IWBBIO 2022

Advanced Incremental Attribute Learning Clustering Algorithm for Medical and Healthcare Applications

Authors: Siwar GORRAB
Fahmi BEN REJAB
Kaouther NOUIRA

Doctoral student at BESTMOD laboratory, ISG Tunis, Université de Tunis, Tunisie.

January 28th, 2022

BESTMOD

Outline

Section 1

Introduction

Context

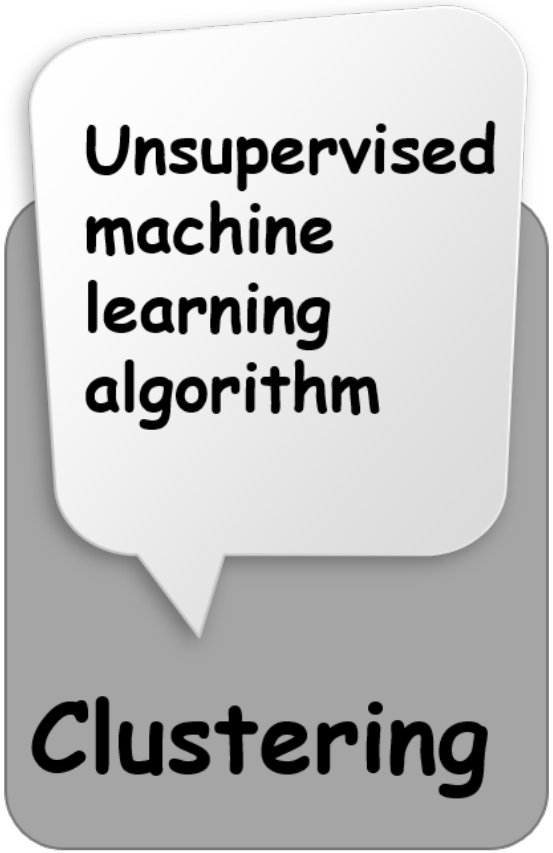
Context

Context

Collect the useful hidden information from these massive mixed data!

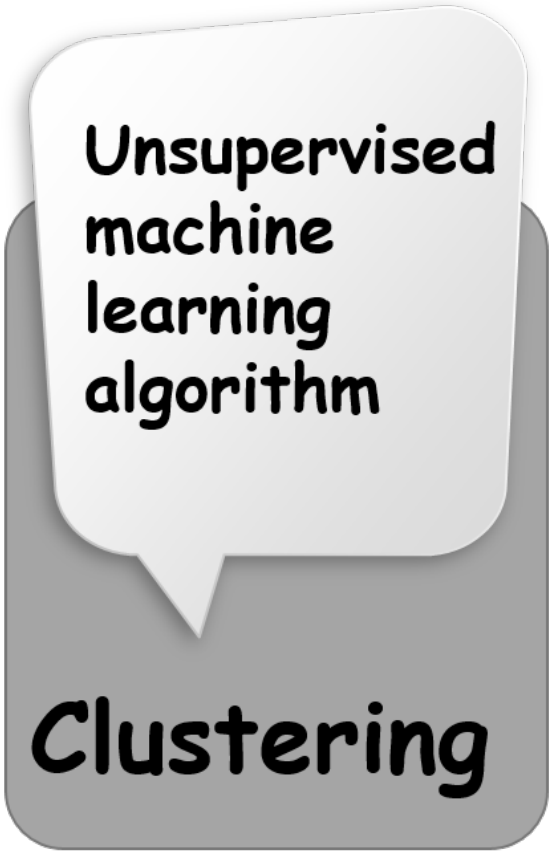
Context

Collect the useful hidden information from these massive mixed data!



Context

Collect the useful hidden information from these massive mixed data!



Context

Limits of batch clustering algorithms

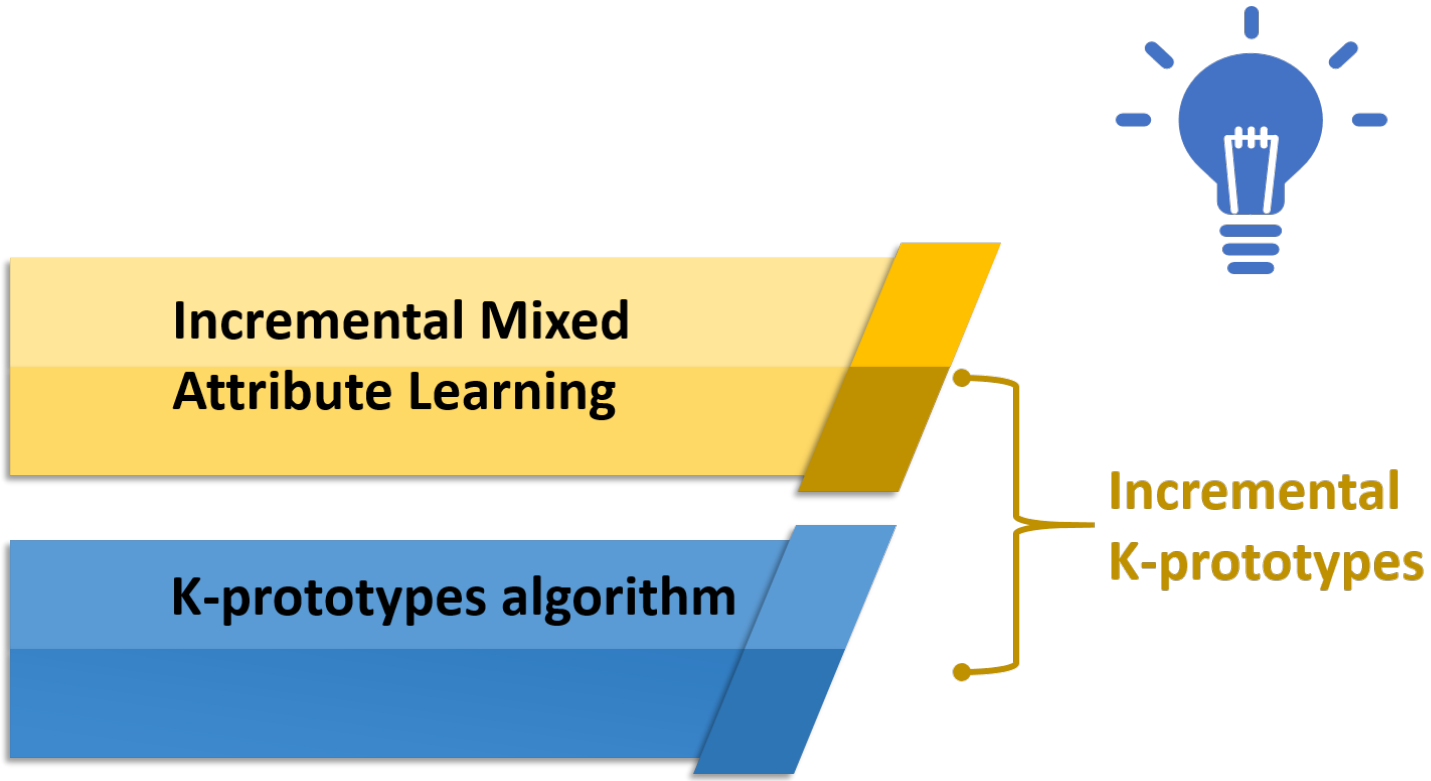
- Generate the best predictor by learning on the **all training data at once**.
- Need the **complete input data** being loaded into memory
  - The requirements of memory space will become **high**.
- Need to **regenerate** their clusters from scratch.
- **Complex** and **slow** analysis.

