Image Processing using Graphs (lecture 5 - clustering and classification)

Alexandre Xavier Falcão

Visual Informatics Laboratory - Institute of Computing - University of Campinas

afalcao@ic.unicamp.br
www.ic.unicamp.br/~afalcao/talks.html

Introduction

- New technologies for data acquisition and storage have provided large datasets with millions (or more) of samples for statistical analysis.

Introduction

- New technologies for data acquisition and storage have provided large datasets with millions (or more) of samples for statistical analysis.
- We need more efficient and effective pattern recognition methods for large datasets.

Introduction

- New technologies for data acquisition and storage have provided large datasets with millions (or more) of samples for statistical analysis.
- We need more efficient and effective pattern recognition methods for large datasets.
- The applications are in many fields of the sciences and engineering.

Introduction

- New technologies for data acquisition and storage have provided large datasets with millions (or more) of samples for statistical analysis.
- We need more efficient and effective pattern recognition methods for large datasets.
- The applications are in many fields of the sciences and engineering.
- Our main focus has been on image analysis.

Introduction

- Each sample s (spel, image or object) of a dataset $\mathcal{Z}$ can be interpreted as a point of a distance space defined by a simple or composite descriptor.

Introduction

- Each sample s (spel, image or object) of a dataset $\mathcal{Z}$ can be interpreted as a point of a distance space defined by a simple or composite descriptor.
- We wish to design a classifier which can assign the correct label for any sample $s \in \mathcal{Z}$.

Introduction

- Each sample s (spel, image or object) of a dataset $\mathcal{Z}$ can be interpreted as a point of a distance space defined by a simple or composite descriptor.
- We wish to design a classifier which can assign the correct label for any sample $s \in \mathcal{Z}$.
- In supervised learning, a labeled set $\mathcal{T} \subset \mathcal{Z}$ is available to train the classifier.

Introduction

- Each sample s (spel, image or object) of a dataset $\mathcal{Z}$ can be interpreted as a point of a distance space defined by a simple or composite descriptor.
- We wish to design a classifier which can assign the correct label for any sample $s \in \mathcal{Z}$.
- In supervised learning, a labeled set $\mathcal{T} \subset \mathcal{Z}$ is available to train the classifier.
- In unsupervised learning, there is no knowledge about the labels in $\mathcal{T}$. Clusters can be found and class labels may be assigned to them based on some prior knowledge.

Some common mistakes are to assume that

Some common mistakes are to assume that

- the classes/clusters form compact clouds of points in the distance space.

Some common mistakes are to assume that

- the classes/clusters form compact clouds of points in the distance space.
- they do not overlap each other.

Introduction

Some common mistakes are to assume that

- the classes/clusters form compact clouds of points in the distance space.
- they do not overlap each other.
- one cluster corresponds to one class.

Introduction

Some common mistakes are to assume that

- the classes/clusters form compact clouds of points in the distance space.
- they do not overlap each other.
- one cluster corresponds to one class.
- the probability density function of the classes/clusters present known shapes for parametric modeling.

Introduction

- We assume that two samples in a same cluster/class should be at least connected by a chain of nearby samples (transitive property).

Introduction

- We assume that two samples in a same cluster/class should be at least connected by a chain of nearby samples (transitive property).
- A graph $(\mathcal{T}, \mathcal{A})$ is defined by an adjacency relation $\mathcal{A}$ between training samples using the distance space.

Introduction

- We assume that two samples in a same cluster/class should be at least connected by a chain of nearby samples (transitive property).
- A graph $(\mathcal{T}, \mathcal{A})$ is defined by an adjacency relation $\mathcal{A}$ between training samples using the distance space.
- A connectivity function $f(\pi_t)$ assigns a value to any path π_t from its root $R(\pi_t)$ to its terminal node t.

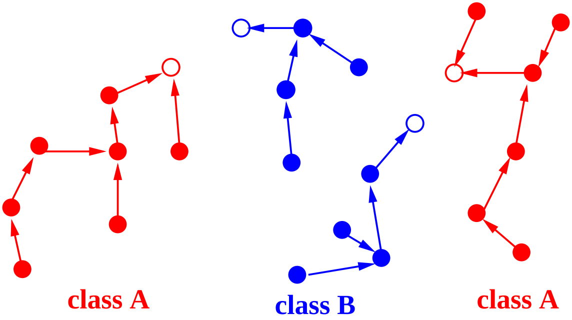
Introduction

- We assume that two samples in a same cluster/class should be at least connected by a chain of nearby samples (transitive property).
- A graph $(\mathcal{T}, \mathcal{A})$ is defined by an adjacency relation $\mathcal{A}$ between training samples using the distance space.
- A connectivity function $f(\pi_t)$ assigns a value to any path π_t from its root $R(\pi_t)$ to its terminal node t.
- The minimization (maximization) of the connectivity map

