Learning to Classify Seismic Images with Deep Optimum-Path Forest

Luis Afonso[1]
Michelle Kuroda[2], Alexandre Vidal[2]
Alexandre Falcão[3]
João Papa[4]

[1]Department of Computing, UFSCar-Brazil
[2]Institute of Geosciences, UNICAMP-Brazil
[3]Institute of Computing, UNICAMP-Brazil
[4]Department of Computing, UNESP/Bauru-Brazil

Outline

- Introduction
- Optimum-Path Forest
- Learning Deep Representations
- Methodology
- Experimental Results
- Conclusions

Introduction

- Image classification plays important role in wide range of applications:
  - Remote sensing-driven tools.

Introduction

- Image classification plays important role in wide range of applications:
  - Remote sensing-driven tools.
  - Medical image analysis.

Introduction

- Issues:
  - Lack of labeled data.

Introduction

- Issues:
  - Lack of labeled data.
  - Huge amount of images.

Introduction

- Issues:
  - Lack of labeled data.
  - Huge amount of images.
- Active learning-based techniques:

Introduction

- Issues:
  - Lack of labeled data.
  - Huge amount of images.
- Active learning-based techniques:
  - Also requires human interaction.

Introduction

- Issues:
  - Lack of labeled data.
  - Huge amount of images.
- Active learning-based techniques:
  - Also requires human interaction.
- Deep learning: Unsupervised learning.

Introduction

- A few existing solutions:
  - K-means

Introduction

- A few existing solutions:
  - K-means
  - Mean-shift

Introduction

- A few existing solutions:
  - K-means
  - Mean-shift
  - Self-Organizing Map

Introduction

- A few existing solutions:
  - K-means
  - Mean-shift
  - Self-Organizing Map
- Issue:

Introduction

- A few existing solutions:
  - K-means
  - Mean-shift
  - Self-Organizing Map
- Issue:
  - Number of clusters

Introduction

- A few existing solutions:
  - K-means
  - Mean-shift
  - Self-Organizing Map
- Issue:
  - Number of clusters
  - Get trapped in local optima

Introduction

- Graph-based clustering algorithms - Optimum Path Forest:
    - Samples are represented by nodes of a graph.

Introduction

- Graph-based clustering algorithms - Optimum Path Forest:
  - Samples are represented by nodes of a graph.
  - Competition-based learning process.

Introduction

- Graph-based clustering algorithms - Optimum Path Forest:
    - Samples are represented by nodes of a graph.
    - Competition-based learning process.
    - Application in bag-of-visual words.

Introduction

- Graph-based clustering algorithms - Optimum Path Forest:
  - Samples are represented by nodes of a graph.
  - Competition-based learning process.
  - Application in bag-of-visual words.
  - Similar centroids when compared to K-means.

Introduction

- Graph-based clustering algorithms - Optimum Path Forest:
  - Issue: "what if the value of k is known?"

Introduction

- Graph-based clustering algorithms - Optimum Path Forest:
  - Issue: "what if the value of k is known?"
  - k_{max} parameter.

Introduction

- Graph-based clustering algorithms - Optimum Path Forest:
  - Issue: "what if the value of k is known?"
  - k_{max} parameter.
  - High cost to find optimum value of k.

Introduction

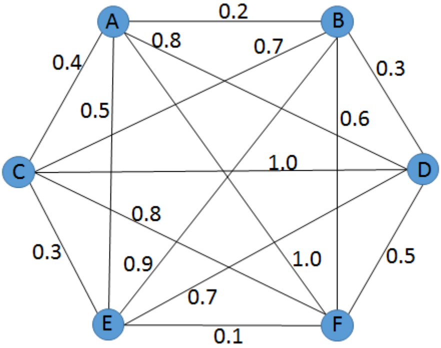
- Graph-based clustering algorithms - Optimum Path Forest:
    - **Deep-driven approach to reach a number of clusters close to the desired**.

Introduction

- Graph-based clustering algorithms - Optimum Path Forest:
  - **Deep-driven approach to reach a number of clusters close to the desired**.
  - Unsupervised learning applied in different views of the data.

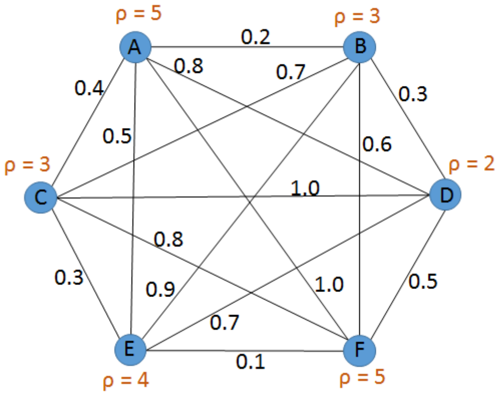
Optimum-Path Forest

- Nodes are connected by edges weighted by their distance (e.g., Euclidean distance).

