Maximal Closed Set and Half-Space Separations in Finite Closure Systems

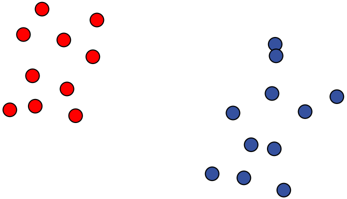
Florian Seiffarth, Tamás Horváth, Stefan Wrobel

Fraunhofer IAIS

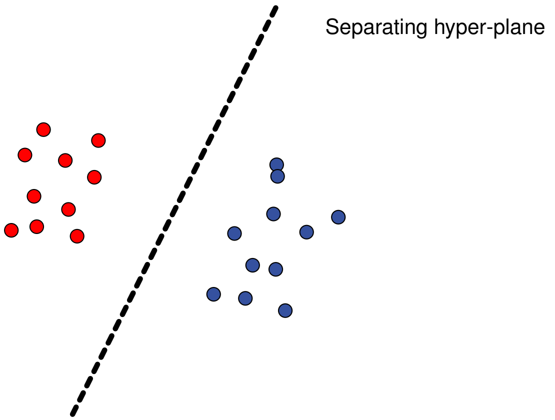
UNIVERSITÄT BONN

Hyper-plane Separation in $\mathbb{R}^d$

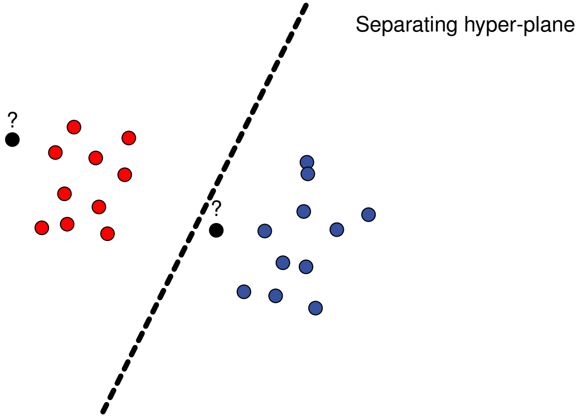
Hyper-plane Separation in $\mathbb{R}^d$



Hyper-plane Separation in $\mathbb{R}^d$

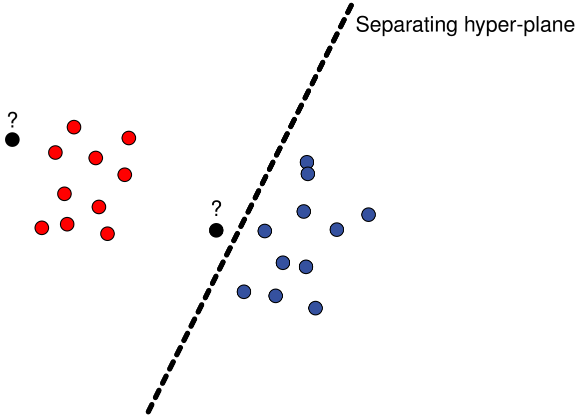


Hyper-plane Separation in $\mathbb{R}^d$



- classical machine learning problem (Rosenblatt, 1958), well-understood

Hyper-plane Separation in $\mathbb{R}^d$



- classical machine learning problem (Rosenblatt, 1958), well-understood
- not unique

This Work: Generalization to Finite Set Systems

Definition:

- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

This Work: Generalization to Finite Set Systems

Motivation:

Definition:

- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

This Work: Generalization to Finite Set Systems

Definition:

- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

Motivation:

- theoretical:

This Work: Generalization to Finite Set Systems

Definition:

- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

Motivation:

- theoretical:

Theorem (Kakutani):

Two sets in $\mathbb{R}^d$ are **separable** by a hyper-plane $\iff$ their convex hulls are **disjoint**

This Work: Generalization to Finite Set Systems

Definition:

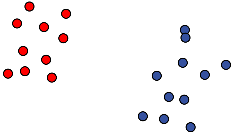
- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

Motivation:

- theoretical:

Theorem (Kakutani):

Two sets in $\mathbb{R}^d$ are **separable** by a hyper-plane $\iff$ their convex hulls are **disjoint**



This Work: Generalization to Finite Set Systems

Definition:

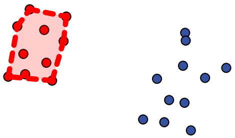
- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

Motivation:

- theoretical:

Theorem (Kakutani):

Two sets in $\mathbb{R}^d$ are **separable** by a hyper-plane $\iff$ their convex hulls are **disjoint**



This Work: Generalization to Finite Set Systems

Definition:

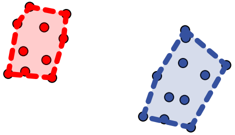
- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

Motivation:

- theoretical:

Theorem (Kakutani):

Two sets in $\mathbb{R}^d$ are **separable** by a hyper-plane $\iff$ their convex hulls are **disjoint**



This Work: Generalization to Finite Set Systems

Definition:

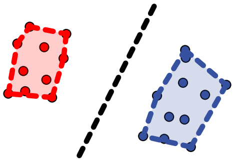
- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

Motivation:

- theoretical:

Theorem (Kakutani):

Two sets in $\mathbb{R}^d$ are **separable** by a hyper-plane $\iff$ their convex hulls are **disjoint**



This Work: Generalization to Finite Set Systems

Definition:

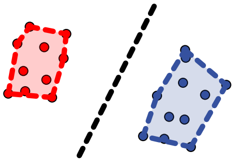
- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

Motivation:

- theoretical:

Theorem (Kakutani):

Two sets in $\mathbb{R}^d$ are **separable** by a hyper-plane $\iff$ their convex hulls are **disjoint**



- practical:

This Work: Generalization to Finite Set Systems

Definition:

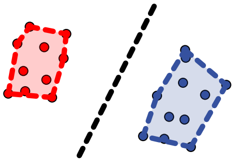
- set system $(E, \mathcal{F})$: set E with $\mathcal{F} \subseteq 2^E$
- *finite* set system: $|E| < \infty$

Motivation:

- theoretical:

Theorem (Kakutani):

Two sets in $\mathbb{R}^d$ are **separable** by a hyper-plane $\iff$ their convex hulls are **disjoint**



- practical: structured input space (examples in a few minutes)

This Work: Generalization to Finite Set Systems

This Work: Generalization to Finite **Closure** Systems

This Work: Generalization to Finite **Closure** Systems

What is what:

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$ $\rightarrow$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d \rightarrow E$ (ground set)

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set
all convex hulls	$\rightarrow$	

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set
all convex hulls	$\rightarrow$	closure system

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

Definition:

closure system $(E, \mathcal{C})$:

- $\emptyset, E \in \mathcal{C}$
- $A, B \in \mathcal{C} \Rightarrow A \cap B \in \mathcal{C}$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set
all convex hulls	$\rightarrow$	closure system
half-space	$\rightarrow$	

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

Definition:

closure system $(E, \mathcal{C})$:

- $\emptyset, E \in \mathcal{C}$
- $A, B \in \mathcal{C} \Rightarrow A \cap B \in \mathcal{C}$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set
all convex hulls	$\rightarrow$	closure system
half-space	$\rightarrow$	closed set with closed complement

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

Definition:

closure system $(E, \mathcal{C})$:

- $\emptyset, E \in \mathcal{C}$
- $A, B \in \mathcal{C} \Rightarrow A \cap B \in \mathcal{C}$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set
all convex hulls	$\rightarrow$	closure system
half-space	$\rightarrow$	closed set with closed complement
hyper-plane	$\rightarrow$	

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

Definition:

closure system $(E, \mathcal{C})$:

- $\emptyset, E \in \mathcal{C}$
- $A, B \in \mathcal{C} \Rightarrow A \cap B \in \mathcal{C}$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\to$	E (ground set)
convexity	$\to$	closure operator
convex hull	$\to$	closed set
all convex hulls	$\to$	closure system
half-space	$\to$	closed set with closed complement
hyper-plane	$\to$	*no correspondance*

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

Definition:

closure system $(E, \mathcal{C})$:

- $\emptyset, E \in \mathcal{C}$
- $A, B \in \mathcal{C} \Rightarrow A \cap B \in \mathcal{C}$

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set
all convex hulls	$\rightarrow$	closure system
half-space	$\rightarrow$	closed set with closed complement
hyper-plane	$\rightarrow$	*no correspondance*

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

Definition:

closure system $(E, \mathcal{C})$:

- $\emptyset, E \in \mathcal{C}$
- $A, B \in \mathcal{C} \Rightarrow A \cap B \in \mathcal{C}$

Questions:

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set
all convex hulls	$\rightarrow$	closure system
half-space	$\rightarrow$	closed set with closed complement
hyper-plane	$\rightarrow$	*no correspondance*

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

Definition:

closure system $(E, \mathcal{C})$:

- $\emptyset, E \in \mathcal{C}$
- $A, B \in \mathcal{C} \Rightarrow A \cap B \in \mathcal{C}$

Questions:

- separation algorithm

This Work: Generalization to Finite **Closure** Systems

What is what:

$\mathbb{R}^d$	$\rightarrow$	E (ground set)
convexity	$\rightarrow$	closure operator
convex hull	$\rightarrow$	closed set
all convex hulls	$\rightarrow$	closure system
half-space	$\rightarrow$	closed set with closed complement
hyper-plane	$\rightarrow$	*no correspondance*

