Kernel on Bag of Paths For Measuring Shapes Similarity

Suard Frédéric, Rakotomamonjy Alain and Bensrhair Abdelaziz

Laboratoire Informatique, Traitement de l'Information, Systèmes EA4051
Institut National des Sciences Appliquées (INSA) de Rouen, Université de Rouen,
Email: frederic.suard@insa-rouen.fr
http://asi.insa-rouen.fr/~fsuard/

Introduction

Context :

- shape recognition,
- structured representation,
- application to shock graph retrieval.

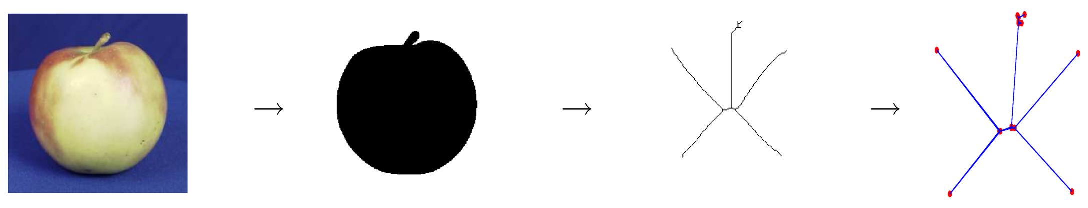
Using graph for comparison :

- shape represented with skeleton,
- skeleton comparison ⇔ graph comparison,
- graph kernel : kernel on set approach,
- graph : set of subgraphs, set of paths.

Shape representation with graph

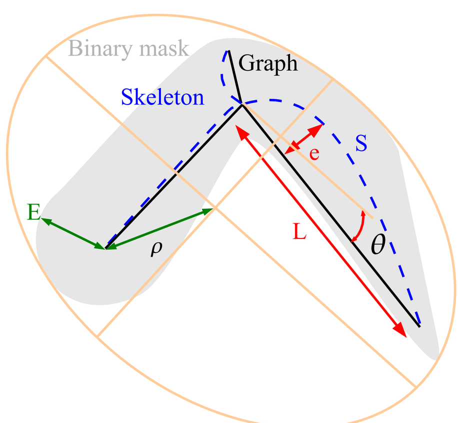
How to build a graph ?

1. original image,
2. binary mask,
3. morphological skeleton,
4. transform skeleton into graph :
   - vertices : skeleton branch ending or branches intersection,
   - edges.



5. label graph :
   edge :
   - length (L),
   - skeleton length (s),
   - orientation (θ),
   - distance between skeleton and edge (e).

   vertex :
   - distance to the center of mass (ρ),
   - distance to the nearest shape edge (E).