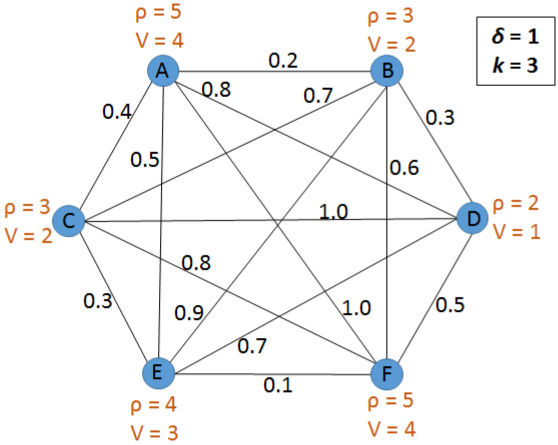


Optimum-Path Forest

- Nodes are weighted by a probability density function ρ.

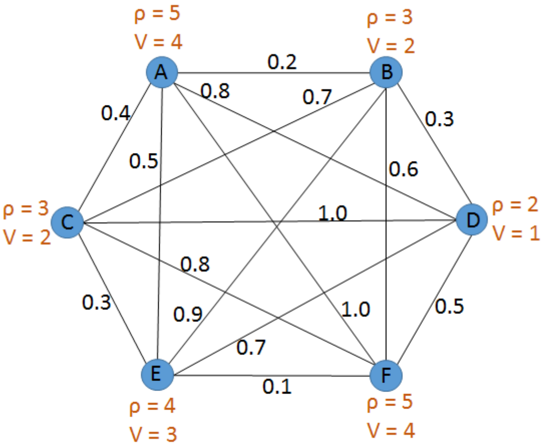


Optimum-Path Forest



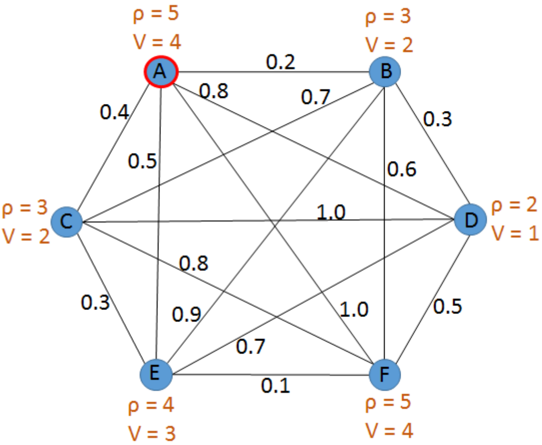
Optimum-Path Forest

- Create a priority queue Q ordered by the value of V.



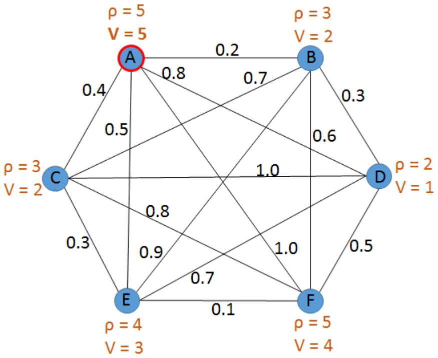
Optimum-Path Forest

- Remove from Q the node with the highest value.



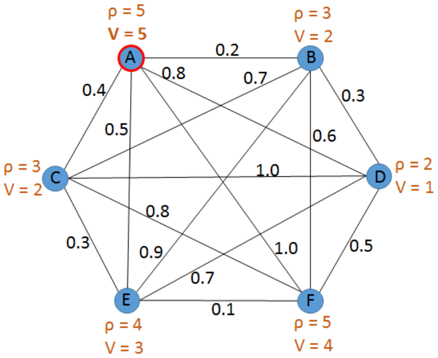
Optimum-Path Forest

- Check if node's predecessor is null. Update V and create new label l, if so.

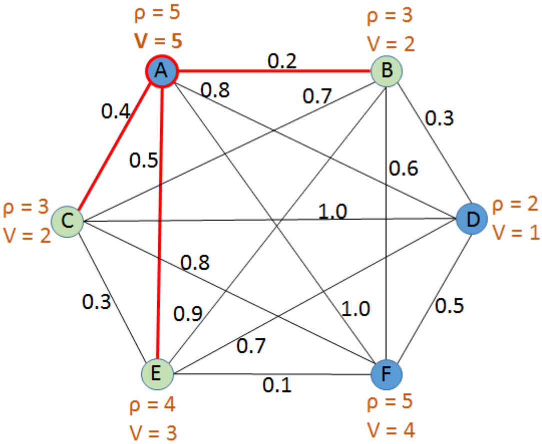


Optimum-Path Forest

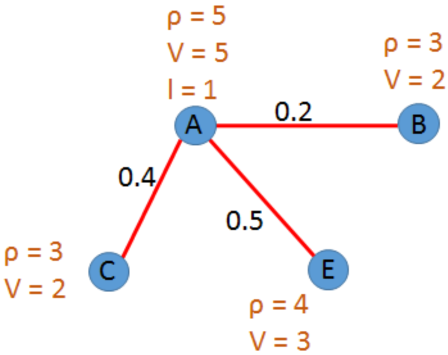
- Check if node's predecessor is null. Update V and create new label l, if so.
- Node becomes a prototype.