Definition:

closure operator c over E:

- $A \subseteq c(A)$
- $A \subseteq B \Rightarrow c(A) \subseteq c(B)$
- $c(c(A)) = c(A)$

Definition:

closure system $(E, \mathcal{C})$:

- $\emptyset, E \in \mathcal{C}$
- $A, B \in \mathcal{C} \Rightarrow A \cap B \in \mathcal{C}$

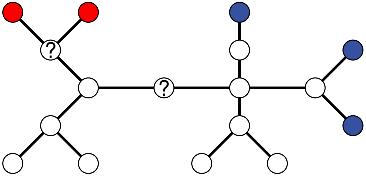
Questions:

- separation algorithm
- classification algorithm

Some Practical Motivations:

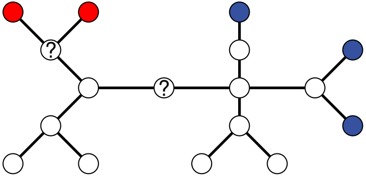
Some Practical Motivations: Graphs

Some Practical Motivations: Graphs

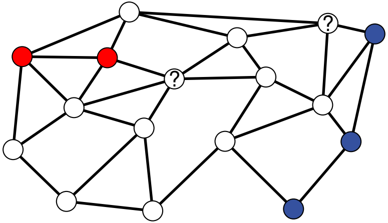


Trees

Some Practical Motivations: Graphs

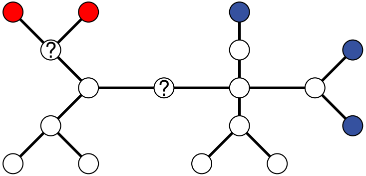


Trees

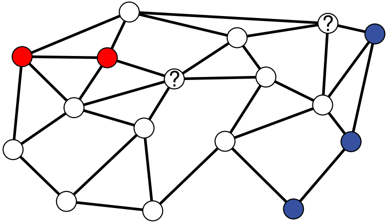


Graphs (e.g. molecule, social graph, . . .)

Some Practical Motivations: Graphs



Trees



Graphs (e.g. molecule, social graph, ...)

- requires some **semantically** meaningful definition of closure system/operator

Example: Closure on trees

Example: Closure on trees

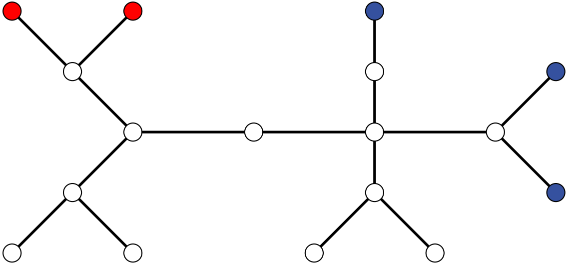
- A node set X is **closed** if it contains all shortest paths between nodes in X.

Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.

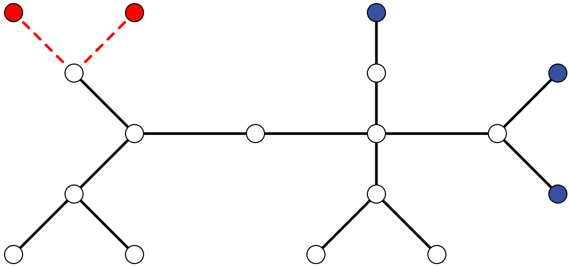
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



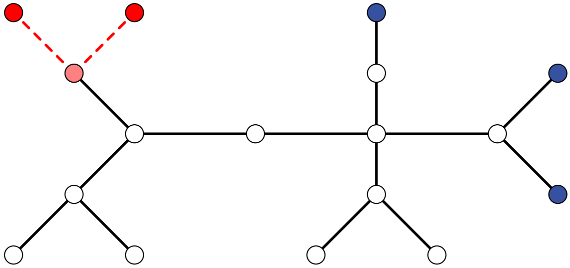
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



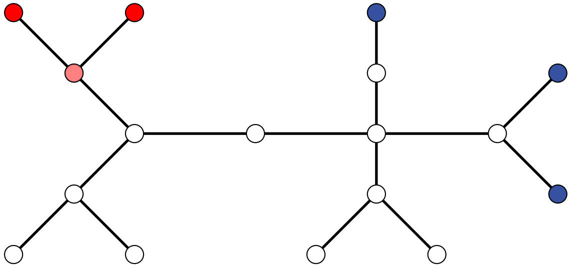
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



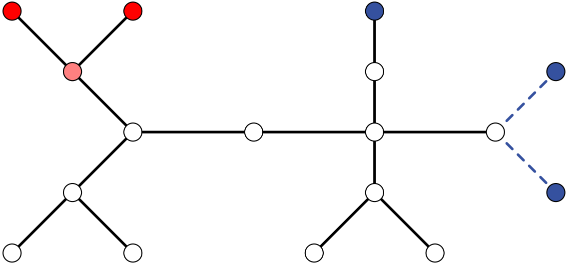
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