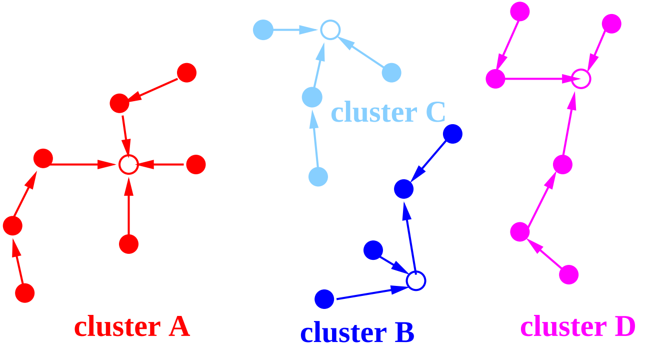
$$V(s) = \min_{\forall t \in \Pi(\mathcal{T}, \mathcal{A}, t)} \{f(\pi_t)\}$$

  produces an optimum-path forest rooted at nodes called prototypes.

In supervised learning, each class is an optimum-path forest rooted at its prototypes, which propagate the class label to the remaining nodes of the forest.



In unsupervised learning, each cluster is an optimum-path tree rooted at some prototype, which propagates a cluster label to the remaining nodes of the tree.



- This methodology does not assume known shapes, non-overlapping classes, or parametric models.

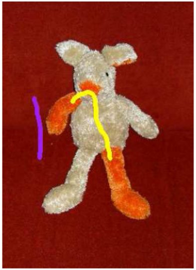
Introduction

- This methodology does not assume known shapes, non-overlapping classes, or parametric models.
- Both learning approaches are fast and robust for training sets of reasonable sizes.

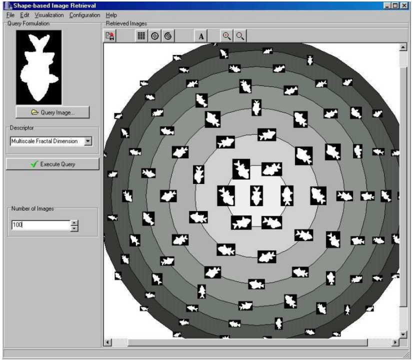
Introduction

- This methodology does not assume known shapes, non-overlapping classes, or parametric models.
- Both learning approaches are fast and robust for training sets of reasonable sizes.
- Label propagation to new samples $t \in \mathcal{Z} \backslash \mathcal{T}$ is efficiently performed based on a local processing of the forest's attributes and distances between nodes $s \in \mathcal{T}$ and t.

Organization of this lecture

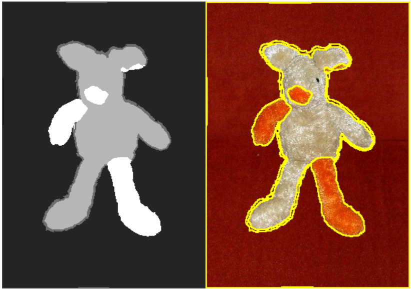


- Supervised classification by OPF [1].



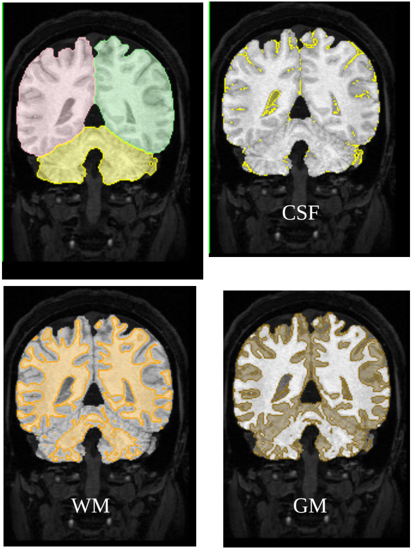
- Supervised classification by OPF [1].
- Its application to image retrieval [2].

Organization of this lecture

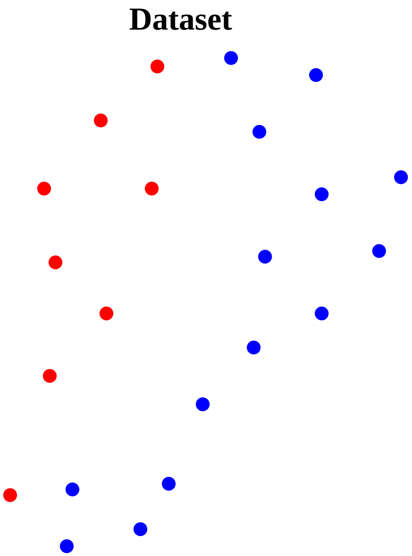


- Supervised classification by OPF [1].
- Its application to image retrieval [2].
- Clustering by OPF [3].