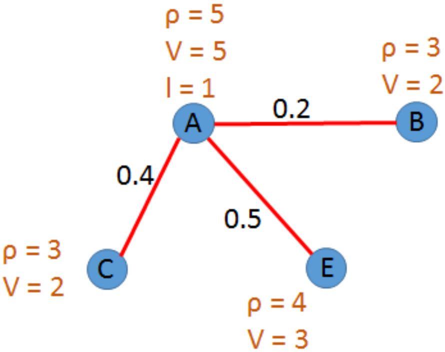
Optimum-Path Forest



Optimum-Path Forest



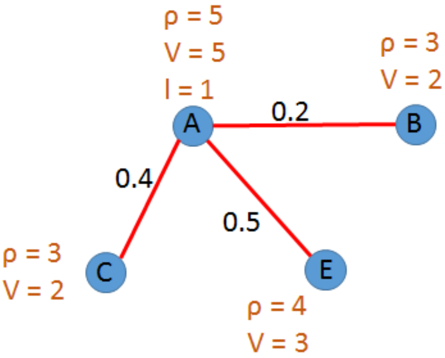
Optimum-Path Forest



A – B

tmp = min{V(A), ρ(B)}
tmp = min{5, 3}

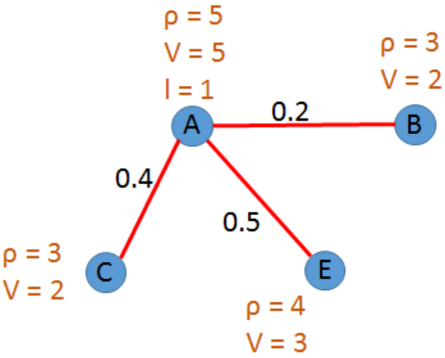
Optimum-Path Forest



A – B

tmp = min{V(A), ρ(B)}
tmp = min{5, 3}
tmp = 3

Optimum-Path Forest

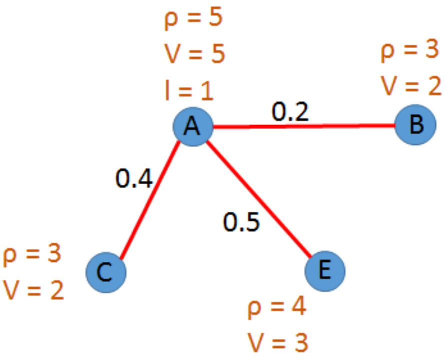


A – B

tmp = min{V(A), ρ(B)}
tmp = min{5, 3}
tmp = 3

tmp > V(B)?

Optimum-Path Forest

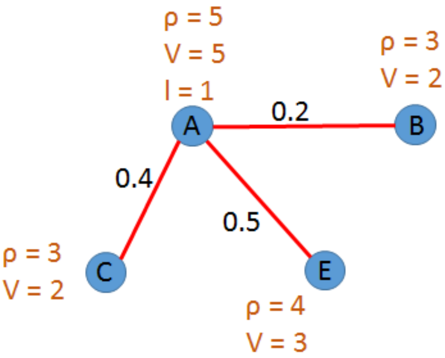


A – B

tmp = min{V(A), ρ(B)}
tmp = min{5, 3}
tmp = 3

tmp > V(B)?
3 > 2 ?

Optimum-Path Forest



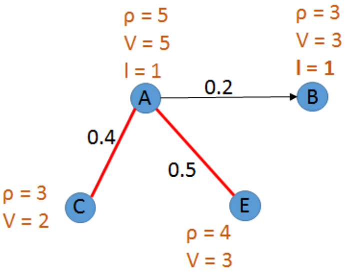
A – B

tmp = min{V(A), ρ(B)}
tmp = min{5, 3}
tmp = 3

tmp > V(B)?
3 > 2 ?
Yes!

Optimum-Path Forest

- $V(B) \leftarrow tmp$



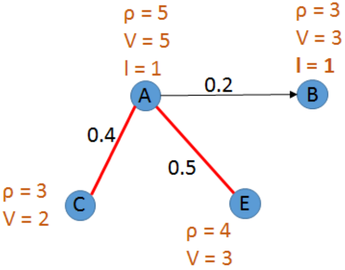
A – B

tmp = min{V(A), ρ(B)}
tmp = min{5, 3}
tmp = 3

tmp > V(B)?
3 > 2 ?
Yes!