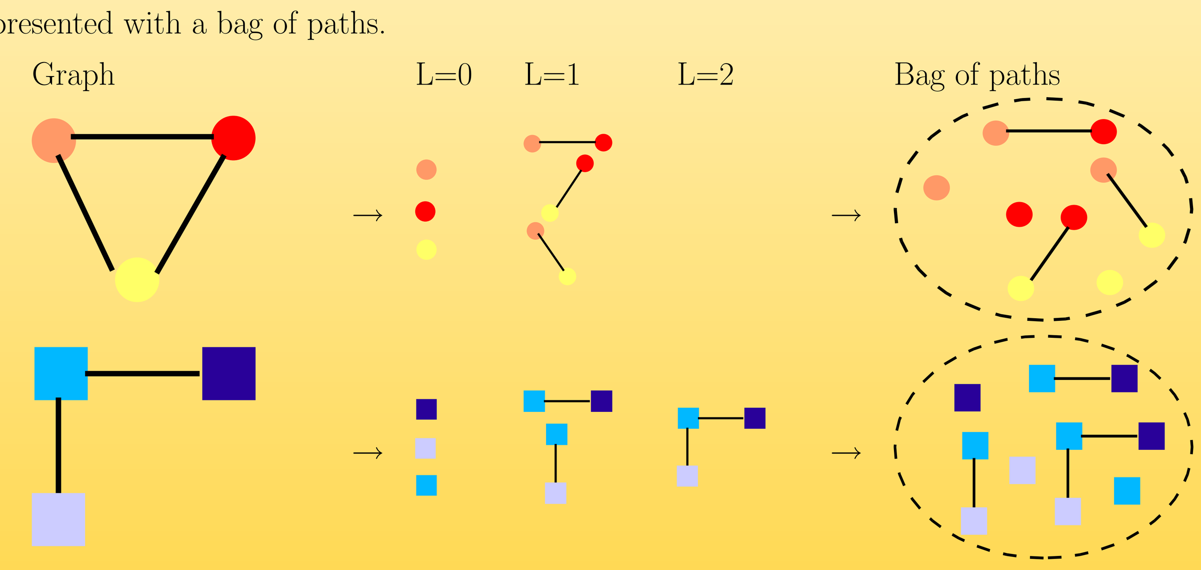
Set of paths

- V : set of vertices,
- E : set of edges,
- labeled graph : $G = (V, E)$,
- path : $h = \{v_1, \cdots, v_n\}$
  - generated using random walk or shortest path,
  - threshold for path length.

⇒ h is a subgraph of G

- set of m paths : $P = \{h_1, h_2, ..., h_m\}$

⇒ G is represented with a bag of paths.



Similarity between path :

$$K_L(h, h') = \begin{cases} 0 & \text{, if path lengths are different} \\ K_v(l(v_1), l(v_1')) \prod_{i=2}^{n} K_e(l(v_{i-1}, v_i), l(v_{i-1}', v_i')) K_v(l(v_i), l(v_i')) & \text{, otherwise} \end{cases}$$

- K_v kernel on vertex label,
- K_e kernel on edge label.

References

[1] F. Demirci, A. Shokoufandeh, L. Bretzner, and S. Dickinson. Object recognition as many-to-many feature matching. *Internation Journal of Computer Vision*, 69(2):203–222, 2006.

[2] F. Desobry, M. Davy, and W.J. Fitzgerald. A class of kernels for sets of vectors. In *Proceedings of the 13th European Symposium on Artificial Neural Networks*, 2005.

[3] B. Haasdonk and C. Bahlmann. Learning with distance substitution kernels. In Springer, editor, *Pattern Recognition - Proc. of the 26th DAGM Symposium*, 2004.

[4] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

[5] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing shock graphs. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2001.

[6] C. Wallraven, B.Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *Proceedings of International Conference on Computer Vision*, pages 257–264, 2003.

Graph kernel

Wallraven [6] : building a kernel on set ⇔ computing a similarity between each element of sets :

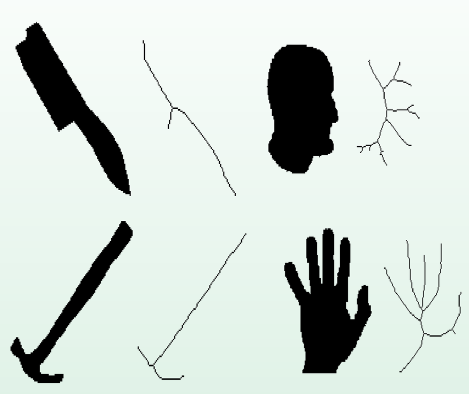
$$K(G, G') = \sum_{h} \sum_{h'} K_L(h, h')$$

