Learning Functions in Graphical Domains

Marco Gori
University of Siena (Italy)
Outline

• Learning functions in graphical domains (graphs in data ... graphs in functions) - motivations

• Graphical Neural Networks

• Experimental results
Paradigm Shift?

• Learning functions defined on the nodes (outputs on nodes, arcs)

• The learning environment:
  • A *unique graph* (a collection of “nodes” can always be regarded as a special disconnected graph)
  • Graphs to express the examples and the relationships between the examples
Re-foundation of Calculus
(see e.g. A. Bensoussan & J.-L. Menaldi)

Difference Equations on Weighted Graphs (with A. Bensoussan). (Journal of Convex Analysis (Special issue in honor of Claude LeMarechal), 12 (2005), pp. 13-44.)

• Notions of boundary, Hilbert space of functions, norms & semi-norms (Sobolev spaces)
• Extensions of differential operators ... Green’s formula
• Variational problems, Harmonic functions, Dirichlet & Neumann’ problems
• Relationships with random walk
What’s a Castle?

Supervision: which parts compose the castle? (back nodes)

Segmentation and recognition: perhaps ... it’s better not to separate in two steps ...

DaMn Bertinoro 2005
Ranking Web Pages

Rank?

absolute/relative

set a *supervision value* (topology & content)

we put supervision on “some nodes” and ask inference on the rest ... if I can set relationships ... the same problem holds everywhere

The distinction learning set/test *has a different meaning!*

DaMn Bertinoro 2005
Web Spam?

is this spam?

absolute/relative

www.ing.unisi.it  www.ing.unisi.it/people

www.ing.unisi.it/~franco

www.ing.unisi.it/~marco

www.uow.edu.au/~markus

www.uow.edu.au/~act

Again: set a *supervision value* (topology & content)

DaMn Bertinoro 2005
Quantitative structure activity relationship (QSAR)

Unlike previous two examples ... a global property
Formal Statement

\[ \mathcal{L} = (G, T) \text{ where } G = (N, E) \text{ is graph} \]

\[ T \text{ a is set of pairs } \{(n_i, t_i)| n_i \in N, t_i \in \mathbb{R}^m, 1 \leq q\} \]

A unique graph!
A target on some nodes (arcs)
A State-Based Model

Local computation

We force relaxation to the fixed point!

\[ x_1 = f_w(l_1, x_2, x_3, x_4, x_6, l_{(1,2)}, l_{(3,1)}, l_{(1,4)}, l_{(6,1)}, l_2, l_3, l_4, l_6, \emptyset, 1, 0, 1) \]

The neighbor of node 1

DaMn Bertinoro 2005
What Does Learning Mean?

Banach fixed-point Theorem

\[ e_w = \sum_{i=1}^{q} (t_i - \varphi_w(G, n_i))^2 + \beta L \left( \| \frac{\partial F_w}{\partial x} \| \right) \]

\( L(y) \) is \((y - \mu)^2 \) if \( y > \mu \) and 0, otherwise, and the parameter \( \mu \in (0, 1) \)
Computational Capabilities

Crucial question

The system dynamics is imposed by the “contraction map hyp”. Does it limit somehow the computational capabilities?
Unfolding Equivance

non-equivalent nodes

equivalent nodes

unfolding trees having depth 3

DaMn Bertinoro 2005
Preserving the Unfolding-Equivalence

Definition 3 A function \( l : G \times N \rightarrow \mathbb{R}^m \) is said to preserve the unfolding equivalence on \( G \), if \( n \sim u \) implies \( l(G, n) = l(G, u) \), for any nodes \( n, u \) of \( nG \). The set of functions that preserve the unfolding equivalence on \( G \) will be denoted by \( \mathcal{F}(G) \).

“Everything” preserving the unfolding-equivalence can be calculated

DaMn Bertinoro 2005
What Cannot Be Calculated?