Optimum-Path Forest

- $V(B) \leftarrow tmp$
- $I(B) \leftarrow I(A)$

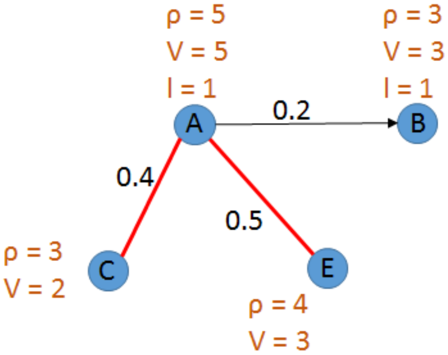


A – B

tmp = min{V(A), ρ(B)}
tmp = min{5, 3}
tmp = 3

tmp > V(B)?
3 > 2 ?
Yes!

Optimum-Path Forest

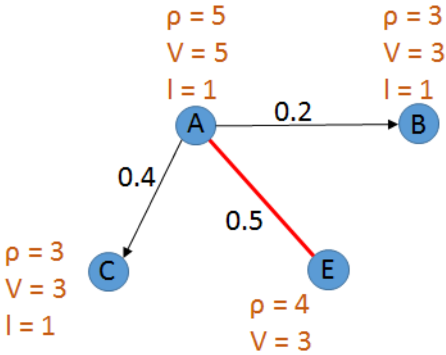


A – C

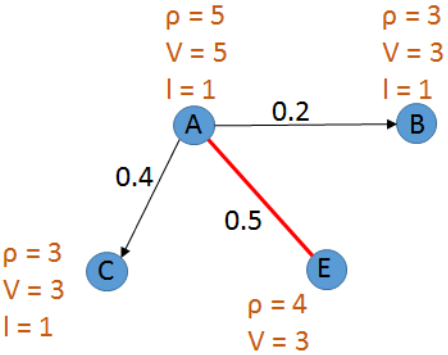
tmp = min{V(A), ρ(C)}
tmp = min{5, 3}
tmp = 3

tmp > V(C) ?
3 > 2 ? Yes

Optimum-Path Forest



Optimum-Path Forest

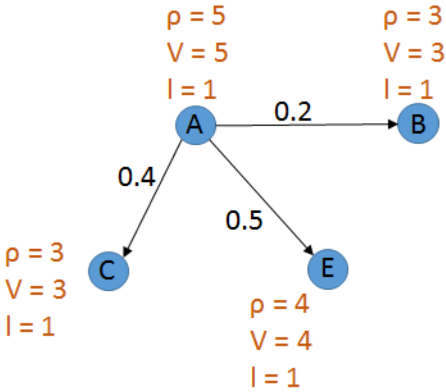


A – E

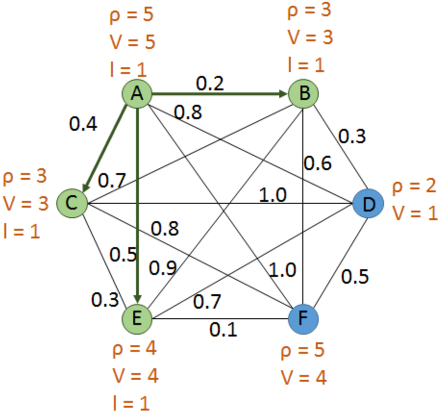
tmp = min{V(A), ρ(4)}
tmp = min{5, 4}
tmp = 4