Organization of this lecture

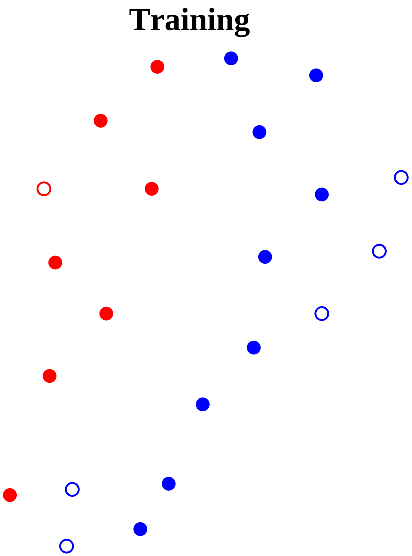


- Supervised classification by OPF [1].
- Its application to image retrieval [2].
- Clustering by OPF [3].
- Its application to 3D brain tissue segmentation [4].

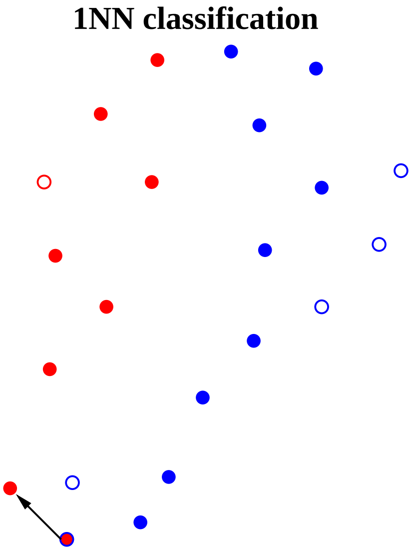


- Consider samples from two classes of a dataset.

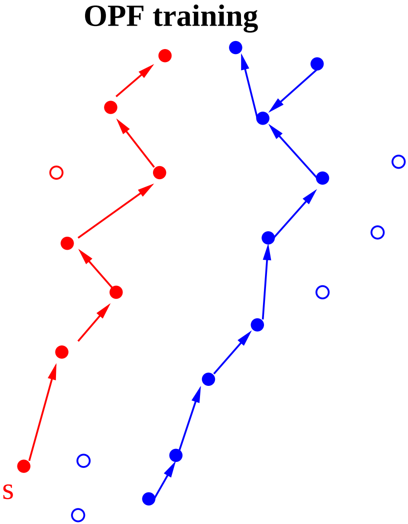
Supervised classification



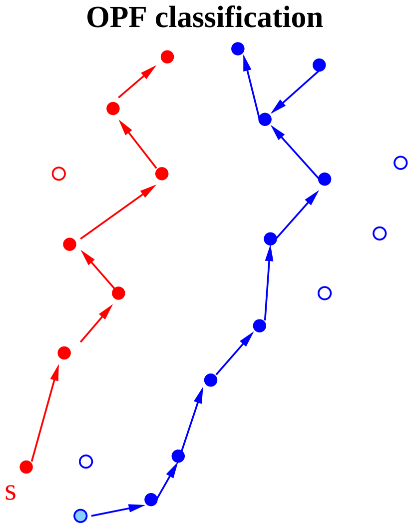
- Consider samples from two classes of a dataset.
- A training set (filled bullets) may not represent data distribution.



- Consider samples from two classes of a dataset.
- A training set (filled bullets) may not represent data distribution.
- Classification by nearest neighbor fails, when training samples are close to test samples (empty bullets) from other classes.



- We can create an optimum-path forest, where $V(s)$ is penalized when s is not closely connected to its class.



- We can create an optimum-path forest, where $V(s)$ is penalized when s is not closely connected to its class.
- $V(s)$ can then be used to reduce the power of s to classify new samples.

- We interpret $(\mathcal{T}, \mathcal{A})$ as a complete graph with undirected arcs between training samples.

Supervised learning

- We interpret $(\mathcal{T}, \mathcal{A})$ as a complete graph with undirected arcs between training samples.
- For a given set $\mathcal{S} \subset \mathcal{T}$ of prototypes from all classes, the connectivity map $V(t)$ is minimized for

$$
\begin{aligned}
f_{\max}(\langle t \rangle) &= \begin{cases} 0 & \text{if } t \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases} \\
f_{\max}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{\max}(\pi_s), d(s,t)\}
\end{aligned}
$$

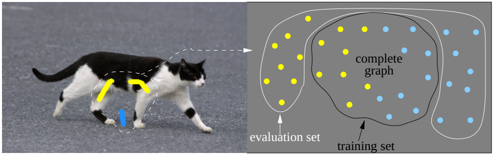
where $d(s,t)$ is the distance between s and t as computed by a descriptor.