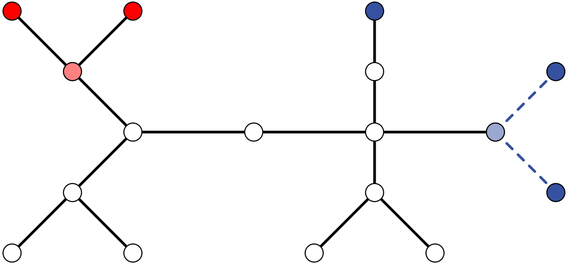
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



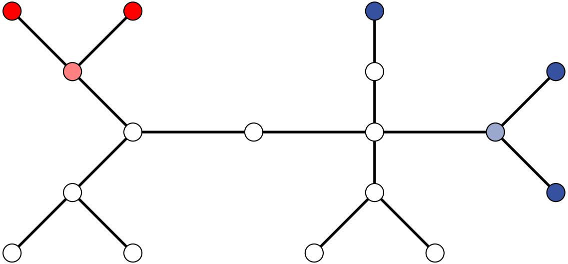
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



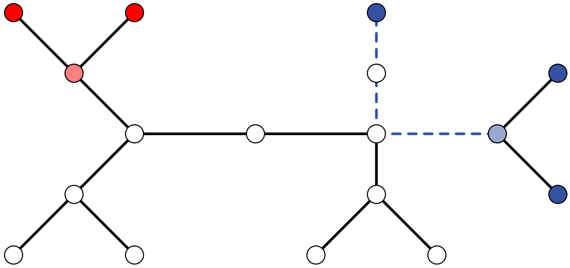
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



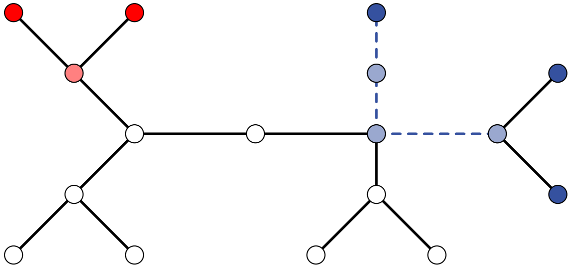
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



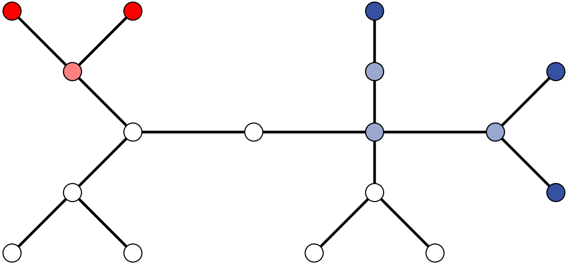
Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.

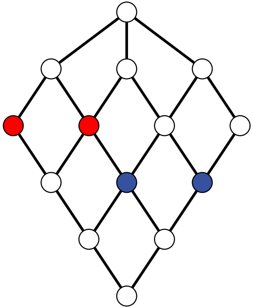


Example: Closure on trees

- A node set X is **closed** if it contains all shortest paths between nodes in X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



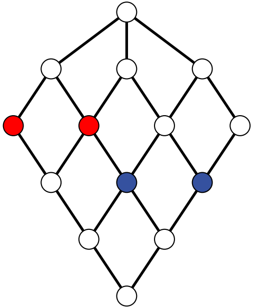
Some Practical Motivations: Lattices



Lattices (e.g. subsumption lattice, formal concept lattice, . . .)

Inductive Logic Programming (ILP) — Formal Concept Analysis (FCA)

Some Practical Motivations: Lattices



Lattices (e.g. subsumption lattice, formal concept lattice, . . .)

subsumption lattice: Inductive Logic Programming (ILP)

formal concept lattice: Formal Concept Analysis (FCA)

- requires some **semantically** meaningful definition of closure system/operator

Example: Subsumption sub-lattice ILP

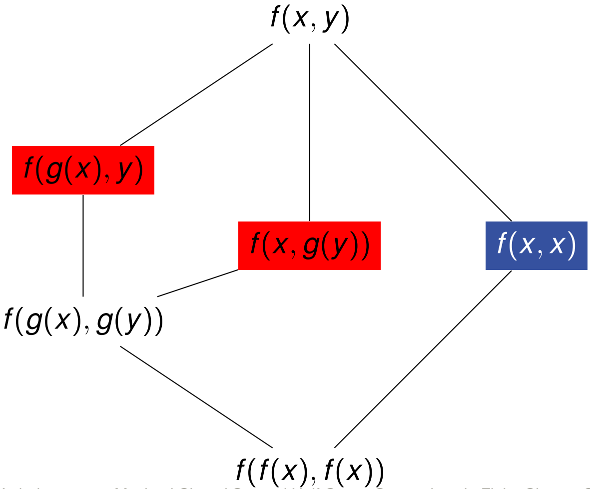
- A set X is **closed** if it contains all elements between the meet (glb) and the join (lub) of X.