tmp > V(E) ?
4 > 3 ? Yes

Optimum-Path Forest

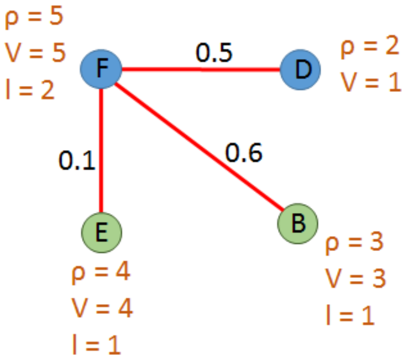


Optimum-Path Forest



F → E → B → C → D

Optimum-Path Forest

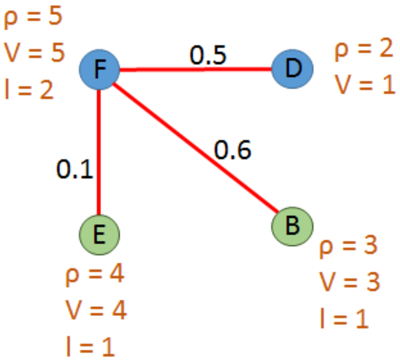


F – E

tmp = min{V(F), ρ(E)}
tmp = min{5, 4}
tmp = 4

tmp > V(E) ?
4 > 4 ? No

Optimum-Path Forest

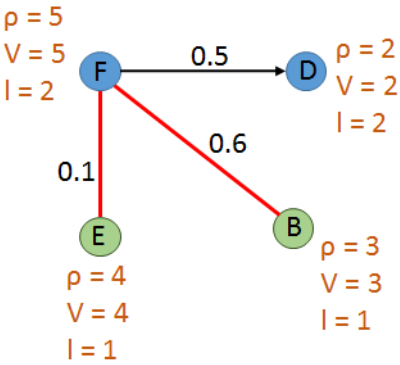


F – D

tmp = min{V(F), ρ(D)}
tmp = min{5, 2}
tmp = 2

tmp > V(D) ?
2 > 1 ? Yes

Optimum-Path Forest

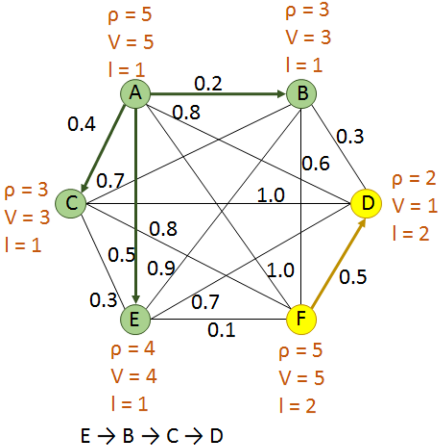


F – B

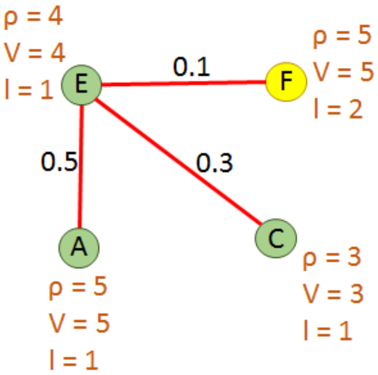
tmp = min{V(F), ρ(B)}
tmp = min{5, 3}
tmp = 3

tmp > V(B) ?
3 > 3 ? Yes

Optimum-Path Forest



Optimum-Path Forest

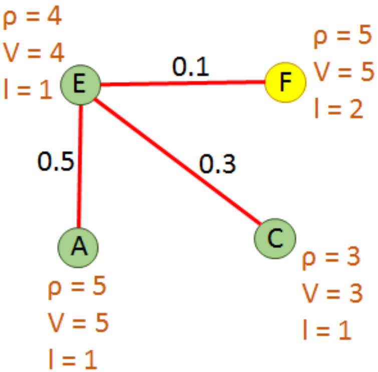


E – F

tmp = min{V(E), ρ(F)}
tmp = min{4, 5}
tmp = 4

tmp > V(F) ?
4 > 5 ? No

Optimum-Path Forest

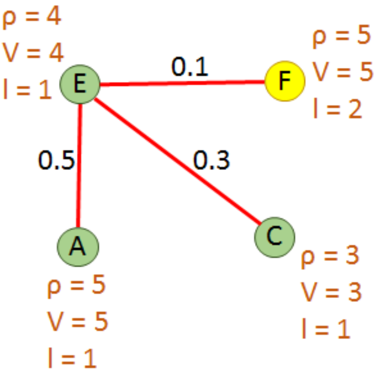


E – C

tmp = min{V(E), ρ(C)}
tmp = min{4, 3}
tmp = 4

tmp > V(C) ?
3 > 3 ? No

Optimum-Path Forest

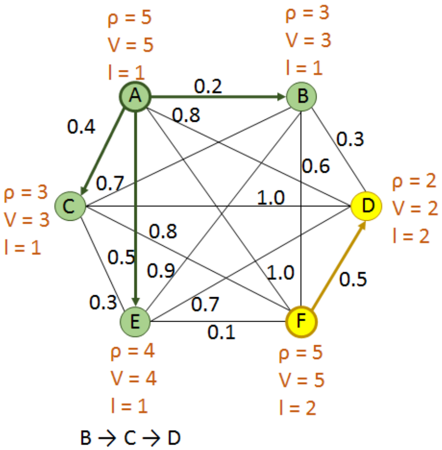


E – A

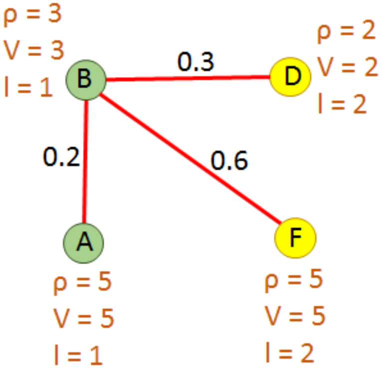
tmp = min{V(E), ρ(A)}
tmp = min{4, 5}
tmp = 4