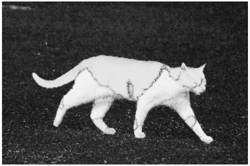
- We interpret $(\mathcal{T}, \mathcal{A})$ as a complete graph with undirected arcs between training samples.
- For a given set $\mathcal{S} \subset \mathcal{T}$ of prototypes from all classes, the connectivity map $V(t)$ is minimized for

$$
\begin{aligned}
f_{\max}(\langle t \rangle) &= \begin{cases} 0 & \text{if } t \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases} \\
f_{\max}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{\max}(\pi_s), d(s,t)\}
\end{aligned}
$$

  where $d(s, t)$ is the distance between s and t as computed by a descriptor.
- The prototypes are the closest samples between classes.

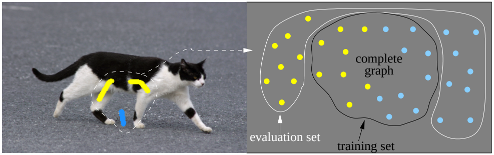
We used this idea to enhance objects in lecture 3 where $\mathcal{Z} = \mathcal{D}_I$.

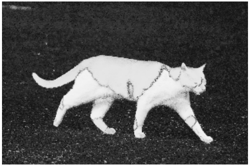




Supervised learning

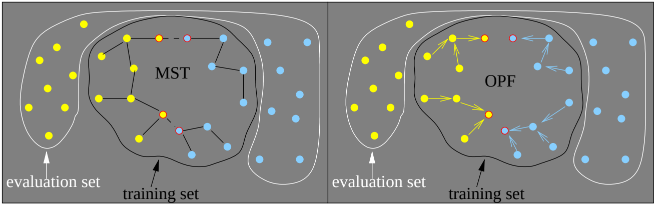
We used this idea to enhance objects in lecture 3 where $\mathcal{Z} = \mathcal{D}_I$.



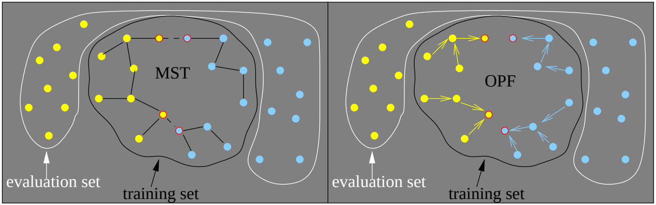


Even marker nodes may constitute large labeled sets, but they can be divided into a smaller training set $\mathcal{T}$ and a larger evaluation set $\mathcal{E}$ such that the most representative samples for $\mathcal{T}$ can be learned from $\mathcal{E}$.

Supervised learning

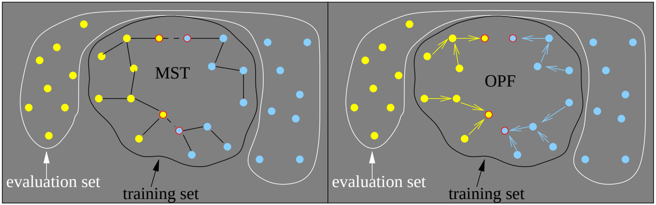


Supervised learning



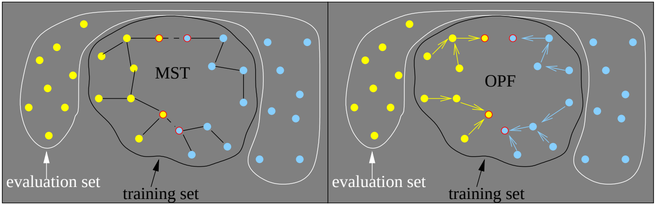
- A minimum spanning tree is computed in $(\mathcal{T}, \mathcal{A})$ and nodes that share arcs between distinct classes are taken as prototypes in $\mathcal{S}$.

Supervised learning



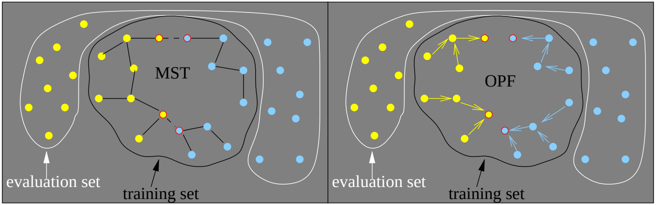
- A minimum spanning tree is computed in $(\mathcal{T}, \mathcal{A})$ and nodes that share arcs between distinct classes are taken as prototypes in $\mathcal{S}$.
- Object and background are then represented by optimum-path forests rooted in $\mathcal{S}$ (i.e., a pixel classifier).