Context

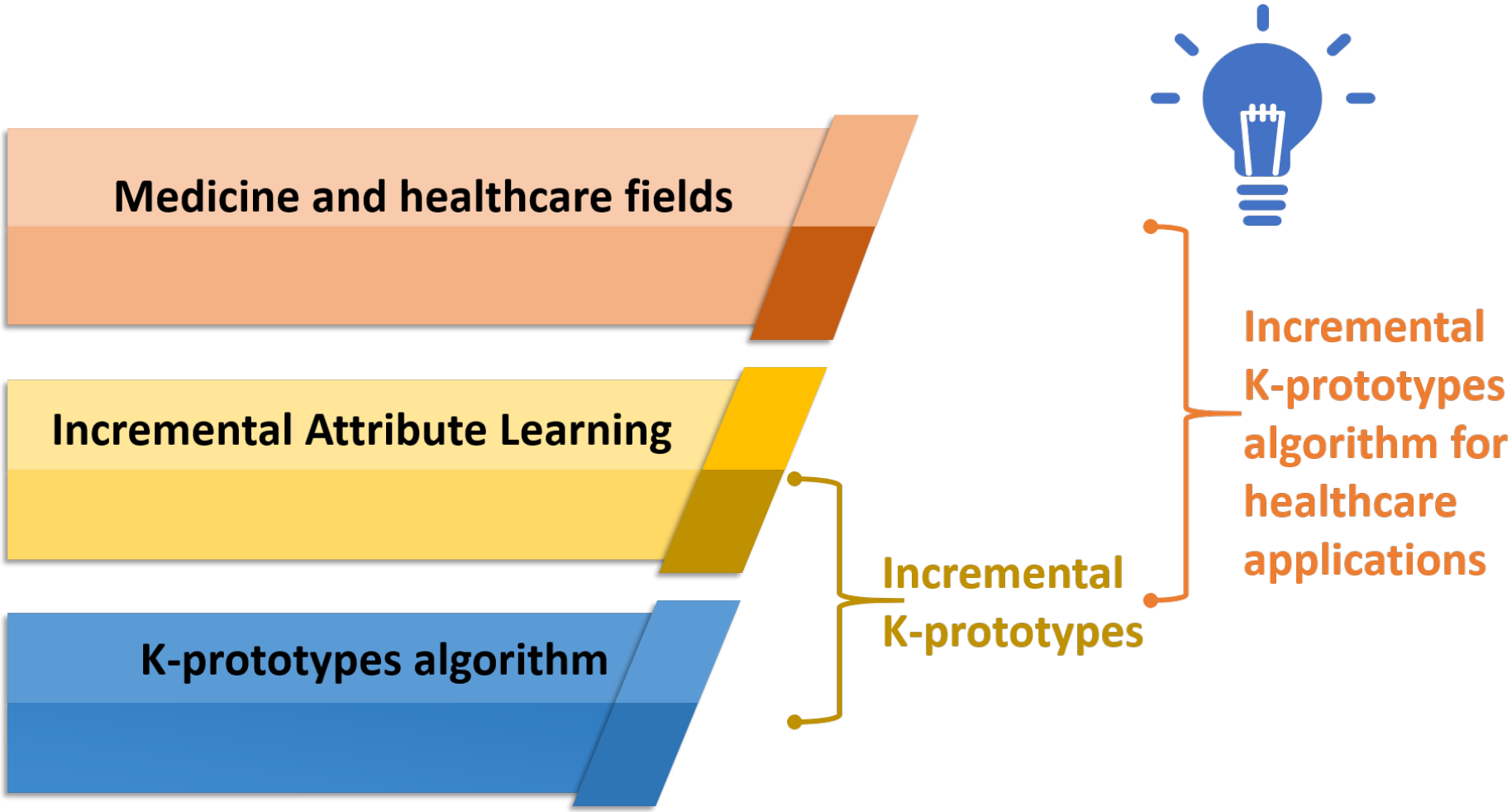
Incremental clustering

- ✓ Handle a bulk of **updates** owing to the training samples which become available one after another **over time**.
- ✓ Mixed data is processed **sequentially** over flexible time windows.

Objectives


Incremental Mixed
Attribute Learning
K-prototypes algorithm
Incremental
K-prototypes

Objectives


Medicine and healthcare fields
Incremental Attribute Learning
K-prototypes algorithm
Incremental K-prototypes
Incremental K-prototypes algorithm for healthcare applications

Section 2

Basic concepts

K-prototypes

Proposed by **Huang** in **1998** as an **extension to the k-means** algorithm to deal with **mixed data**

- Combinig the **k-modes** and **k-means** algorithms.
- Simplicity, scalability and speed of convergence.

✓ Its objective is to group the data set X into k clusters while minimizing the cost function

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} d(x_i - c_j), \tag{1}$$

K-prototypes

Proposed by **Huang** in **1998** as an **extension to the k-means** algorithm to deal with **mixed data**

- Combinig the **k-modes** and **k-means** algorithms.
- Simplicity, scalability and speed of convergence.

✓ Its objective is to group the data set X into k clusters while minimizing the cost function

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} d(x_i - c_j), \tag{1}$$

K-prototypes

Proposed by **Huang** in **1998** as an **extension to the k-means** algorithm to deal with **mixed data**

- Combinig the **k-modes** and **k-means** algorithms.
- Simplicity, scalability and speed of convergence.

✓ Its objective is to group the data set X into k clusters while minimizing the cost function

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} d(x_i - c_j), \tag{1}$$

K-prototypes

Proposed by **Huang** in **1998** as an **extension to the k-means** algorithm to deal with **mixed data**

- Combinig the **k-modes** and **k-means** algorithms.
- Simplicity, scalability and speed of convergence.

✓ Its objective is to group the data set X into k clusters while minimizing the cost function

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} d(x_i - c_j), \tag{1}$$

K-prototypes

- ✓ Provides a novel definition of distance (dissimilarity measure) between a data point and a cluster center

$$d(x_i - c_j) = \sum_{r=1}^{m_r} \sqrt{(x_{ir} - c_{jr})^2} + \sum_{t=1}^{m_t} \delta(x_{it}, c_{jt}), \tag{2}$$

- The first term is the **squared Euclidean distance** measure on the **numeric** attributes.
- The second term is the **simple matching dissimilarity** measure on the **categorical** attributes.

