Medical Image Processing for Interventional Applications

Random Walker – Algorithm

Online Course – Unit 41

Andreas Maier, Stefan Steidl, Frank Schebesch
Pattern Recognition Lab (CS 5)

virtuelle hochschule bayern

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Topics

Problem Statement

K-way image segmentation

- User-defined **seeds**
- Indicating regions of the image belonging to K objects

Random walk

- Labeling an **unseeded** pixel by resolving the question:
  What is the probability of a random walker starting at this pixel that it first reaches seed point k?
- Selecting the label of the most probable seed destination for each pixel
- Biasing the random walker to avoid crossing sharp intensity gradients

Problem Statement

Image as discrete object

- **Graph** with a fixed number of vertices and edges
- Each node represents one pixel in the image.
- Edges connect neighboring pixels: e. g., 4-connectivity (2-D), 6-connectivity (3-D), 8-connectivity (2-D).
- A real-valued **weight** is assigned to each edge representing the likelihood that a random walker will cross this edge.
  $\rightarrow$ Weight of zero: the random walker may not move along that edge.
- Purely combinatorial operators:
  - No discretization
  - No discretization errors or ambiguities

Problem Statement

Edge weights for adjacent pixels *i* and *j*

Gaussian weighting function:

$$w_{ij} = \exp\left(-\beta(g_i - g_j)^2\right)$$

where

- g_i: image intensity at pixel i
- β: only free parameter!

- Useful operation: prior normalization of the square gradients:

$$\forall e_{ij} \in E: \ (g_i - g_j)^2 \in [0, 1]$$

- Modification to handle color or general vector-valued data: $(g_i - g_j)^2 \longrightarrow ||\boldsymbol{g}_i - \boldsymbol{g}_j||^2$

Problem Statement

Four mathematically equivalent ways (Grady, 2006)

1. *“If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s.”*

Problem Statement

Four mathematically equivalent ways (Grady, 2006)

1. *“If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s.”*
2. *“If the seeds are alternately replaced by grounds/unit voltage sources, assign the pixel to the label for which its seeds being ‘on’ produces the greatest electrical potential.”*

Problem Statement

Four mathematically equivalent ways (Grady, 2006)

1. *"If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s."*
2. *"If the seeds are alternately replaced by grounds/unit voltage sources, assign the pixel to the label for which its seeds being 'on' produces the greatest electrical potential."*
3. *"Assign the pixel to the label for which its seeds have the largest effective conductance (i. e., smallest effective resistance) with the pixel."*

Problem Statement

Four mathematically equivalent ways (Grady, 2006)

1. *“If a random walker leaving the pixel is most likely to first reach a seed bearing label s, assign the pixel to label s.”*
2. *“If the seeds are alternately replaced by grounds/unit voltage sources, assign the pixel to the label for which its seeds being ‘on’ produces the greatest electrical potential.”*
3. *“Assign the pixel to the label for which its seeds have the largest effective conductance (i. e., smallest effective resistance) with the pixel.”*
4. *“If a 2-tree is drawn randomly from the graph (with probability given by the product of weights in the 2-tree), assign the pixel to the label for which the pixel is most likely to remain connected to.”*

Topics

Random Walks for Image Segmentation

Algorithm
- Dirichlet Integral
- Decomposition
- Solution

Summary
- Take Home Messages
- Further Readings

Algorithm

Combinatorial Laplacian matrix L

$$L_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -w_{ij} & \text{if } v_i \text{ and } v_j \text{ are adjacent nodes,} \\ 0 & \text{otherwise,} \end{cases}$$

where

- L_{ij} is indexed by vertices v_i and v_j,
- $d_i = \sum w(e_{ij})$ for all edges e_{ij} incident on node v_i.

Algorithm

Example: Pixels of a 4×4 *image*

	1	2	3	4
1	v_1	v_2	v_3	v_4
2	v_5	v_6	v_7	v_8
3	v_9	v_{10}	v_{11}	v_{12}
4	v_{13}	v_{14}	v_{15}	v_{16}

Algorithm

Example: Pixels of a 4×4 *image* and the according *combinatorial Laplacian matrix L*

	1	2	3	4
1	v_1	v_2	v_3	v_4
2	v_5	v_6	v_7	v_8
3	v_9	v_{10}	v_{11}	v_{12}
4	v_{13}	v_{14}	v_{15}	v_{16}

$$L = \left[\begin{array}{cccccccccccccccc}
d_1 & -w_{1,2} & 0 & 0 & -w_{1,5} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-w_{1,2} & d_2 & -w_{2,3} & 0 & 0 & -w_{2,6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -w_{2,3} & d_3 & -w_{3,4} & 0 & 0 & -w_{3,7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -w_{3,4} & d_4 & 0 & 0 & 0 & -w_{4,8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-w_{1,5} & 0 & 0 & 0 & d_5 & -w_{5,6} & 0 & 0 & -w_{5,9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -w_{2,6} & 0 & 0 & -w_{5,6} & d_6 & -w_{6,7} & 0 & 0 & -w_{6,10} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -w_{3,7} & 0 & 0 & -w_{6,7} & d_7 & -w_{7,8} & 0 & 0 & -w_{7,11} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -w_{4,8} & 0 & 0 & -w_{7,8} & d_8 & 0 & 0 & 0 & -w_{8,12} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -w_{5,9} & 0 & 0 & 0 & d_9 & -w_{9,10} & 0 & 0 & -w_{9,13} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -w_{6,10} & 0 & 0 & -w_{9,10} & d_{10} & -w_{10,11} & 0 & 0 & -w_{10,14} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -w_{7,11} & 0 & 0 & -w_{10,11} & d_{11} & -w_{11,12} & 0 & 0 & -w_{11,15} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{8,12} & 0 & 0 & -w_{11,12} & d_{12} & 0 & 0 & 0 & -w_{12,16} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{9,13} & 0 & 0 & 0 & d_{13} & -w_{13,14} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{10,14} & 0 & 0 & -w_{13,14} & d_{14} & -w_{14,15} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{11,15} & 0 & 0 & -w_{14,15} & d_{15} & -w_{15,16} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_{12,16} & 0 & 0 & -w_{15,16} & d_{16}
\end{array}\right]$$