Supervised learning



- Prototypes compete among themselves and nodes in the evaluation set $\mathcal{E}$ are classified in the tree whose prototype offers an optimum path to it.

Supervised learning



- Prototypes compete among themselves and nodes in the evaluation set $\mathcal{E}$ are classified in the tree whose prototype offers an optimum path to it.
- Misclassified nodes in $\mathcal{E}$ are replaced by non-prototypes in $\mathcal{T}$ and the whole process is repeated for a few iterations in order to select the most representative nodes for $\mathcal{T}$.

Classification

- For any $t \in \mathcal{Z} \backslash \mathcal{T}$,

$$V(t) \quad = \quad \min_{\forall s \in \mathcal{T}} \{\max\{V(s), d(s,t)\}\}.$$

Classification

- For any $t \in \mathcal{Z} \backslash \mathcal{T}$,

$$V(t) \quad = \quad \min_{\forall s \in \mathcal{T}} \{\max\{V(s), d(s,t)\}\}.$$

- Let $s^* \in \mathcal{T}$ be the node that satisfies this equation, then the class of t is assumed to be $L(s^*)$.

- For any $t \in \mathcal{Z} \backslash \mathcal{T}$,

$$V(t) \quad = \quad \min_{\forall s \in \mathcal{T}} \{\max\{V(s), d(s,t)\}\}.$$

- Let $s^* \in \mathcal{T}$ be the node that satisfies this equation, then the class of t is assumed to be $L(s^*)$.
- Let $V_o(t)$ and $V_b(t)$ be the optimum values in the above equation for object and background forests, then a fuzzy object membership $\frac{V_b(t)}{V_o(t)+V_b(t)}$ can be assigned to every spel $t \in \mathcal{D}_I$.

Supervised OPF-training algorithm

Algorithm

– SUPERVISED TRAINING BY OPTIMUM-PATH FOREST

1. *For each* $t \in \mathcal{T} \backslash \mathcal{S}$*, set* $V(t) \leftarrow +\infty$.
2. *For each* $t \in \mathcal{S}$*, set* $L(t) \leftarrow \lambda(t)$, $V(t) \leftarrow 0$ *and insert* t *in* Q.
3. *While* Q *is not empty, do*
4. *Remove from* Q *a node* s *such that* $V(s)$ *is minimum.*
5. *Insert* s *in* $\mathcal{T}'$.
6. *For each* $t \in \mathcal{T}$ *such that* $V(t) > V(s)$*, do*
7. *Compute* $tmp \leftarrow \max\{V(s), d(s,t)\}$.
8. *If* $tmp < V(t)$*, then*
9. *If* $V(t) \neq +\infty$*, remove* t *from* Q.
10. *Set* $V(t) \leftarrow tmp$ *and* $L(t) \leftarrow L(s)$.
11. *Insert* t *in* Q.

Classification

The role of the ordered set $\mathcal{T}'$ is to speed up classification [5], which can halt when $\max\{V(s), d(s,t)\} < V(s')$ for a node s' whose position in $\mathcal{T}'$ succeeds the position of s, while evaluating

$$V(t) \quad = \quad \min_{\forall s \in \mathcal{T}'} \{\max\{V(s), d(s,t)\}\}.$$

Prototype estimation

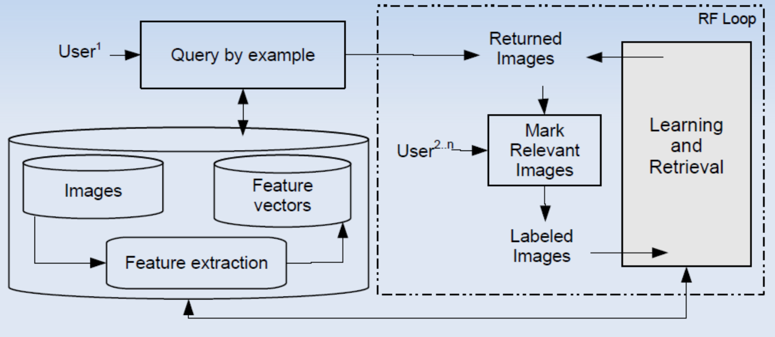
The minimum spanning tree can be obtained from the same algorithm by

- using a non-smooth function

$$
\begin{aligned}
f_{mst}(\langle t \rangle) &= \begin{cases} 0 & \text{for an arbitrary node } t \in \mathcal{T} \\ +\infty & \textit{otherwise}, \end{cases} \\
f_{mst}(\pi_s \cdot \langle s, t \rangle) &= w(s, t),
\end{aligned}
$$

- and replacing $V(t) > V(s)$ in Line 6 by $V(t) = +\infty$ or $t \in Q$.