Example: Subsumption sub-lattice ILP

- A set X is **closed** if it contains all elements between the meet (glb) and the join (lub) of X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.

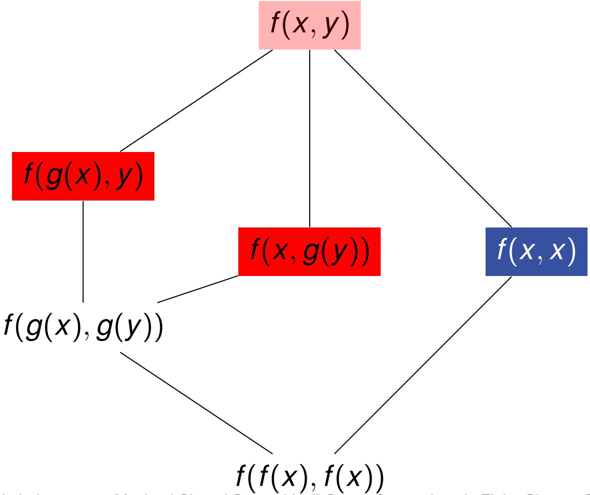
Example: Subsumption sub-lattice ILP

- A set X is **closed** if it contains all elements between the meet (glb) and the join (lub) of X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



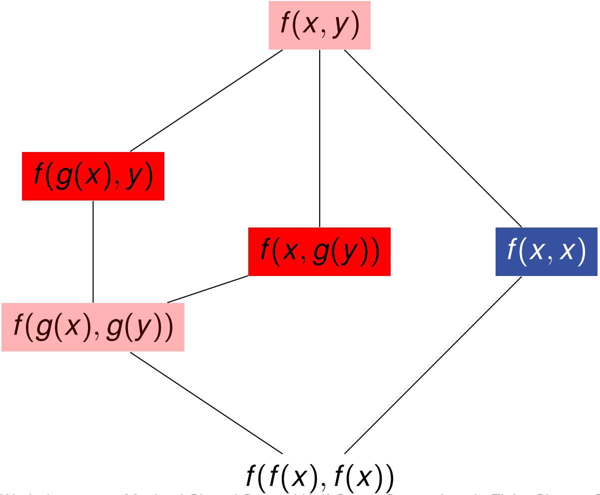
Example: Subsumption sub-lattice ILP

- A set X is **closed** if it contains all elements between the meet (glb) and the join (lub) of X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



Example: Subsumption sub-lattice ILP

- A set X is **closed** if it contains all elements between the meet (glb) and the join (lub) of X.
- The **closure** $c(Y)$ is the smallest closed set containing Y.



Problem Definition

Problem Definition

Half-Space-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and two sets $A, B \subseteq E$, **decide** if they are half-space separable in $(E, \mathcal{C})$.

Problem Definition

Half-Space-Separation Problem:

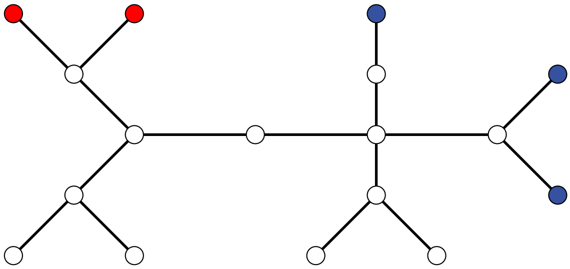
Given a closure system $(E, \mathcal{C})$ and two sets $A, B \subseteq E$, **decide** if they are half-space separable in $(E, \mathcal{C})$.

Remark: the closure system is given implicitly via a closure operator.

Problem Definition

Half-Space-Separation Problem:

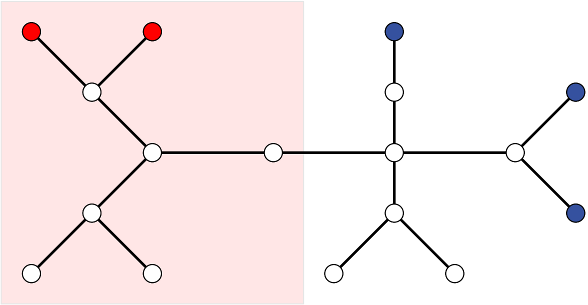
Given a closure system $(E, \mathcal{C})$ and two sets $A, B \subseteq E$, **decide** if they are half-space separable in $(E, \mathcal{C})$.