Algorithm

Combinatorial formulation of the Dirichlet integral

$$D(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x} = \frac{1}{2}\sum_{e_{ij}\in E} w_{ij}(x_i - x_j)^2$$

Partitioning the vertices into two sets:

- marked/seed nodes V_M,
- unseeded nodes V_U,

such that $V_M \cup V_U = V$ and $V_M \cap V_U = \varnothing$.

Without loss of generality: The nodes in $\boldsymbol{L}$ and $\boldsymbol{x}$ are **ordered**, i. e., seed nodes are first, unseeded nodes are second.

Algorithm

Decomposition

$$D[\boldsymbol{x}_U] = \frac{1}{2}\left(\boldsymbol{x}_M{}^{\mathsf{T}} \;\; \boldsymbol{x}_U{}^{\mathsf{T}}\right)\begin{bmatrix}\boldsymbol{L}_M & \boldsymbol{B}\\ \boldsymbol{B}^{\mathsf{T}} & \boldsymbol{L}_U\end{bmatrix}\begin{pmatrix}\boldsymbol{x}_M\\ \boldsymbol{x}_U\end{pmatrix} = \frac{1}{2}\left(\boldsymbol{x}_M{}^{\mathsf{T}}\boldsymbol{L}_M\boldsymbol{x}_M + 2\boldsymbol{x}_U{}^{\mathsf{T}}B^{\mathsf{T}}\boldsymbol{x}_M + \boldsymbol{x}_U{}^{\mathsf{T}}\boldsymbol{L}_U\boldsymbol{x}_U\right)$$

$\boldsymbol{L}$ is positive semi-definite, i. e., the only critical points of $D[\boldsymbol{x}]$ will be minima.

Algorithm

Differentiating w. r. t. $\boldsymbol{x}_U$ and finding the critical points:

$$\boldsymbol{L}_U \boldsymbol{x}_U = -\boldsymbol{B}^\top \boldsymbol{x}_M$$

- System of linear equations with $|V_U|$ unknowns
- Equation will be non-singular
  - if the graph is connected, or
  - if every connected component contains a seed.

Algorithm

Solution to the combinatorial Dirichlet problem for label s

- x_i^s: probability (potential) assumed at node v_i for label s
- Set of labels: $\forall v_j \in V_M : \; Q(v_j) = s, s \in \mathbb{Z}, 0 < s \leq K$
- $V_M \times 1$ vector $\boldsymbol{m}^s$:

$$m_j^s = \begin{cases} 1 \text{ if } Q(v_j) = s, \\ 0 \text{ if } Q(v_j) \neq s \end{cases}$$

Algorithm

Solution to the combinatorial Dirichlet problem for label s

- x_i^s: probability (potential) assumed at node v_i for label s
- Set of labels: $\forall v_j \in V_M: \; Q(v_j) = s, s \in \mathbb{Z}, 0 < s \leq K$
- $V_M \times 1$ vector $\boldsymbol{m}^s$:

$$m_j^s = \begin{cases} 1 \text{ if } Q(v_j) = s, \\ 0 \text{ if } Q(v_j) \neq s \end{cases}$$

Solution for one label:

$$\boldsymbol{L}_U \boldsymbol{x}^s = -\boldsymbol{B}^\top \boldsymbol{m}^s$$

Algorithm

Solution to the combinatorial Dirichlet problem for label s

- x_i^s: probability (potential) assumed at node v_i for label s
- Set of labels: $\forall v_j \in V_M: \; Q(v_j) = s, s \in \mathbb{Z}, 0 < s \leq K$
- $V_M \times 1$ vector $\boldsymbol{m}^s$:

$$m_j^s = \begin{cases} 1 \text{ if } Q(v_j) = s, \\ 0 \text{ if } Q(v_j) \neq s \end{cases}$$

Solution for one label:

$$\boldsymbol{L}_U \boldsymbol{x}^s = -\boldsymbol{B}^\top \boldsymbol{m}^s$$

Solution for all labels:

$$\boldsymbol{L}_U \boldsymbol{X} = -\boldsymbol{B}^\top \boldsymbol{M}$$

where $\boldsymbol{X}, \boldsymbol{M}$ are matrices with K columns taken by each $\boldsymbol{x}^s$ and $\boldsymbol{m}^s$, respectively.

Algorithm

Note:

- At any node the probabilities x_i^s will sum to unity:

$$\forall v_i \in V : \sum_s x_i^s = 1 .$$

- Hence, only $K - 1$ sparse linear systems must be solved.

Topics

Random Walks for Image Segmentation

Algorithm
- Dirichlet Integral
- Decomposition
- Solution

Summary
- Take Home Messages
- Further Readings

Take Home Messages

- For the segmentation using a random walker, we describe pixel transitions from one pixel to a neighboring pixel in form of graph edges.
- In the algorithm the combinatorial Dirichlet problem has to be solved to find a segmentation result.

Further Readings

These slides are based on the following publication:

L. Grady. "Random Walks for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.11 (Nov. 2006), pp. 1768–1783. DOI: 10.1109/TPAMI.2006.233

His implementations in Matlab can be downloaded here:

- Graph Analysis Toolbox
- Random Walker