The OPF classifier has provided effective and efficient image retrieval from a few iterations of relevance feedback.



Application to Image Retrieval

In each iteration of relevance feedback,

- the relevant and irrelevant images are the nodes of a complete graph $(\mathcal{T}, \mathcal{A})$.

In each iteration of relevance feedback,

- the relevant and irrelevant images are the nodes of a complete graph $(\mathcal{T}, \mathcal{A})$.
- An OPF classifier is projected and used to select relevant candidates from the image database $\mathcal{Z}$.

In each iteration of relevance feedback,

- the relevant and irrelevant images are the nodes of a complete graph $(\mathcal{T}, \mathcal{A})$.
- An OPF classifier is projected and used to select relevant candidates from the image database $\mathcal{Z}$.
- The relevant candidates are ordered based on their average distances to the relevant prototypes.

For a query image using the Corel database and the BIC image descritor [6].

Application to Image Retrieval

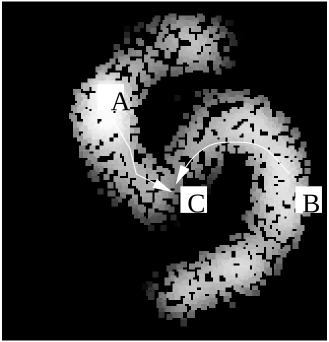
First iteration only returns the 30 closest images to the query one.

Application to Image Retrieval

After three iterations, the 30 most relevant images are.

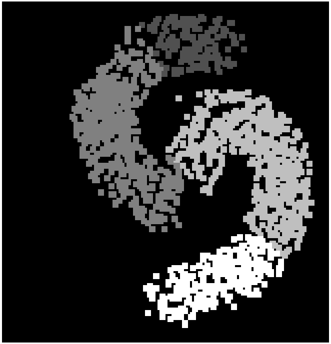
Clustering

For unsupervised learning, we estimate a probability density function (pdf) and the maxima of the pdf compete with each other, such that each cluster will be an optimum-path tree rooted at one maximum of the pdf.



Clustering

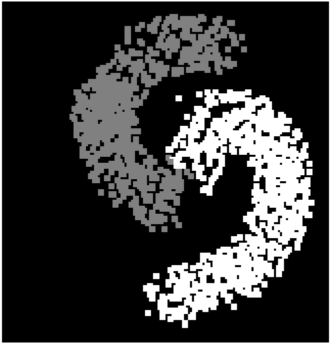
For unsupervised learning, we estimate a probability density function (pdf) and the maxima of the pdf compete with each other, such that each cluster will be an optimum-path tree rooted at one maximum of the pdf.



It is also possible to eliminate clusters of irrelevant maxima by choice of the connectivity function.

Clustering

For unsupervised learning, we estimate a probability density function (pdf) and the maxima of the pdf compete with each other, such that each cluster will be an optimum-path tree rooted at one maximum of the pdf.



It is also possible to eliminate clusters of irrelevant maxima by choice of the connectivity function.

Clustering

The unlabeled training samples form a knn-graph $(\mathcal{T}, \mathcal{A}_k)$ with adjacency relation

$$\mathcal{A}_k \quad : \quad (s,t) \in \mathcal{A}_k \text{ (or } t \in \mathcal{A}_k(s)) \text{ if } t \text{ is } k \text{ nearest neighbor of } s \text{ using the distance space.}$$

The best value of k is the one whose clustering produces a minimum normalized graph cut in $(\mathcal{T}, \mathcal{A}_k)$.

Clustering

The graph is weighted on the arcs $(s,t) \in \mathcal{A}_k$ by $d(s,t)$ and on the nodes by the pdf $\rho(s)$.

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|\mathcal{A}_k(s)|} \sum_{\forall t \in \mathcal{A}_k(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right)$$

where $\sigma = \frac{d_f}{3}$ and $d_f = \max_{\forall (s,t) \in \mathcal{A}_k}\{d(s,t)\}$. The pdf is usually normalized within an interval $[1, K]$.

Clustering

The connectivity map $V(t)$ is maximized for

$$
\begin{aligned}
f_{\min}(\langle t\rangle) &= \begin{cases} \rho(t) & \text{if } t \in \mathcal{R} \\ \rho(t)-1 & \text{otherwise} \end{cases} \\
f_{\min}(\pi_s \cdot \langle s,t\rangle) &= \min\{f_{\min}(\pi_s), \rho(t)\}
\end{aligned}
$$

where $\mathcal{R}$ is the root set found on-the-fly and arcs are added in $\mathcal{A}_k$ to guarantee arc symmetry on the plateaus of the pdf.