Warning!
The symmetry constraints

Impossible!
The function doesn’t preserve unfolding equivalence

DaMn Bertinoro 2005
(Efficient) Gradient Computation

Classic Problem

Calculs of Variations, control literature

Almeida/Pineda (ANNs)

DaMn Bertinoro 2005
Experimental Results

Learning tasks in which the concept is based on:

- the topologic properties only
- the content of the nodes only
- the joint dependence on both topology and content
The Clique Problem

“1” if the node is part of a 3-clique

no information in the nodes!
The Clique Problem

• 2000 random graphs, 20 nodes, 300 training set, 300 validation set

• 5-clique, 71.8% is “1”

• 2 states, 2-10 hidden units

• 84.6% on test set
## Subgraph matching

Like for the “castle problem”
the whole picture
the “castle”

600 graphs (TS, VS, TS) random labels, but S is in G
both label and links
s=5, h=5

---

<table>
<thead>
<tr>
<th>Number of nodes in S</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>9</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>neural</td>
<td>92.4</td>
<td>90.0</td>
<td>90.0</td>
<td>84.3</td>
<td>89.1</td>
</tr>
<tr>
<td>linear</td>
<td>93.3</td>
<td>84.5</td>
<td>86.7</td>
<td>84.7</td>
<td>87.3</td>
</tr>
<tr>
<td>FNN</td>
<td>81.4</td>
<td>78.2</td>
<td>79.6</td>
<td>82.2</td>
<td>80.3</td>
</tr>
<tr>
<td>neural</td>
<td>91.3</td>
<td>87.7</td>
<td>84.9</td>
<td>83.3</td>
<td>86.8</td>
</tr>
<tr>
<td>linear</td>
<td>90.4</td>
<td>85.8</td>
<td>85.3</td>
<td>80.6</td>
<td>85.5</td>
</tr>
<tr>
<td>FNN</td>
<td>85.2</td>
<td>73.2</td>
<td>65.2</td>
<td>75.5</td>
<td>74.8</td>
</tr>
<tr>
<td>neural</td>
<td>89.8</td>
<td>84.6</td>
<td>79.9</td>
<td></td>
<td>84.8</td>
</tr>
<tr>
<td>linear</td>
<td>91.3</td>
<td>84.4</td>
<td>79.2</td>
<td></td>
<td>85.0</td>
</tr>
<tr>
<td>FNN</td>
<td>84.2</td>
<td>66.9</td>
<td>64.6</td>
<td></td>
<td>71.9</td>
</tr>
<tr>
<td>neural</td>
<td>93.3</td>
<td>84.0</td>
<td>77.8</td>
<td></td>
<td>85.0</td>
</tr>
<tr>
<td>linear</td>
<td>92.2</td>
<td>84.0</td>
<td>77.7</td>
<td></td>
<td>84.7</td>
</tr>
<tr>
<td>FNN</td>
<td>91.6</td>
<td>73.7</td>
<td>67.0</td>
<td></td>
<td>77.4</td>
</tr>
<tr>
<td>neural</td>
<td>91.8</td>
<td>90.2</td>
<td>85.9</td>
<td>81.3</td>
<td></td>
</tr>
<tr>
<td>linear</td>
<td>91.9</td>
<td>88.5</td>
<td>85.1</td>
<td>80.6</td>
<td></td>
</tr>
<tr>
<td>FNN</td>
<td>83.3</td>
<td>81.8</td>
<td>71.3</td>
<td>72.3</td>
<td></td>
</tr>
</tbody>
</table>

**Total**
- neural: 87.3
- linear: 86.5
- FNN: 77.2

---

G: 30 nodes, S: 5 nodes
only label of the node
The Half-Hot Problem

\[ \tau(G, n) = 1 \text{ for a half of the nodes of } G \text{ and } \tau(G, n) = -1 \]

300 regular graphs 4-10 nodes

it doesn’t preserve unfolding equivalence!

\[ e_w = \sum_{n,u \in N, n \neq u} (o_n - 1)(o_u - 1) \]

\[ \delta_i = \frac{|G_i|}{2} - b. \]

hot nodes
Learning PR-like Functions

Learn the PageRank (PR)
Each node has a label with 2 Booleans

\[ o_i = \begin{cases} 
2 \times PR_i & \text{if } a = 0, b = 1 \text{ or } a = 1, b = 0 \\
PR_i & \text{otherwise}
\end{cases} \]

5,000 nodes
50 + 50 (Training & Validation Sets), 5 hidden units (linear model)

DaMn Bertinoro 2005
Experimental steps

• s/w based on MathLab 7.0.1

• It’s evolving ... updating of feature
  (functions on arcs, non-stationarity, ...)

• download s/w at http://airgroup.dii.unisi.it
Conclusions

- Graphs in data ... graphs in functions (Compiling data & function on the same graph)

- Only preliminary experimental results ...

- We’d love trying with problems like Web spam, social networks (blogs)!

- Related approaches to be investigated (mainly with Kernel machines: Risi Kondor, Bernard Schoelkopf), but also with ILP + Statistics
Related Papers

download from http://nautilus.ing.unisi.it


• A technical report (draft)

Comments very welcome!
Thanks for your attention!