Remark: the closure system is given implicitly via a closure operator.

Half-Space-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and two sets $A, B \subseteq E$, **decide** if they are half-space separable in $(E, \mathcal{C})$.

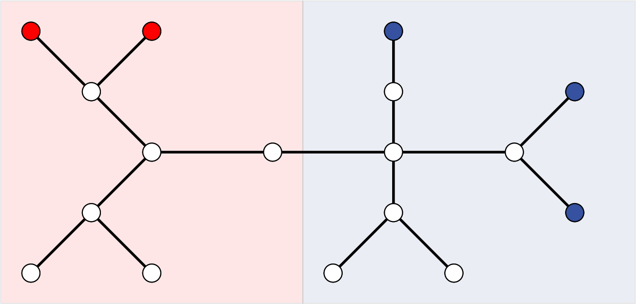


Remark: the closure system is given implicitly via a closure operator.

Problem Definition

Half-Space-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and two sets $A, B \subseteq E$, **decide** if they are half-space separable in $(E, \mathcal{C})$.



Remark: the closure system is given implicitly via a closure operator.

Some Negative Results

Some Negative Results

Proposition: The Kakutani Theorem **does not hold** for closure systems

Some Negative Results

Proposition: The Kakutani Theorem **does not hold** for closure systems

- i.e., there are disjoint closed sets which have no half-space separation

Some Negative Results

Proposition: The Kakutani Theorem **does not hold** for closure systems

- i.e., there are disjoint closed sets which have no half-space separation

Theorem: The Half-Space-Separation problem is **NP-complete**

Some Negative Results

Proposition: The Kakutani Theorem **does not hold** for closure systems

- i.e., there are disjoint closed sets which have no half-space separation

Theorem: The Half-Space-Separation problem is **NP-complete**

Two approaches to overcome:

Some Negative Results

Proposition: The Kakutani Theorem **does not hold** for closure systems

- i.e., there are disjoint closed sets which have no half-space separation

Theorem: The Half-Space-Separation problem is **NP-complete**

Two approaches to overcome:

- problem relaxation $\rightarrow$ find **maximal** separating closed sets
  - **caution**: maximal is not maximum

Some Negative Results

Proposition: The Kakutani Theorem **does not hold** for closure systems

- i.e., there are disjoint closed sets which have no half-space separation

Theorem: The Half-Space-Separation problem is **NP-complete**

Two approaches to overcome:

- problem relaxation $\rightarrow$ find **maximal** separating closed sets
  - **caution**: maximal is not maximum
- resort to Kakutani set systems $\rightarrow$ (set systems where Kakutani theorem holds)

Relaxed Problem Definition

Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print “No” if they do not exist

Relaxed Problem Definition

Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

Relaxed Problem Definition

Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print “No” if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Relaxed Problem Definition

Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

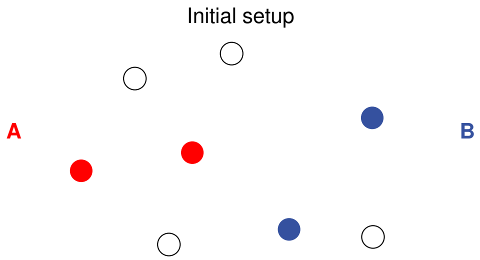
Relaxed Problem Definition

Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator



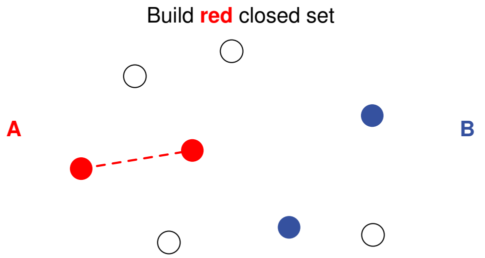
Relaxed Problem Definition

Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator



Relaxed Problem Definition

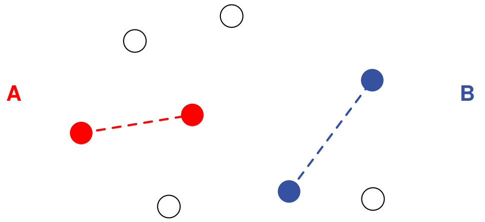
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Build **blue** closed set



Relaxed Problem Definition

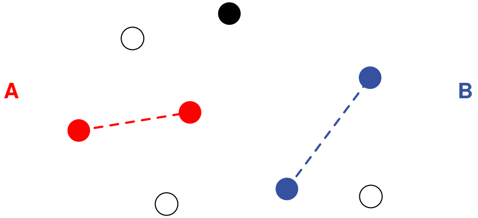
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print “No” if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Take arbitrary point