K-prototypes

- ✓ Provides a novel definition of distance (dissimilarity measure) between a data point and a cluster center

$$d(x_i - c_j) = \sum_{r=1}^{m_r} \sqrt{(x_{ir} - c_{jr})^2} + \sum_{t=1}^{m_t} \delta(x_{it}, c_{jt}), \tag{2}$$

- The first term is the **squared Euclidean distance** measure on the **numeric** attributes.
- The second term is the **simple matching dissimilarity** measure on the **categorical** attributes.

K-prototypes

✓ Provides a novel definition of distance (dissimilarity measure) between a data point and a cluster center

$$d(x_i - c_j) = \sum_{r=1}^{m_r} \sqrt{(x_{ir} - c_{jr})^2} + \sum_{t=1}^{m_t} \delta(x_{it}, c_{jt}), \tag{2}$$

- The first term is the **squared Euclidean distance** measure on the **numeric** attributes.
- The second term is the **simple matching dissimilarity** measure on the **categorical** attributes.

K-prototypes

✓ Provides a novel definition of distance (dissimilarity measure) between a data point and a cluster center

$$d(x_i - c_j) = \sum_{r=1}^{m_r} \sqrt{(x_{ir} - c_{jr})^2} + \sum_{t=1}^{m_t} \delta(x_{it}, c_{jt}), \tag{2}$$

- The first term is the **squared Euclidean distance** measure on the **numeric** attributes.
- The second term is the **simple matching dissimilarity** measure on the **categorical** attributes.

K-prototypes

Limitations

- **Retrains** from the scratch once new **data stream** emerges.
- Stores and processes **all the input data** in the **memory**.

  ↷ *High requirements of memory space!*

- Deals only with **object** learning **in batch**.
- **Inability** to handle **incremental attribute learning** task.

K-prototypes

Limitations

- **Retrains** from the scratch once new **data stream** emerges.
- Stores and processes **all the input data** in the **memory**.

↷ *High requirements of memory space!*

- Deals only with **object** learning **in batch**.
- **Inability** to handle **incremental attribute learning** task.

K-prototypes

Limitations

- **Retrains** from the scratch once new **data stream** emerges.
- Stores and processes **all the input data** in the **memory**.

↷ *High requirements of memory space!*

- Deals only with **object** learning **in batch**.
- **Inability** to handle **incremental attribute learning** task.

K-prototypes

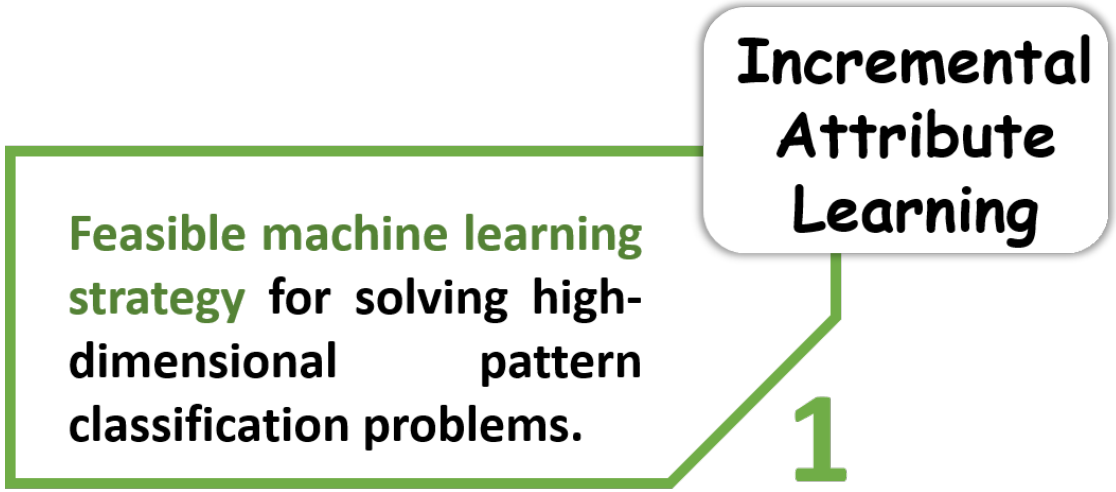
Limitations

- **Retrains** from the scratch once new **data stream** emerges.
- Stores and processes **all the input data** in the **memory**.

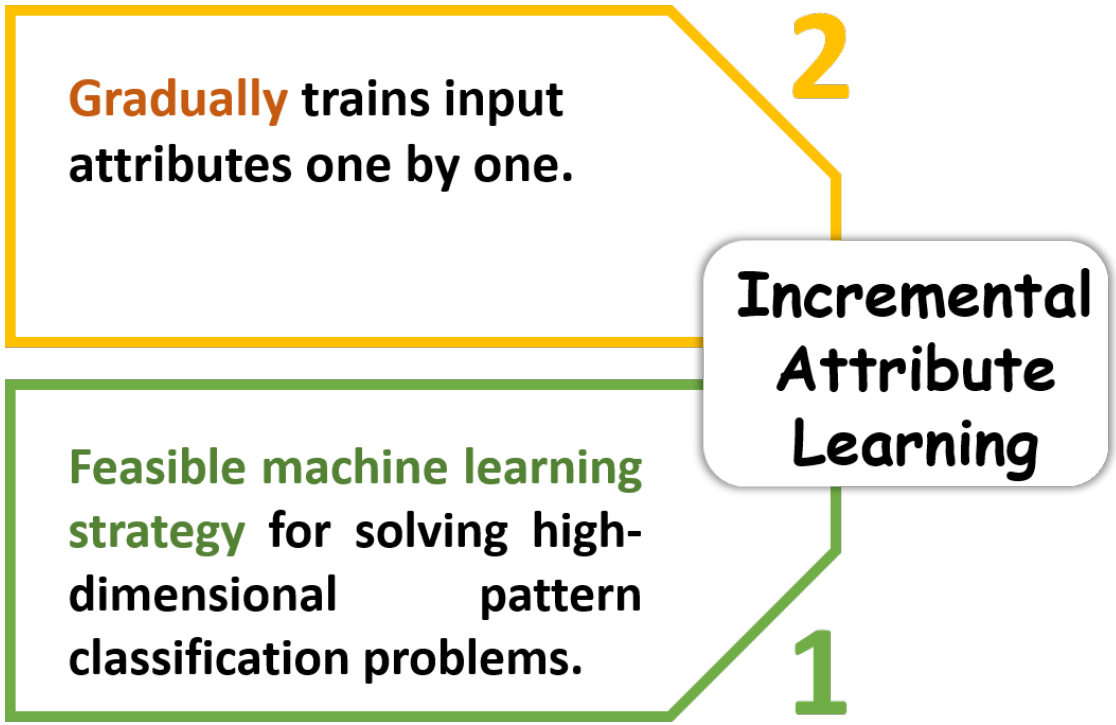
↷ *High requirements of memory space!*

- Deals only with **object** learning **in batch**.
- **Inability** to handle **incremental attribute learning** task.