Algorithm

– CLUSTERING BY OPTIMUM PATH FOREST

1. Set $lb \leftarrow 1$.
2. For each $s \in \mathcal{T}$, set $V(s) \leftarrow \rho(s) - 1$ and insert s in Q.
3. While Q is not empty, do
4. Remove from Q a sample s such that $V(s)$ is maximum
5. Insert s in $\mathcal{T}'$.
6. If $P(s) = nil$, then
7. Set $L(s) \leftarrow lb$, $lb \leftarrow lb + 1$, and $V(s) \leftarrow \rho(s)$.
8. For each $t \in \mathcal{A}_k(s)$ and $V(t) < V(s)$, do
9. Compute $tmp \leftarrow \min\{V(s), \rho(t)\}$.
10. If $tmp > V(t)$ then
11. Set $L(t) \leftarrow L(s)$ and $V(t) \leftarrow tmp$.
12. Update position of t in Q.

Label propagation

The role of the ordered set $\mathcal{T}'$ is to speed up label propagation to new nodes $t \in \mathcal{Z} \backslash \mathcal{T}$ [4], which can halt when s^* is found in

$$V(s^*) = \max_{\forall s \in \mathcal{T}' | d(s,t) \leq \omega(s)} \{V(s)\},$$

where $\omega(s)$ is the maximum distance between s and its k-nearest neighbors in $\mathcal{T}$. The node t then receives label $L(s^*)$.

Application to brain tissue segmentation

After brain segmentation and bias correction.

Application to brain tissue segmentation

After brain segmentation and bias correction.

- The brain voxels are first classified into CSF or GM+WM and then classified into GM or WM, because the method requires different parameters (e.g., different features and $\mathcal{A}_k$) in each case.

Application to brain tissue segmentation

After brain segmentation and bias correction.

- The brain voxels are first classified into CSF or GM+WM and then classified into GM or WM, because the method requires different parameters (e.g., different features and $\mathcal{A}_k$) in each case.
- Let $\mathcal{Z}$ be a set of brain voxels from two classes.

Application to brain tissue segmentation

After brain segmentation and bias correction.

- The brain voxels are first classified into CSF or GM+WM and then classified into GM or WM, because the method requires different parameters (e.g., different features and $\mathcal{A}_k$) in each case.
- Let $\mathcal{Z}$ be a set of brain voxels from two classes.
- A feature vector $\vec{v}(t)$ is assigned to every voxel $t \in \mathcal{Z}$ and $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$.

Application to brain tissue segmentation

After brain segmentation and bias correction.

- The brain voxels are first classified into CSF or GM+WM and then classified into GM or WM, because the method requires different parameters (e.g., different features and $\mathcal{A}_k$) in each case.
- Let $\mathcal{Z}$ be a set of brain voxels from two classes.
- A feature vector $\vec{v}(t)$ is assigned to every voxel $t \in \mathcal{Z}$ and $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$.
- A small training set $\mathcal{T} \subset \mathcal{Z}$ is obtained by random sampling.

Application to brain tissue segmentation

After brain segmentation and bias correction.

- The brain voxels are first classified into CSF or GM+WM and then classified into GM or WM, because the method requires different parameters (e.g., different features and $\mathcal{A}_k$) in each case.
- Let $\mathcal{Z}$ be a set of brain voxels from two classes.
- A feature vector $\vec{v}(t)$ is assigned to every voxel $t \in \mathcal{Z}$ and $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$.
- A small training set $\mathcal{T} \subset \mathcal{Z}$ is obtained by random sampling.
- The OPF clustering can find in $\mathcal{T}$ groups of voxels, mostly from a same class.

Application to brain tissue segmentation

After brain segmentation and bias correction.

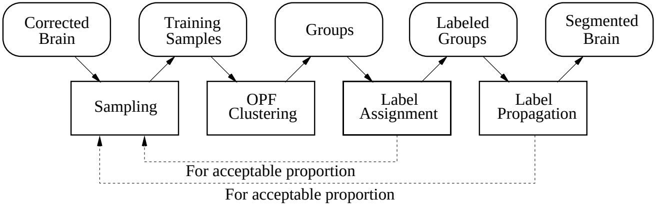
- The brain voxels are first classified into CSF or GM+WM and then classified into GM or WM, because the method requires different parameters (e.g., different features and $\mathcal{A}_k$) in each case.
- Let $\mathcal{Z}$ be a set of brain voxels from two classes.
- A feature vector $\vec{v}(t)$ is assigned to every voxel $t \in \mathcal{Z}$ and $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$.
- A small training set $\mathcal{T} \subset \mathcal{Z}$ is obtained by random sampling.
- The OPF clustering can find in $\mathcal{T}$ groups of voxels, mostly from a same class.
- Class labels are assigned to each group and propagated to the remaining voxels in $\mathcal{Z}$.

