Karlsruhe Institute of Technology

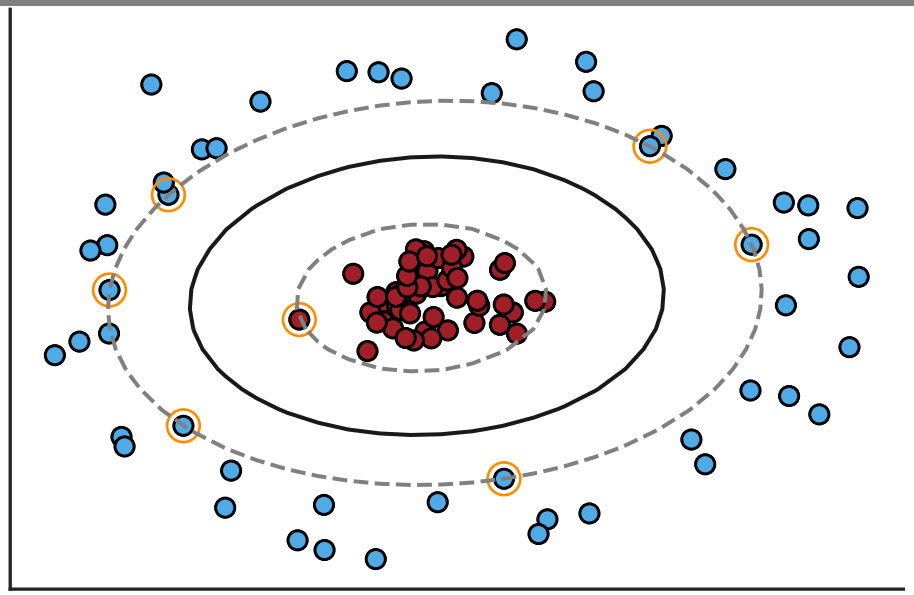
Faster Support Vector Machines

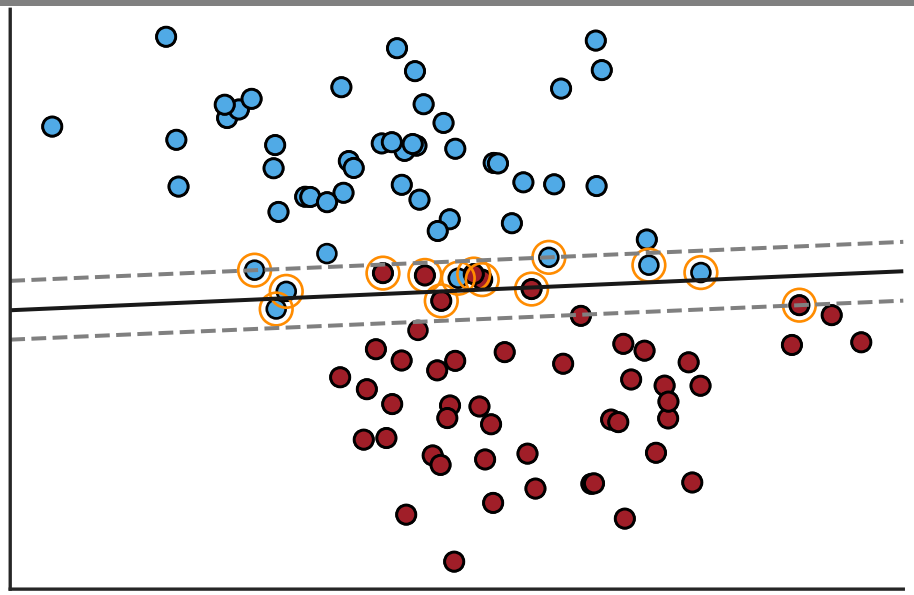
ALENEX'19 · January 8, 2019

Sebastian Schlag[†], **Matthias Schmitt**[†], **Christian Schulz**[‡]