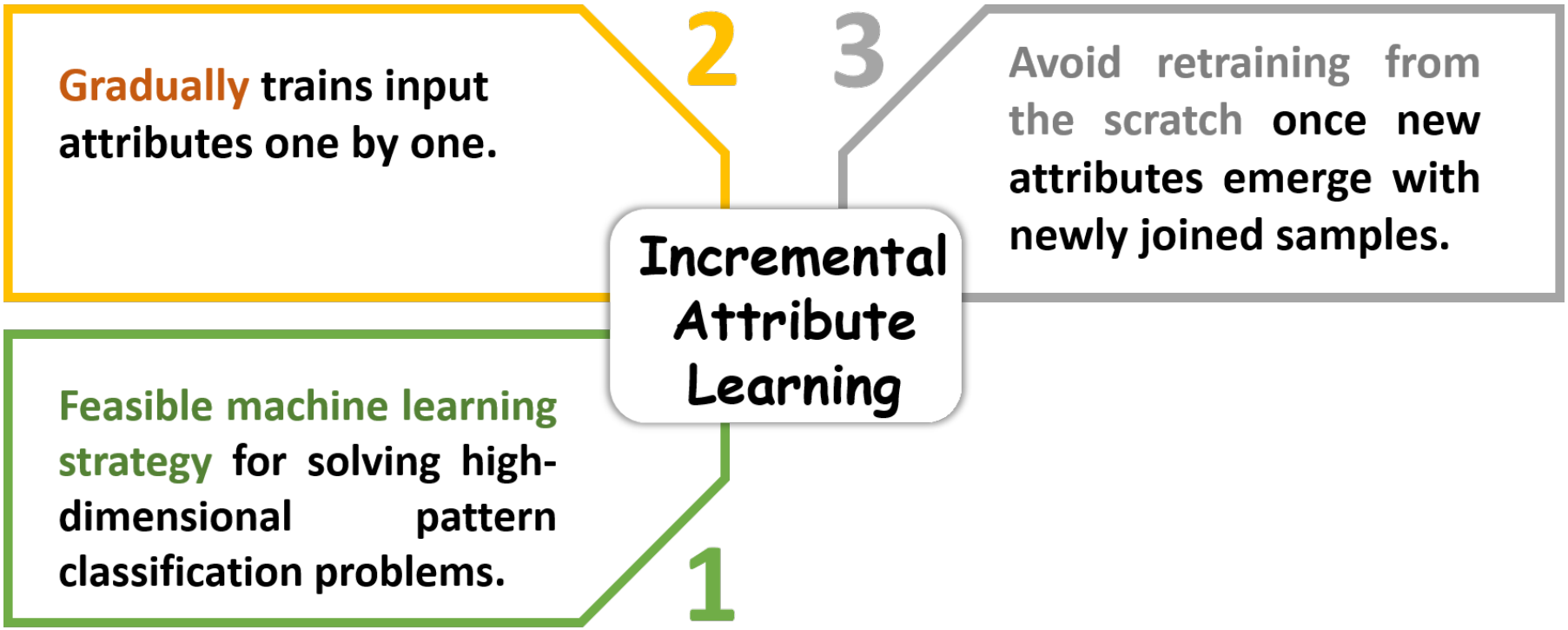
Incremental Attribute Learning



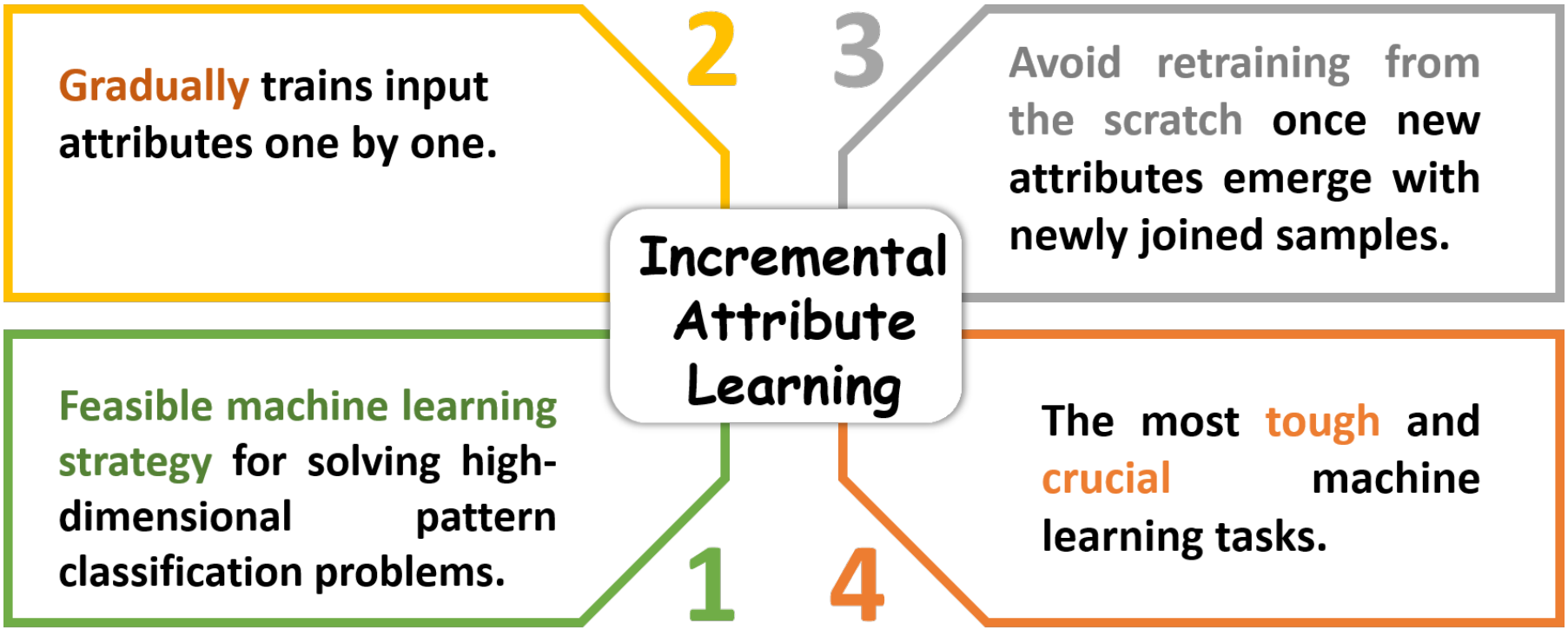
Incremental Attribute Learning



Incremental Attribute Learning



Incremental Attribute Learning



Problematic

1. **Mixed** medical data with **new emerging attributes** has not yet been evaluated.
2. Mining these massive data sets **faster** and **more accurately**.
3. Making it fluently accessible for **predictive analysis**.

↷ *remains a key challenge of the health care and medical industry!*

Problematic

1. **Mixed** medical data with **new emerging attributes** has not yet been evaluated.
2. Mining these massive data sets **faster** and **more accurately**.
3. Making it fluently accessible for **predictive analysis**.