Relaxed Problem Definition

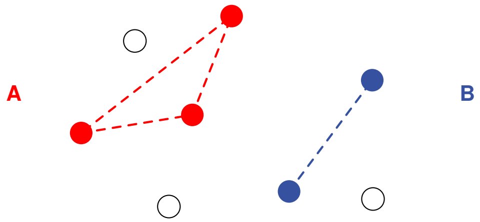
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print “No” if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Add to **red** closed if possible



Relaxed Problem Definition

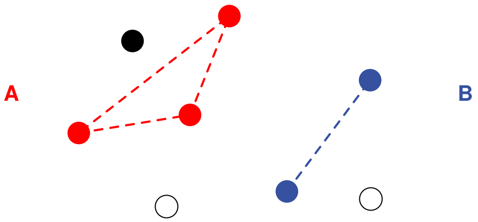
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Take arbitrary point



Relaxed Problem Definition

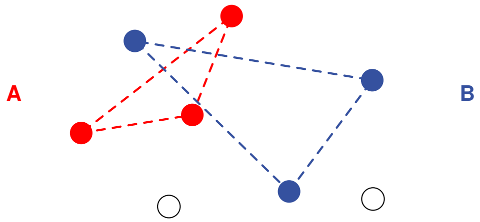
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Add to **blue** closed if possible



Relaxed Problem Definition

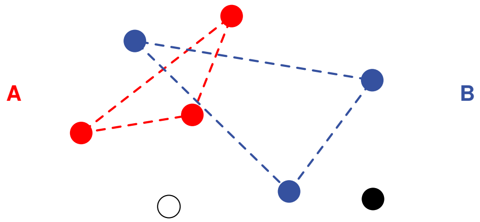
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Take arbitrary point



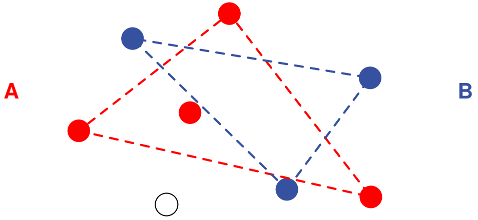
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print “No” if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Add to **red** closed if possible



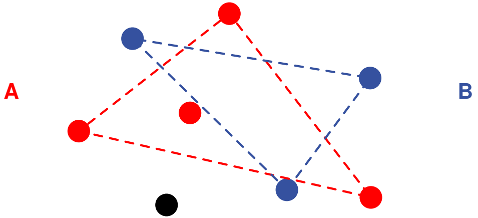
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Take arbitrary point



Relaxed Problem Definition

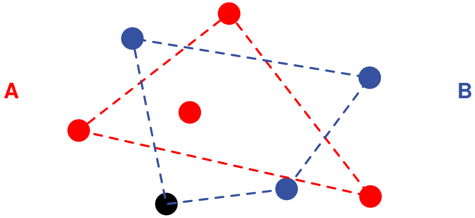
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Cannot added to **blue** because of intersection



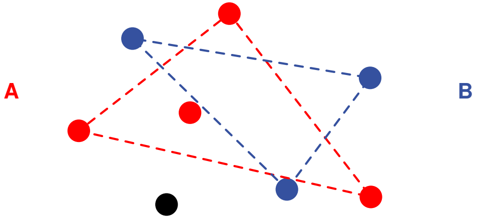
Relaxed Problem Definition

Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator



Relaxed Problem Definition

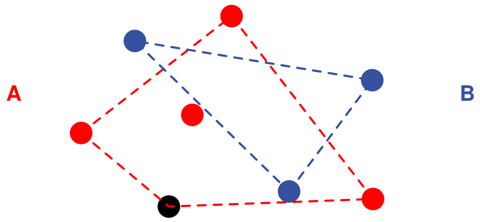
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print “No” if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Cannot added to red because of intersection



Relaxed Problem Definition

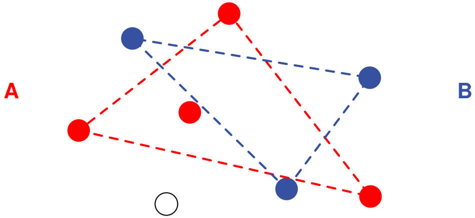
Maximal-Closed-Set-Separation Problem:

Given a closure system $(E, \mathcal{C})$ and sets $A, B \subseteq E$, find **maximal** disjoint closed sets separating A and B; print "No" if they do not exist

Our Solution: simple **greedy** algorithm

- very **generic** algorithm: requires only the ground set and the closure operator

Maximal closed set separation



Results

Theorem: It solves the Maximal Closed Set Separation problem by calling the closure operator at most $2|E| - 2$ times.

Results

Theorem: It solves the Maximal Closed Set Separation problem by calling the closure operator at most $2|E| - 2$ times.