- **Kashima [4]:** $K(G_1, G_2) = \sum_{h_1, h_2 \in V_1^\star, V_2^\star} p_1(h_1) p_2(h_2) K_L(h_1, h_2)$
  - p_1, p_2 : probabilities to generate random walk,
  - positive definite.
- **Mean kernel :** $K(G_1, G_2) = K(P_1, P_2) = \frac{1}{N_1}\frac{1}{N_2} \sum_{i:h_i \in P_1} \sum_{j:h_j \in P_2} K_L(h_i, h_j)$
  - all paths are compared,
  - positive definite.
- **Max matching kernel :** $K(G_1, G_2) = K(P_1, P_2) = \frac{1}{2}[\hat{K}(P_1, P_2) + \hat{K}(P_2, P_1)]$
  - best comparison retained,
  - $\hat{K}(P_1, P_2) = \frac{1}{|P_1|} \sum_{i:h_i \in P_1} \max_{j:h_j \in P_2} K_L(h_i, h_j)$
- **Matching kernel :** $\hat{K}(P_1, P_2) = \frac{1}{|P_1|}\frac{1}{|P_2|} \sum_{i:h_i \in P_1} \sum_{j:h_j \in P_2} K_{d_L}(h_i, h_j)$
  - Haasdonk [3] : $\max_{j:h_j \in P_2} K_L(h_i, h_j)$ approximated with $\sum_{j:h_j \in P_2} K_{d_L}(h_i, h_j)$
  - $K_{d_L}(h_1, h_2) = \exp\left(-\frac{d_L(h_1,h_2)^2}{2\sigma^2}\right)$,
  - $d_L(h_1, h_2)^2 = K_L(h_1, h_1) + K_L(h_2, h_2) - 2K_L(h_1, h_2)$.
  - positive definite for all $\sigma > 0$.
- **Path level set kernel :** $K(G_1, G_2) = K(P_1, P_2) = \langle b_{P_1}, b_{P_2} \rangle \cdot \sum_{i:h_i \in P_1} \sum_{j:h_j \in P_2} \alpha_i^{P_1} \alpha_j^{P_2} K_L(h_i, h_j)$
  - Desobry [2] : comparing the probability distribution of each set of paths using one-class SVM,
  - optimization problem : $\min_{f \in \mathcal{H}, b} \frac{1}{2}\|f\|_{\mathcal{H}}^2 + \frac{1}{\nu n} \sum_i \max(0, b - f(x_i)) - b$
  - $f_{P_1}(h) = \sum_i \alpha_i^{P_1} K_L(h_i, h) - b_{P_1}$ and $f_{P_2}(h) = \sum_j \alpha_j^{P_2} K_L(h_j, h) - b_{P_2}$
  - positive definite.

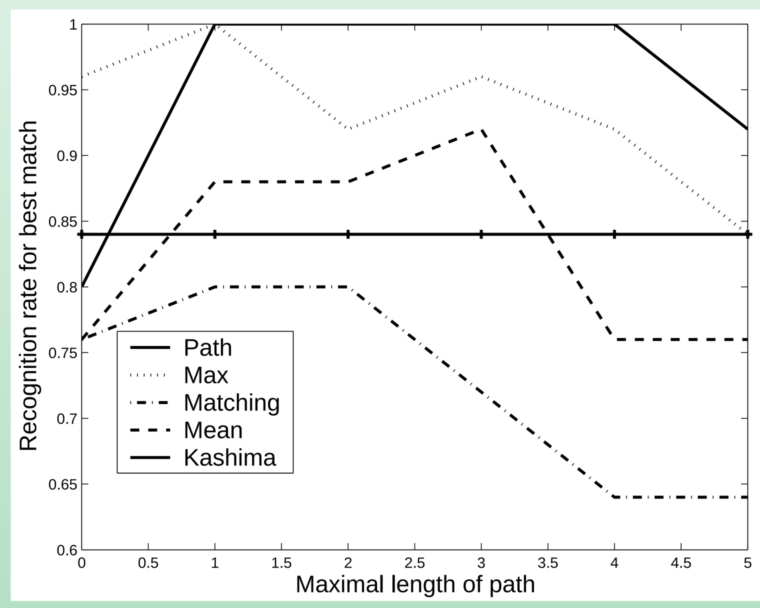
Shock graph mining

- Rudger tools database :25 objects, 8 classes,
- shape retrieval : ranking for each shape query,
- vertex label : E, ρ,
- edge label : L.

→ compare graph formulations
→ path length influence



Performance of each graph kernel according to path length :



- Sebastian [5], Demirci [1] : 96%,
- path-level set, max matching kernel : 100%,
- path length : 2 or 3 for best performance.

Query result for each object of the database :

Conclusion

Conclusions :

- Graph kernel for measuring graph similarity,
- New graph kernel formulations,
- Application to shock graph mining.

Perspectives :

- Add other information : local histograms, texture,
- Explore other ways for graph designing : points of interest.