↷ *remains a key challenge of the health care and medical industry!*

Problematic

1. **Mixed** medical data with **new emerging attributes** has not yet been evaluated.
2. Mining these massive data sets **faster** and **more accurately**.
3. Making it fluently accessible for **predictive analysis**.

↷ *remains a key challenge of the health care and medical industry!*

Problematic

1. **Mixed** medical data with **new emerging attributes** has not yet been evaluated.
2. Mining these massive data sets **faster** and **more accurately**.
3. Making it fluently accessible for **predictive analysis**.

↷ ***remains a key challenge of the health care and medical industry!***

Section 3

Proposed Incremental K-prototypes

Definition of IK-prototypes

IK-prototypes

- Big data solution in medicine and healthcare fields, through incremental attribute learning context.
- Proposed towards handling **mixed** large scale data in the form of **continuously** emerging data streams
  - escorted with **new added features**.

⇒ As **data stream** proceeds, **IK-prototypes** tackles both **incremental object** and **attribute** learning at the same deal.

Definition of IK-prototypes

IK-prototypes

- Big data solution in medicine and healthcare fields, through incremental attribute learning context.
- Proposed towards handling **mixed** large scale data in the form of **continuously** emerging data streams
  - escorted with **new added features**.

⇒ As **data stream** proceeds, **IK-prototypes** tackles both **incremental object** and **attribute** learning at the same deal.

Definition of IK-prototypes

Objectives

1. Manage the incremental attribute learning task in medical healthcare field.
2. Respect better basics of clustering in terms of dispersion of elements within and between clusters.
3. Reduce time processing when assigning the incoming objects with new attributes to their appropriate clusters.

Definition of IK-prototypes

Objectives

1. Manage the incremental attribute learning task in medical healthcare field.
2. Respect better basics of clustering in terms of dispersion of elements within and between clusters.
3. Reduce time processing when assigning the incoming objects with new attributes to their appropriate clusters.

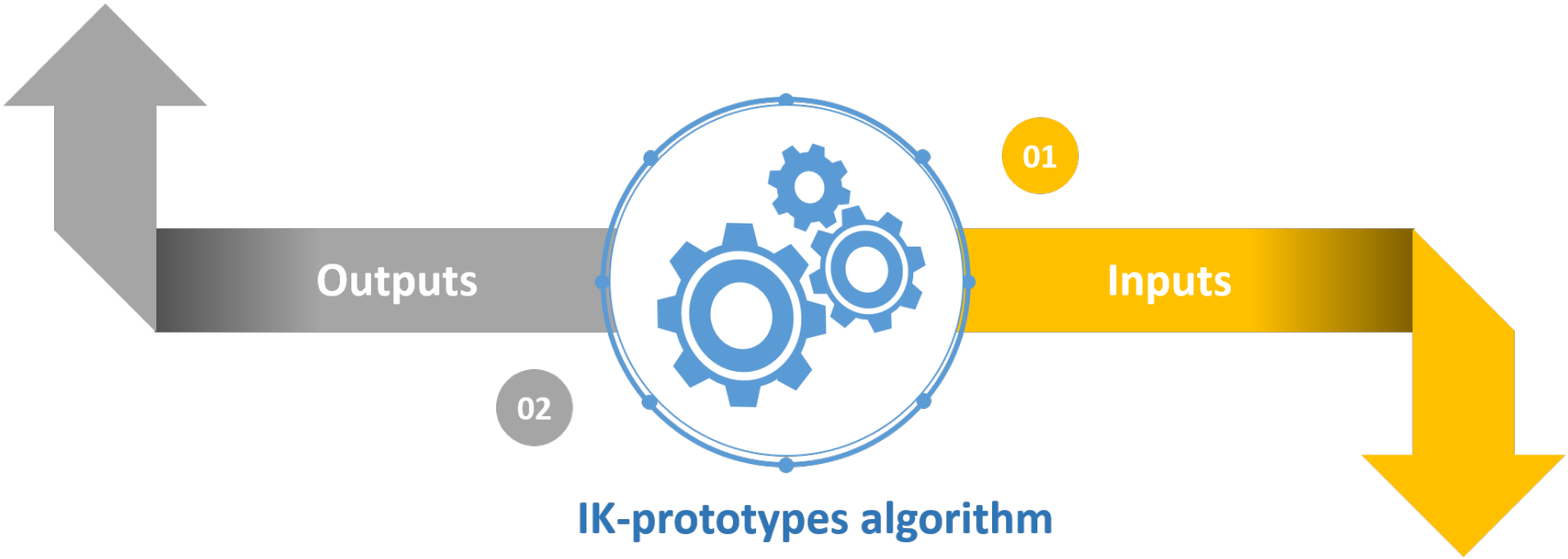
Definition of IK-prototypes

Objectives

1. Manage the incremental attribute learning task in medical healthcare field.
2. Respect better basics of clustering in terms of dispersion of elements within and between clusters.
3. Reduce time processing when assigning the incoming objects with new attributes to their appropriate clusters.