tmp > V(A) ?
4 > 5 ? No

Optimum-Path Forest



Optimum-Path Forest

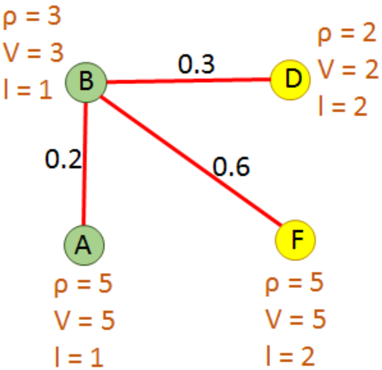


B – A

tmp = min{V(B), ρ(A)}
tmp = min{3, 5}
tmp = 3

tmp > V(A) ?
3 > 5 ? No

Optimum-Path Forest

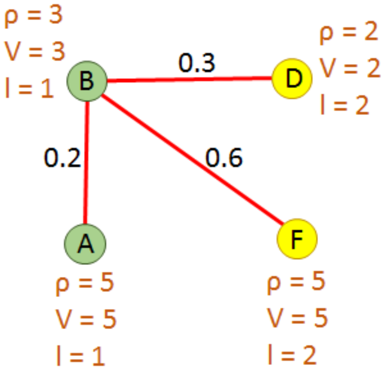


B – D

tmp = min{V(B), ρ(D)}
tmp = min{3, 2}
tmp = 2

tmp > V(D) ?
2 > 2 ? No

Optimum-Path Forest

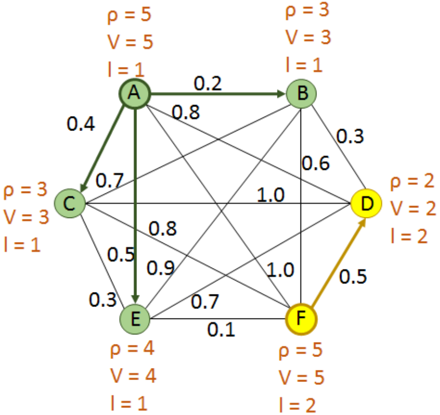


B – F

tmp = min{V(B), ρ(F)}
tmp = min{3, 5}
tmp = 3

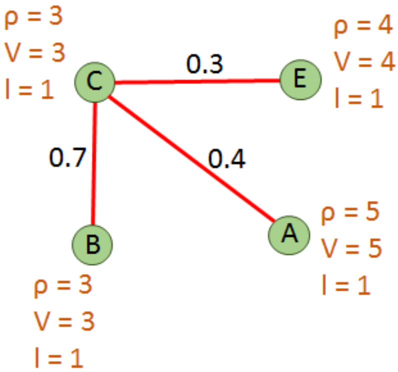
tmp > V(F) ?
3 > 5 ? No

Optimum-Path Forest



C → D

Optimum-Path Forest

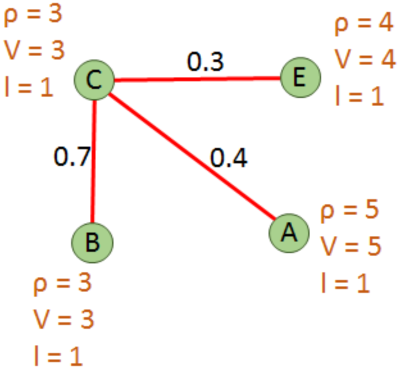


C - E

tmp = min{V(C), ρ(E)}
tmp = min{3, 4}
tmp = 3

tmp > V(E) ?
3 > 4 ? No

Optimum-Path Forest

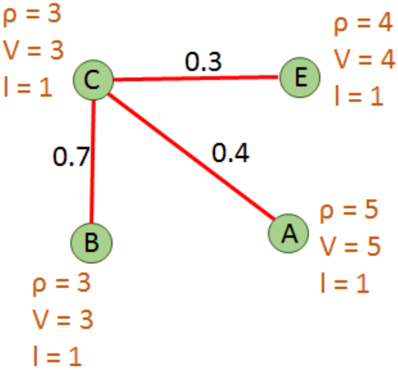


C - A

tmp = min{V(C), ρ(A)}
tmp = min{3, 5}
tmp = 3

tmp > V(A) ?
3 > 5 ? No

Optimum-Path Forest

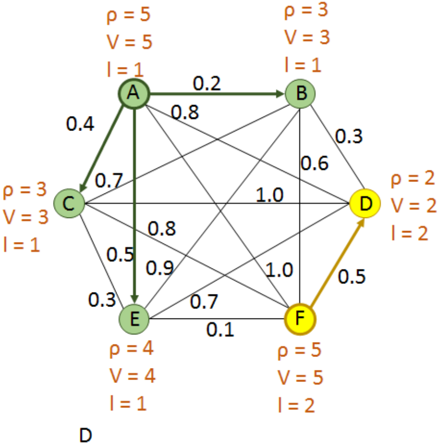


C - B

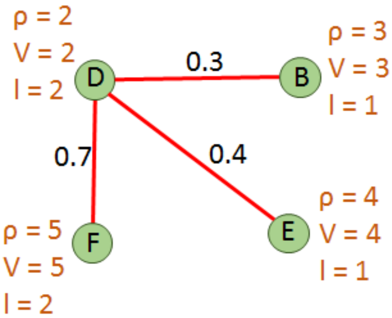
tmp = min{V(C), ρ(B)}
tmp = min{3, 3}
tmp = 3

