normalized to form σ as

$$\sigma = \frac{\tilde{\sigma}}{\sqrt[p]{\sum_{i=1}^{m} |\tilde{\sigma}_i|^p}}$$

Nonlinear MK

- Not an easy thing

Nonlinear MK

- Not an easy thing
- How to determine the sort?

Nonlinear MK

- Not an easy thing
- How to determine the sort?
- $N!$ “walks” w.r.t. the ChI

Nonlinear MK

- Not an easy thing
- How to determine the sort?
- $N!$ "walks" w.r.t. the ChI
- Each "walk" is a different feature space ...

Nonlinear MK

- Not an easy thing
- How to determine the sort?
- $N!$ "walks" w.r.t. the ChI
- Each "walk" is a different feature space ...
- Of course, this are FI centric questions