Definition of IK-prototypes

- K clusters with new attributes



- Input mixed data
- Data streams with additive patient's attributes

IK-prototypes steps

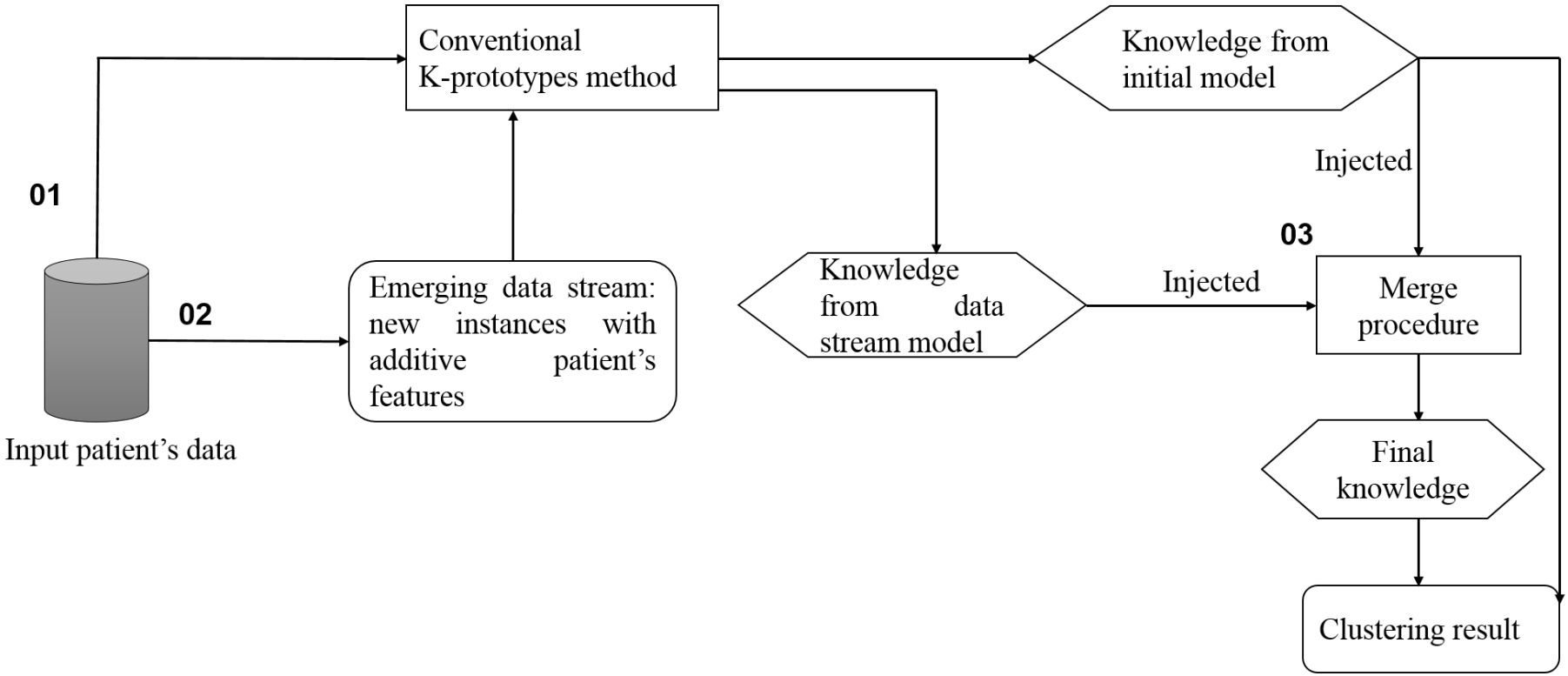


Figure 1: An overview of the proposed Incremental k-prototypes through IAL context

Merge process

- Merge the knowledge coming from both models
  - ↷ each two **similar clusters** are combined together
  - ↷ return to the initial k

Merge process

- Merge the knowledge coming from both models
  - ↷ each two **similar clusters** are combined together
  - ↷ return to the initial k

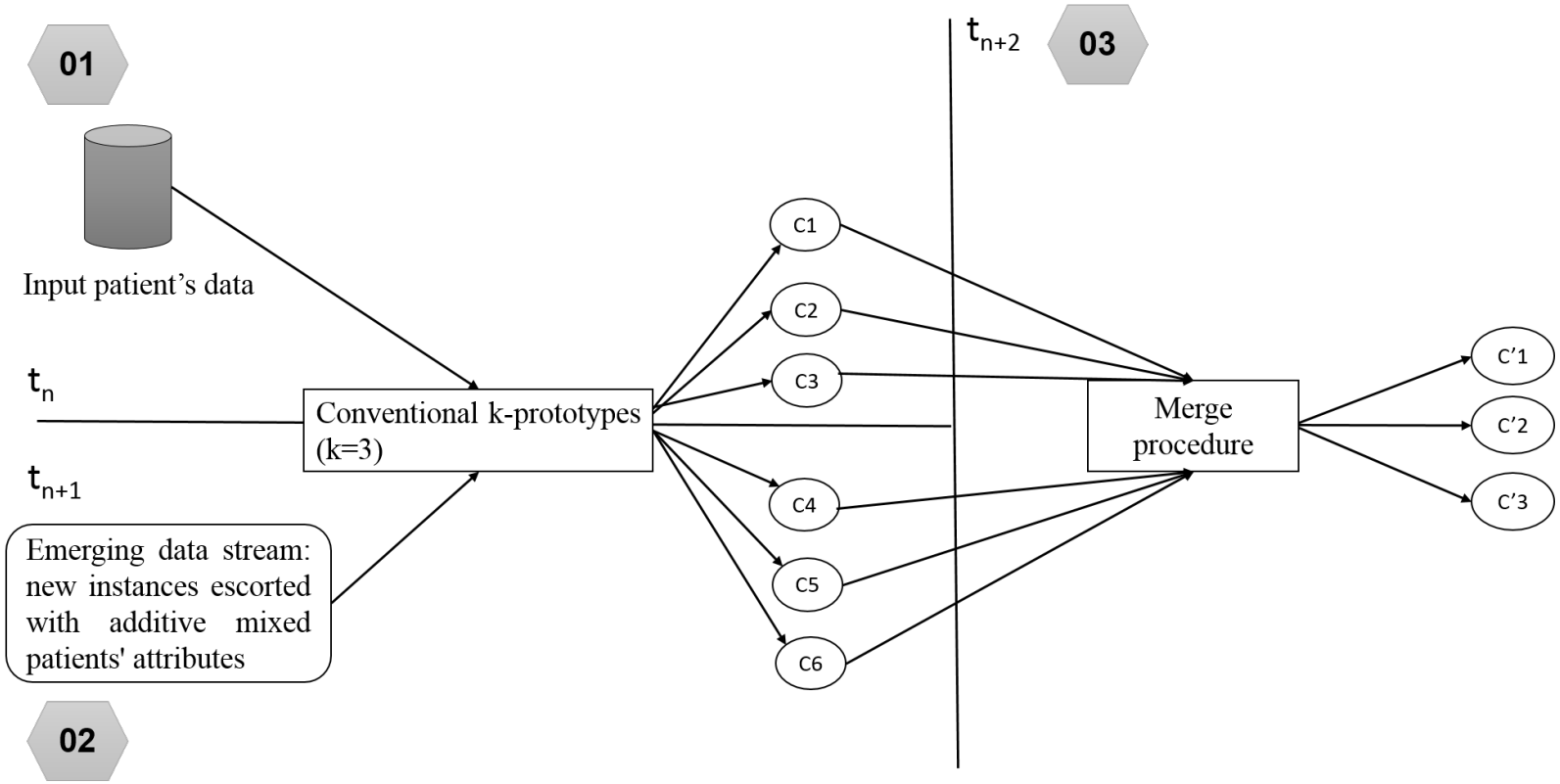


Figure 2: Workflow of the merge process

Merge process

Similarity measure

1. **Davies-Bouldin index (DB)** calculates the average similarity between clusters
   - Similarity based on a comparison between the **distance** between clusters and **the size** of the clusters themselves.
   - The **lower DB** is, the **better** partition of clusters is.
2. **Calinski-Harabasz Index (CH)** is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters.
   - The **CH** score is **high** when clusters are **dense** and **well separated**.

Merge process

Similarity measure

1. **Davies-Bouldin index (DB)** calculates the average similarity between clusters
   - Similarity based on a comparison between the **distance** between clusters and **the size** of the clusters themselves.
   - The **lower DB** is, the **better** partition of clusters is.
2. **Calinski-Harabasz Index (CH)** is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters.
   - The **CH** score is **high** when clusters are **dense** and **well separated**.

Merge process

- Our algorithm will choose the clusters corresponding to the **highest DB** value, **coinciding** with the **lowest CH** score.