tmp > V(B) ?
3 > 3 ? No

Optimum-Path Forest



Optimum-Path Forest

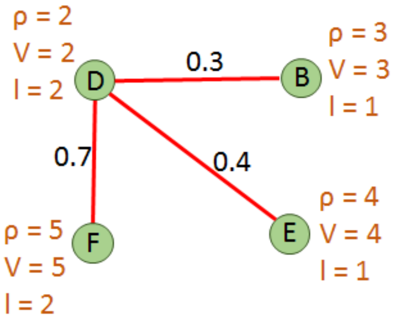


D - B

tmp = min{V(D), ρ(B)}
tmp = min{2, 3}
tmp = 2

tmp > V(B) ?
2 > 3 ? No

Optimum-Path Forest

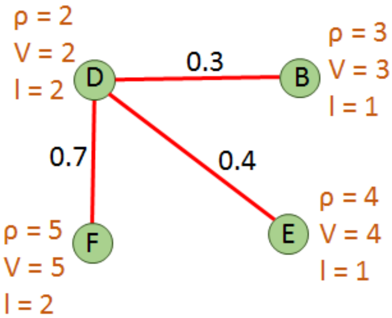


D - F

tmp = min{V(D), ρ(F)}
tmp = min{2, 5}
tmp = 2

tmp > V(F) ?
3 > 5 ? No

Optimum-Path Forest

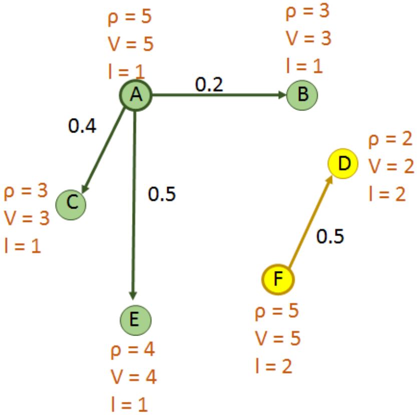


D - E

tmp = min{V(D), ρ(E)}
tmp = min{2, 4}
tmp = 2

tmp > V(E) ?
2 > 4 ? No

Optimum-Path Forest



Learning Deep Representations

- OPF does not require the number of clusters beforehand.

Learning Deep Representations

- OPF does not require the number of clusters beforehand.

- Playing with k_{max} can be prohibitive for large datasets.

Learning Deep Representations

- OPF does not require the number of clusters beforehand.
- Playing with k_{max} can be prohibitive for large datasets.
- Apply multiple clustering layers.

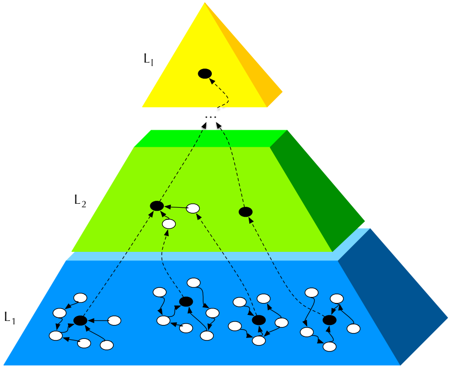
Learning Deep Representations

- OPF does not require the number of clusters beforehand.
- Playing with k_{max} can be prohibitive for large datasets.
- Apply multiple clustering layers.
- Prototypes of the first layer are the samples of the second layer, and so on.

Learning Deep Representations

- OPF does not require the number of clusters beforehand.
- Playing with k_{max} can be prohibitive for large datasets.
- Apply multiple clustering layers.
- Prototypes of the first layer are the samples of the second layer, and so on.
- Prototypes are located at regions of high density.

Learning Deep Representations



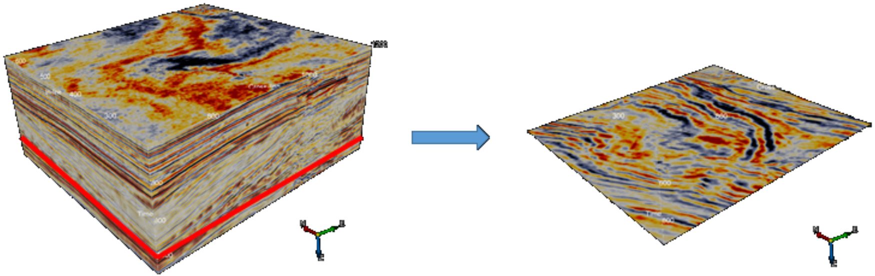
Methodology and Experimental Results

- Unlabeled dataset:
  - 3D seismic data

- Labeled dataset:
  - CIFAR10
  - CIFAR100
  - MNIST

Methodology and Experimental Results

- Seismic Images:

