Active & Interactive Graph Matching

Francesc Serratosa
Universitat Rovira i Virgili
Active & Interactive Graph Matching

**Graph matching:** The aim is to find the best labelling between nodes of two graphs such that the cost of this labelling is the minimum among all possible labellings.

**Active learning:** The aim is to achieve a greater accuracy with fewer classified training examples through choosing the data from which it learns.

**Interactive learning:** The aim is to query some selected data and present it to an oracle (automatic system or a human annotator) for correctly classify it.

**Our model:**
- The learner queries the graph node that it is supposed to produce a greater impact on the labelling between both graphs.
- The oracle answers which is the graph node of the second graph that it has to be matched.
Image Correspondence Process with human interaction
Interactive Graph Matching
Active & Interactive Graph Matching
Structural Pattern Recognition based on Graphs

\[ C_f(g^1, g^2) = \sum_{v^1_i \in \Sigma^1_v} c_v(v^1_i, v^2_\alpha) + \sum_{e^1_{ij} \in \Sigma^1_e} c_e(e^1_{ij}, e^2_{aab}) \]
Interactive Graph Matching

Node Costs:

If $True(v_1^1, v_2^2)$ then
- $C_v[i, a] = 0$
- $C_v[i, b] = \infty \forall b \neq a$
- $C_v[j, a] = \infty \forall j \neq i$

If $False(v_1^1, v_2^2)$ then
- $C_v[i, a] = \infty$

Arc Costs:

If $True(v_1^1, v_2^2) \land True(v_1^2, v_2^2)$ then
- $C_e[i, j, a, b] = 0 \land C_e[j, i, b, a] = 0$
- $C_e[i, j, a', b'] = \infty \land C_e[j, i, a', b'] = \infty \forall a \neq a'$ and $b \neq b'$
- $C_e[i', j', a, b] = \infty \land C_e[i', j', b, a] = \infty \forall i \neq i'$ and $j \neq j'$

If $False(v_1^1, v_2^2)$ then
- $C_e[i, j', a, b'] = \infty \land C_e[j', i, b', a] = \infty \land C_e[j', i, a, b'] = \infty \land C_e[j', i, a, b'] = \infty \forall i \neq j'$ and $a \neq b'$
Active & Interactive Graph Matching

Algorithm Active Graph Matching
Input: Attributed Graphs $g^1$ and $g^2$
Output: Labelling $f$ and Cost $C_f$
$C_v^0, C_e^0 = Initialise_Cost(g^1, g^2); C_v = C_v^0; C_e = C_e^0.$
$f = Graph\_Matching(C_v, C_e).$

Do
\begin{align*}
  v^1* &= Active\_Query(P, f).
  v^2* &= Oracle\_Feedback(g^1, g^2, v^1*, f).
  w_1 &= Set(v^1, v^2*).
  C_v &= Interactive\_Node\_Costs(w, C_v).
  C_e &= Interactive\_Edge\_Costs(w, C_e).
  f &= Graph\_Matching(C_v, C_e).
\end{align*}

Since Stop
Compute $C_f(C_v, C_e)$
End Algorithm
Active Learning Strategies

Four strategies to select a node $v^1*$ of $g^1$ that have to be queried to an oracle.
The oracle feedback is the node of $g^2$: $v^{2*} = f(v^{1*})$

*Least Confident (LC)*

*Least Confident given the Current Labelling (LCCL)*

*Maximum Entropy (ME)*

*Expected Model Change (EMC)*
Least Confident (LC)

This strategy queries the node that its highest probability of belonging to a class is the lower one between all the elements.

\[ v^{2(i)} = \arg\max_{\forall j=\{1,\ldots,n\}} P[v^1_i, v^2_j]; \forall i = \{1,\ldots,n\} \]

\[ v^{1*}_{\text{LC}} = \arg\min_{\forall i=\{1,\ldots,n\}|Q(i) = \text{False}} P[v^1_i, v^{2(i)}] \]
Least Confident given the Current Labelling (LCCL)

This strategy queries the node that has the minimum probability given the current labelling.

\[ v_{LCCL}^* = \arg\min_{\forall i=\{1,\ldots,n\}|Q(i) = \text{False}} P[v_i^1, f(v_i^1)] \]
Maximum Entropy (ME)

This strategy queries the node that has the maximum Shannon Entropy given the probabilities.

\[
v_{ME}^{1*} = \arg\max_{\forall i = \{1, \ldots, n\} | Q(i) = False} - \sum_{j=1}^{n} P[v_i^1, v_j^2] \cdot \log(P[v_i^1, v_j^2])
\]
Expected Model Change (EMC)

This strategy queries the node that would impart the greatest change to the current labelling if we knew its class.

\[ R_i = \max_{v_j \in \{1, \ldots, n\}} \{P[v_i^1, v_j^2]\} - P[v_i^1, f(v_i^1)] \]

\[ v_{EMC}^1 = \arg\max_{\forall i \in \{1, \ldots, n\} \land Q(i) = False} \{R_i\} \]
Active & Interactive Graph Matching
Practical Evaluation

Hamming distance respect of the number of iterations

Hotel

House

LCCL: , LC: , ME: , EMC: and Random:
Practical Evaluation

Matching cost respect of the number of iterations

![Graphs showing matching cost over iterations for Hotel and House, with different methods indicated: LCCL, LC, ME, EMC, and Random.](image)
Conclusions

- Four different strategies to be applied on an active graph-matching algorithm

- It is not needed to modify the code of the graph matching algorithms:
  - They read the probability matrix and
  - write the matrix costs

- Experimental validation shows that the Least Confident with Current labelling (LCCL) tends faster to find the optimal labelling