	C1	C2	C3	C4	C5	C6
C1	0.0	****	****	****	****	****
C2	1.431	0. 0	****	****	****	****
C3	0.718	1.051	0.0	****	****	****
C4	**2.196**	1.382	0.796	0.0	****	****
C5	0.938	1.178	1.187	1.144	0.0	****
C6	0.689	0.953	1.597	0.776	1.464	0.0

	C1	C2	C3	C4	C5	C6
C1	10000	****	****	****	****	****
C2	102.047	10000	****	****	****	****
C3	450.566	241.2	10000	****	****	****
C4	**33.759**	95.732	316.75	10000	****	****
C5	249.744	180.158	204.4	146.258	10000	****
C6	302.609	176.655	67.716	210.596	77.58	10000

Figure 3: DB index and CH score matrices for each two clusters

✓ Indexes represented in bold refer to the same couple of clusters that will be merged.

⇒ *The **incremental attribute learning** is established without retraining from the scratch.*

Merge process

- Our algorithm will choose the clusters corresponding to the **highest DB** value, **coinciding** with the **lowest CH** score.

	C1	C2	C3	C4	C5	C6
C1	0.0	****	****	****	****	****
C2	1.431	0. 0	****	****	****	****
C3	0.718	1.051	0.0	****	****	****
C4	**2.196**	1.382	0.796	0.0	****	****
C5	0.938	1.178	1.187	1.144	0.0	****
C6	0.689	0.953	1.597	0.776	1.464	0.0

	C1	C2	C3	C4	C5	C6
C1	10000	****	****	****	****	****
C2	102.047	10000	****	****	****	****
C3	450.566	241.2	10000	****	****	****
C4	**33.759**	95.732	316.75	10000	****	****
C5	249.744	180.158	204.4	146.258	10000	****
C6	302.609	176.655	67.716	210.596	77.58	10000

Figure 3: DB index and CH score matrices for each two clusters

✓ Indexes represented in bold refer to the same couple of clusters that will be merged.

⇒ *The incremental attribute learning is established without retraining from the scratch.*

Merge process

- ✓ Indexes represented in bold refer to the same couple of clusters that will be merged.

⇒ *The* ***incremental attribute learning*** *is established without retraining from the scratch.*

Merge process

- The **highest DB** index and the **lowest CH** score may not be the best choices for the merge procedure
  - if they result in different combinations of clusters.

⇒ The algorithm will carry on

1. **Merge** both clusters resulting from the two calculated indexes
2. Calculate the **sum squared distances** of objects to their closest cluster center of the resulted merged clusters
3. Maintaining the cluster with the **lowest SSE**.

Merge process

- The **highest DB** index and the **lowest CH** score may not be the best choices for the merge procedure
  - if they result in different combinations of clusters.

⇒ The algorithm will carry on

1. **Merge** both clusters resulting from the two calculated indexes
2. Calculate the **sum squared distances** of objects to their closest cluster center of the resulted merged clusters
3. Maintaining the cluster with the **lowest SSE**.

Merge process

- The **highest DB** index and the **lowest CH** score may not be the best choices for the merge procedure
  - if they result in different combinations of clusters.

⇒ The algorithm will carry on

1. **Merge** both clusters resulting from the two calculated indexes
2. Calculate the **sum squared distances** of objects to their closest cluster center of the resulted merged clusters
3. Maintaining the cluster with the **lowest SSE**.

Merge process

- The **highest DB** index and the **lowest CH** score may not be the best choices for the merge procedure
  - if they result in different combinations of clusters.

⇒ The algorithm will carry on

1. **Merge** both clusters resulting from the two calculated indexes
2. Calculate the **sum squared distances** of objects to their closest cluster center of the resulted merged clusters
3. Maintaining the cluster with the **lowest SSE**.

Section 4

Experimental results

Real data sets description

Data set	#Instance	#Attribute	Acronym
Stroke Prediction	5110	12	SP
Pharmaceutical Drug Spending	1036	7	PDS
Breast Cancer Wisconsin	569	32	BCW
Personality Scale Analysis	315	8	PSA

- Breast Cancer Wiscowin data set is derived from the UCI machine learning repository.
- The rest of the data sets are imported from Kaggle.

Real data sets description

Data set	#Instance	#Attribute	Acronym
Stroke Prediction	5110	12	SP
Pharmaceutical Drug Spending	1036	7	PDS
Breast Cancer Wisconsin	569	32	BCW
Personality Scale Analysis	315	8	PSA

- Breast Cancer Wiscowin data set is derived from the UCI machine learning repository.
- The rest of the data sets are imported from Kaggle.

Real data sets description

Data set	#Instance	#Attribute	Acronym
Stroke Prediction	5110	12	SP
Pharmaceutical Drug Spending	1036	7	PDS
Breast Cancer Wisconsin	569	32	BCW
Personality Scale Analysis	315	8	PSA

- Breast Cancer Wiscowin data set is derived from the UCI machine learning repository.
- The rest of the data sets are imported from Kaggle.

Evaluation measures

1. **Sum of Squared Errors (SSE ↓)** is the sum of squared distances of objects to their closest cluster centres.
2. **Silhouette Coefficient (SC ↑)** is bounded between **−1** for incorrect clustering and **+1** for highly dense clustering.
   - ✓ A **higher SC score** relates to a model with **better defined** clusters.
3. **Run time (RT ↓)** is the time needed to achieve the final clustering result.