Theorem: It is **optimal** in the number of closure operator calls.

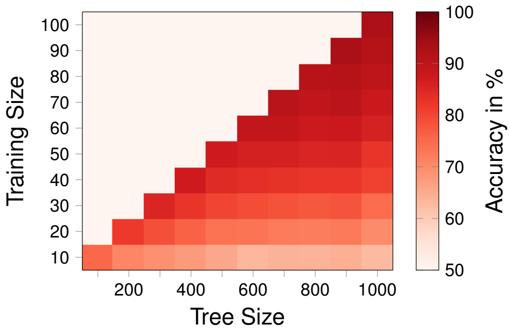
Results

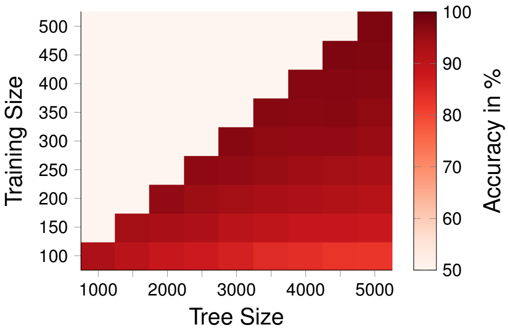
Theorem: It solves the Maximal Closed Set Separation problem by calling the closure operator at most $2|E| - 2$ times.

Theorem: It is **optimal** in the number of closure operator calls.

Theorem: It provides an **algorithmic characterization** of Kakutani closure systems.

Experimental Results on Trees





- classification results of the maximal closed set separation algorithm on trees

Summary

This work: Generalization of half-space separation in $\mathbb{R}^d$ to finite closure systems.

- NP-complete

Summary

This work: Generalization of half-space separation in $\mathbb{R}^d$ to finite closure systems.

- NP-complete
- two problem relaxations:

Summary

This work: Generalization of half-space separation in $\mathbb{R}^d$ to finite closure systems.

- NP-complete
- two problem relaxations:
    - Kakutani closure systems (e.g. trees, distributive lattices with applications to ILP (long version))

Summary

This work: Generalization of half-space separation in $\mathbb{R}^d$ to finite closure systems.

- NP-complete
- two problem relaxations:
  - Kakutani closure systems (e.g. trees, distributive lattices with applications to ILP (long version))
  - maximal closed set separation

Summary

This work: Generalization of half-space separation in $\mathbb{R}^d$ to finite closure systems.

- NP-complete
- two problem relaxations:
  - Kakutani closure systems (e.g. trees, distributive lattices with applications to ILP (long version))
  - maximal closed set separation
- maximal closed set separation problem: general simple greedy algorithm

Summary

This work: Generalization of half-space separation in $\mathbb{R}^d$ to finite closure systems.

- NP-complete
- two problem relaxations:
  - Kakutani closure systems (e.g. trees, distributive lattices with applications to ILP (long version))
  - maximal closed set separation
- maximal closed set separation problem: general simple greedy algorithm
  - no specific structure is utilized

Summary

This work: Generalization of half-space separation in $\mathbb{R}^d$ to finite closure systems.

- NP-complete
- two problem relaxations:
  - Kakutani closure systems (e.g. trees, distributive lattices with applications to ILP (long version))
  - maximal closed set separation
- maximal closed set separation problem: general simple greedy algorithm
  - no specific structure is utilized
  - optimal

Summary

This work: Generalization of half-space separation in $\mathbb{R}^d$ to finite closure systems.

- NP-complete
- two problem relaxations:
  - Kakutani closure systems (e.g. trees, distributive lattices with applications to ILP (long version))
  - maximal closed set separation
- maximal closed set separation problem: general simple greedy algorithm
  - no specific structure is utilized
  - optimal
  - excellent experimental results on trees

On-going/Future Work

- further applications of the special case of (distributive) lattices

On-going/Future Work

- further applications of the special case of (distributive) lattices
  - *long version . . .*

On-going/Future Work

- further applications of the special case of (distributive) lattices
  - *long version . . .*
- enriching the story with distances, hyper-planes etc.

On-going/Future Work

- further applications of the special case of (distributive) lattices
  - *long version . . .*
- enriching the story with distances, hyper-planes etc.
  - need appropriate generalizations

On-going/Future Work

- further applications of the special case of (distributive) lattices
  - *long version . . .*
- enriching the story with distances, hyper-planes etc.
  - need appropriate generalizations
- unique half-space separation (like SVM)

On-going/Future Work

- further applications of the special case of (distributive) lattices
  - *long version . . .*
- enriching the story with distances, hyper-planes etc.
  - need appropriate generalizations
- unique half-space separation (like SVM)
- utilization of the VC-dimension of finite closure systems