Application to brain tissue segmentation

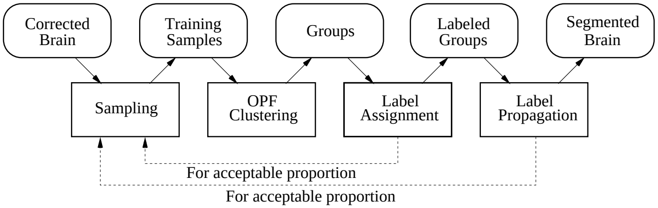
After brain segmentation and bias correction.

- The brain voxels are first classified into CSF or GM+WM and then classified into GM or WM, because the method requires different parameters (e.g., different features and $\mathcal{A}_k$) in each case.
- Let $\mathcal{Z}$ be a set of brain voxels from two classes.
- A feature vector $\vec{v}(t)$ is assigned to every voxel $t \in \mathcal{Z}$ and $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$.
- A small training set $\mathcal{T} \subset \mathcal{Z}$ is obtained by random sampling.
- The OPF clustering can find in $\mathcal{T}$ groups of voxels, mostly from a same class.
- Class labels are assigned to each group and propagated to the remaining voxels in $\mathcal{Z}$.
- The process may be repeated until it achieves an acceptable result.

Brain tissue segmentation

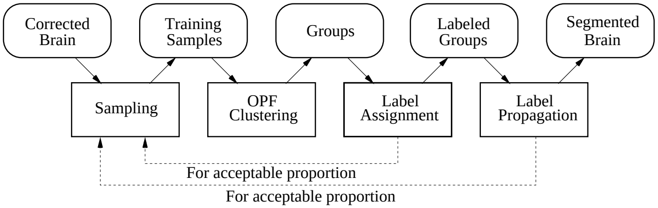


Brain tissue segmentation



- For MRT1-images, group labeling is done from the darkest to the brightest cluster until the size proportion p between the classes is the closest to a previously estimated value p_T, which is obtained by automatic thresholding.

Brain tissue segmentation



- For MRT1-images, group labeling is done from the darkest to the brightest cluster until the size proportion p between the classes is the closest to a previously estimated value p_T, which is obtained by automatic thresholding.
- The acceptance criterion requires that $p \in [p_T - \delta, p_T + \delta]$, whose value of δ increases at every m sampling attempts.

Conclusion

- We presented the design of fast and effective clustering and classification methods based on optimum-path forest.

Conclusion

- We presented the design of fast and effective clustering and classification methods based on optimum-path forest.
- These methods have been succeeded not only in image retrieval [2] and medical imaging [4], but also in several other applications.

Conclusion

- We presented the design of fast and effective clustering and classification methods based on optimum-path forest.
- These methods have been succeeded not only in image retrieval [2] and medical imaging [4], but also in several other applications.
- Their C source code is available in `www.ic.unicamp.br/~afalcao/libopf`.

Acknowledgments

- FAPESP.
- The organizing committee.
- My students and other collaborators.
- You all for the participation.

[1] J.P. Papa, A.X. Falcão, and C.T.N. Suzuki.

Supervised pattern classification based on optimum-path forest.

Intl. Journal of Imaging Systems and Technology, 19(2):120–131, Jun 2009.

[2] A.T. Silva, A.X. Falcão, and L.P. Magalhães.

A new CBIR approach based on relevance feedback and optimum-path forest classification.

Journal of WSCG, 18(1-3):73–80, 2010.

[3] L.M. Rocha, F.A.M. Cappabianco, and A.X. Falcão.

Data clustering as an optimum-path forest problem with applications in image analysis.

Intl. Journal of Imaging Systems and Technology, 19(2):50–68, Jun 2009.

[4] Fábio A.M. Cappabianco, A.X. Falcão, Clarissa L. Yasuda, and J. K. Udupa.

MR-Image Segmentation of Brain Tissues based on Bias Correction and Optimum-Path Forest Clustering.

Technical Report IC-10-07, Institute of Computing, University of Campinas, March 2010.

[5] J. P. Papa, F. A. M. Cappabianco, and A. X. Falcão.
Optimizing optimum-path forest classification for huge datasets.
In *Proceedings of The 20th International Conference on Pattern Recognition*, Istanbul, Turkey, Aug 2010.
(to appear).

[6] R. O. Stehling, M. A. Nascimento, and A. X. Falcao.
A compact and efficient image retrieval approach based on border/interior pixel classification.
In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 102–109, New York, NY, USA, 2002. ACM.