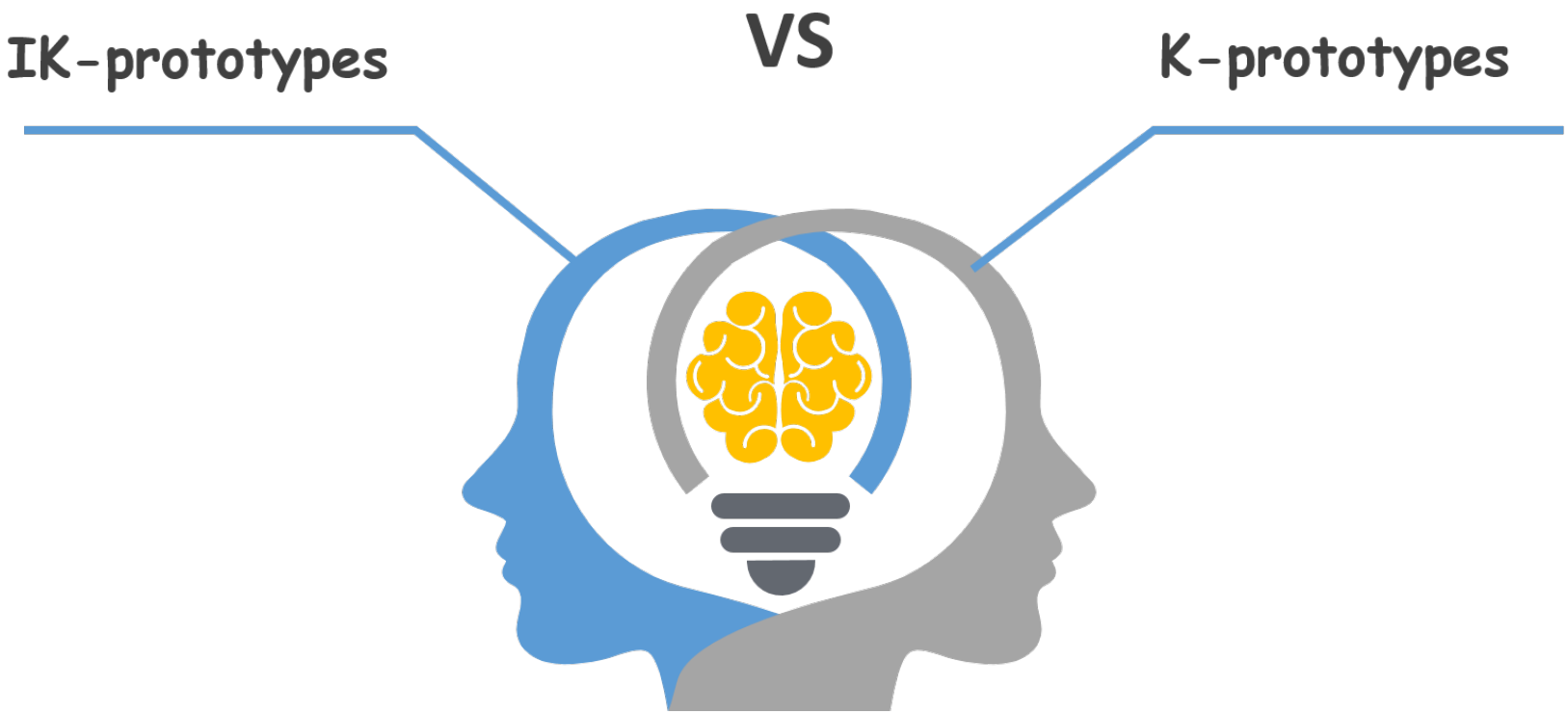
Evaluation measures

1. **Sum of Squared Errors (SSE ↓)** is the sum of squared distances of objects to their closest cluster centres.
2. **Silhouette Coefficient (SC ↑)** is bounded between **-1** for incorrect clustering and **+1** for highly dense clustering.
   - ✓ A **higher SC score** relates to a model with **better defined** clusters.
3. **Run time (RT ↓)** is the time needed to achieve the final clustering result.

Evaluation measures

1. **Sum of Squared Errors (SSE ↓)** is the sum of squared distances of objects to their closest cluster centres.
2. **Silhouette Coefficient (SC ↑)** is bounded between **-1** for incorrect clustering and **+1** for highly dense clustering.
   - ✓ A **higher SC score** relates to a model with **better defined** clusters.
3. **Run time (RT ↓)** is the time needed to achieve the final clustering result.

IK-prototypes VS K-prototypes



Sum of Squared Errors results

The total SSE is

Data sets	SP	PDS	BCW	PSA
K-prototypes	1.920	1.956	2.130	2.587
IK-prototypes	1.651	1.577	1.547	1.682

Sum of Squared Errors results

```
The total SSE is
```

Data sets	SP	PDS	BCW	PSA
K-prototypes	1.920	1.956	2.130	2.587
IK-prototypes	1.651	1.577	1.547	1.682

Sum of Squared Errors results

```
The total SSE is
```

Data sets	SP	PDS	BCW	PSA
K-prototypes	1.920	1.956	2.130	2.587
IK-prototypes	1.651	1.577	1.547	1.682

Silhouette Coefficient results

The SC is

Data sets	SP	PDS	BCW	PSA
K-prototypes	0.363	0.469	0.457	0.487
IK-prototypes	0.388	0.523	0.467	0.576

Silhouette Coefficient results

The SC is

Data sets	SP	PDS	BCW	PSA
K-prototypes	0.363	0.469	0.457	0.487
IK-prototypes	0.388	0.523	0.467	0.576

Silhouette Coefficient results

The SC is

Data sets	SP	PDS	BCW	PSA
K-prototypes	0.363	0.469	0.457	0.487
IK-prototypes	0.388	0.523	0.467	0.576

Run Time results

The RT is

Data sets	SP	PDS	BCW	PSA
K-prototypes	87.788	11.090	7.188	2.956
IK-prototypes	56.229	6.772	4.543	1.794

Run Time results

The RT is

Data sets	SP	PDS	BCW	PSA
K-prototypes	87.788	11.090	7.188	2.956
IK-prototypes	56.229	6.772	4.543	1.794

Run Time results

The RT is

Data sets	SP	PDS	BCW	PSA
K-prototypes	87.788	11.090	7.188	2.956
IK-prototypes	56.229	6.772	4.543	1.794

Section 5

Conclusion and future works

Conclusion

- ✓ We gain clustering method that
  - cluster new emerging attributes with newly added instances in streaming data,
  - provide a well defined model with better separation between clusters,
  - in less time consuming.
- ✓ The IK-prototypes outperforms the k-prototypes method based on different evaluation criteria.
- ✓ Helping in **early detection of diseases**, **treatment recommendations**, and **clinical services to doctors**.

Conclusion

- ✓ We gain clustering method that
  - cluster new emerging attributes with newly added instances in streaming data,
  - provide a well defined model with better separation between clusters,
  - in less time consuming.
- ✓ The IK-prototypes outperforms the k-prototypes method based on different evaluation criteria.
- ✓ Helping in **early detection of diseases**, **treatment recommendations**, and **clinical services to doctors**.

Conclusion

- ✓ We gain clustering method that
  - cluster new emerging attributes with newly added instances in streaming data,
  - provide a well defined model with better separation between clusters,
  - in less time consuming.
- ✓ The IK-prototypes outperforms the k-prototypes method based on different evaluation criteria.
- ✓ Helping in **early detection of diseases**, **treatment recommendations**, and **clinical services to doctors**.

Future works

1. Form an enhanced version of the IK-prototypes algorithm, capable to deal with **evolving feature and object** spaces.
2. Extend our method to be able to handle the **decremental attribute and object** learning aspects in **medicine** and **healthcare** fields.
3. Perform a **feature selection** preprocessing technique before modeling new emerging **medical data streams**.

Future works

1. Form an enhanced version of the IK-prototypes algorithm, capable to deal with **evolving feature and object** spaces.
2. Extend our method to be able to handle the **decremental attribute and object** learning aspects in **medicine** and **healthcare** fields.
3. Perform a **feature selection** preprocessing technique before modeling new emerging **medical data streams**.

Future works

1. Form an enhanced version of the IK-prototypes algorithm, capable to deal with **evolving feature and object** spaces.
2. Extend our method to be able to handle the **decremental attribute and object** learning aspects in **medicine** and **healthcare** fields.
3. Perform a **feature selection** preprocessing technique before modeling new emerging **medical data streams**.

Thank you.