Utilisation of computational intelligence for simplification of linear objects using extended WEA algorithm

Anna Fiedukowicz
Agata Pillich-Kolipińska
Robert Olszewski

1 This work has been supported by the European Union in the framework of European Social Fund through the Warsaw University of Technology Development Programme.

16th ICA Generalisation Workshop, Dresden, 23–24.08.2013
Line simplification

- Widely known and described
- Selection of „important“ vertices
- Different algorithms, mostly one–parameter ones
  - Douglas–Peucker (1973) – distance
  - Wang (1996) – curve radius
  - many others…
Weighted Effective Area

- „Shape-aware” algorithm
  - „effective area” from Visvalingam–Whyatt algorithm
  - flatness (H/W)
  - skewness (H/ML)
  - convexity (↻ – convex, ↺ – concave)

- Defined filters:
  - $W_{\text{flat}}$
  - $W_{\text{skew}}$
  - $W_{\text{convex}}$

$W_{\text{EA}} = W_{\text{flat}} \times W_{\text{skew}} \times W_{\text{convex}} \times EA$
Modified WEA (2)

Non-deterministic approach: knowledge base

Explicit
set of rules
defined by an expert

Implicit
derived from examples
provided by an expert

computational intelligence

Fuzzy Inference Systems

Artificial Neural Networks

Vertex weight = f(\text{flatness, skewness, convexity, EA})

Modified „WEA” (weight of a vertex)
Modified WEA (2)

Vertex weight = \( f(\text{flatness}, \text{skewness}, \text{convexity}, \text{EA}) \)

<table>
<thead>
<tr>
<th>rules</th>
<th>examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defined by an expert, eg.</td>
<td>Important points derived from Topographic DBs and maps 1:10k, 1:50k, 1:250k</td>
</tr>
<tr>
<td>‣ if (EA &gt; 100) and (\text{flatness} &gt; 1) and (\text{skewness} &gt; 0.8 ) then weight = 99;</td>
<td>‣ Polish coastline (SABE 30 and surveying data)</td>
</tr>
<tr>
<td><em>Or more fuzzy definition:</em></td>
<td>‣ Polish rivers</td>
</tr>
<tr>
<td>‣ if EA is big and \text{flatness} is high and \text{skewness} is low then vertex is important</td>
<td>‣ Vertices weights proposed by an expert</td>
</tr>
</tbody>
</table>
Finding $f()$ – ANN (1)

- STATISTICA Neural Networks software
Three types of ANNs tested

Different teaching algorithms tested (backward errors propagation, quasi–Newton, Delta–Bar–Delta, Levenberg–Marquardt etc.)
Testing different structures of same ANN type

More neurons ⇒ higher precision ⇒ output
less generalized ⇒ learning more time-consuming ⇒ longer data processing

Test data – several neurons?
Actual data – several thousands neurons?
ANN – results

WEA NEURO algorithm

N = 456 points
K = 132 points
Rules based knowledge base

- IF building $< 400m^2$ THEN $y = 0.05$

A $\rightarrow$ B

- IF building is small THEN simplify a bit

How you define „small”?
Membership functions

=> Linguistic variables
Fuzzy rules

- IF $x = A$ THEN $y = B$

A $\rightarrow$ B

- IF $x = A$ THEN $y = B$
- And what IF $x = A'$ ? THEN $y = B'$

Graph showing fuzzy logic with membership functions for A and B.
IF area = „big” and flatness = „big” then significance = „big”
Fuzzy reasoning results

WEA FUZZY algorithm

N = 456 points
K = 132 points
Conclusions

- Easy to build
- Black box for user
- Long computation time for complex tasks

- Difficult to define
- Allows process understanding
- Builted once works fast

NEURO

FUZZY
State of work

- Tools independent on GIS

- Preliminary tests on polish coast line

- Combine fuzzy and neuro tools with GIS software

- Check tools on different type of data

Up to now

Future plans
Thank you for your attention!

Here it comes, here comes the weekend...

r.olszewski@gik.pw.edu.pl
a.pillich@gik.pw.edu.pl
a.fiedukowicz@gik.pw.edu.pl