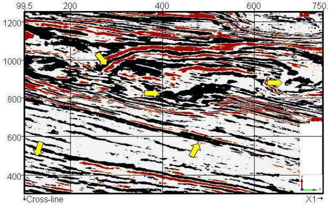


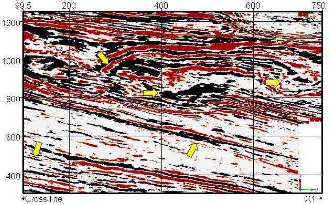
Methodology and Experimental Results

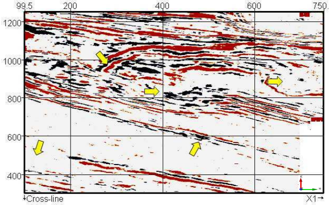
- Seismic Images:

	Layer			
Image	**1**	**2**	**3**	**4**
924	4,102	41	8	3
928	4,135	41	6	2
932	4,074	38	6	2
936	4,144	41	10	2
940	4,193	44	8	2

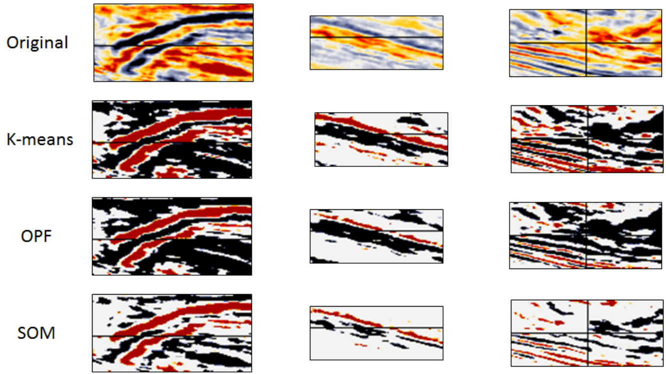
Methodology and Experimental Results





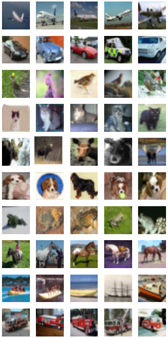


Methodology and Experimental Results



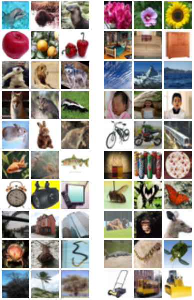
Methodology and Experimental Results

- General-purpose Images:
  - CIFAR-10: 60,000 images of size 32×32 distributed in 10 classes.



Methodology and Experimental Results

- General-purpose Images:
  - CIFAR-100: 60,000 images of size 32×32 distributed in 100 classes (finer) grouped in 20 superclasses (coarser).



Methodology and Experimental Results

- General-purpose Images:
  - MNIST: 70,000 images of handwritten digits distributed in 10 classes.

Methodology and Experimental Results

- General-purpose Images - Metrics:
  - Homogeneity (H): each cluster contains only members of a single class. $H \in [0, 1]$, where $H = 1$ denotes the best result.
  - Completeness (C): all members of a given class are assigned to the same cluster. $C \in [0, 1]$, where $C = 1$ denotes the best result.
  - V-measure (V): this metric is the harmonic mean between homogeneity and completeness, given by:

$$V = 2 * \frac{(H * C)}{(H + C)}. \tag{1}$$

Methodology and Experimental Results

- General-purpose Images:

	Layer			
Dataset	**1**	**2**	**3**	**4**
CIFAR 10	137	121	17	8
CIFAR 100	216	163	24	15
MNIST	221	145	5	2

Methodology and Experimental Results

- General-purpose Images - CIFAR10:

	Technique			
Metric	**OPF**	***k*-means**	**Mean-Shift**	**SOM**
H	0.000	**0.054**	0.001	0.049
C	**0.153**	0.060	0.039	0.056
V	0.000	**0.057**	0.001	0.052

Methodology and Experimental Results

- General-purpose Images - CIFAR100:

	Technique			
Metric	**OPF**	***k*-means**	**Mean-Shift**	**SOM**
H	0.010	**0.033**	0.001	0.030
C	0.069	0.038	**0.077**	0.034
V	0.017	**0.035**	0.003	0.032

Methodology and Experimental Results

- General-purpose Images - MNIST:

	Technique			
Metric	**OPF**	***k*-means**	**Mean-Shift**	**SOM**
H	0.000	0.007	0.000	**0.073**
C	**1.000**	0.024	0.005	0.376
V	0.000	0.011	0.001	**0.122**

Conclusions

- Deep-driven approach using OPF.
- OPF provided gain in resolution in seismic data.
- OPF was able to find number of clusters close to real number in 2 out of 3 datasets.
- Flexible tool for unsupervised learning.
- Techniques are complementary.

Thank you!

Q&A