[†]KIT, [‡]University of Vienna

INSTITUTE OF THEORETICAL INFORMATICS ·





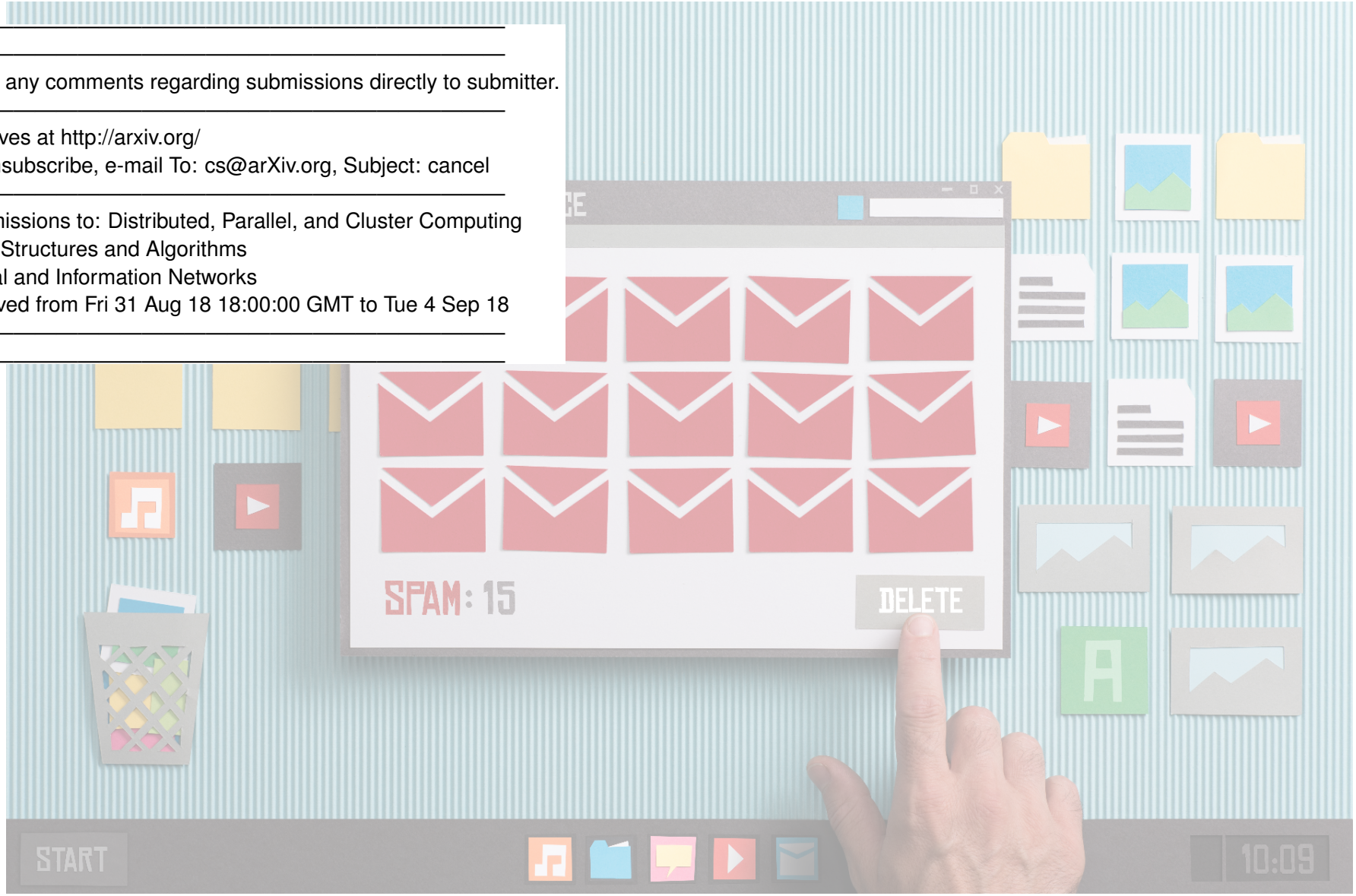
Binary Classification Problem

Binary Classification Problem

Send any comments regarding submissions directly to submitter.

Archives at http://arxiv.org/
To unsubscribe, e-mail To: cs@arXiv.org, Subject: cancel

Submissions to: Distributed, Parallel, and Cluster Computing
Data Structures and Algorithms
Social and Information Networks
received from Fri 31 Aug 18 18:00:00 GMT to Tue 4 Sep 18

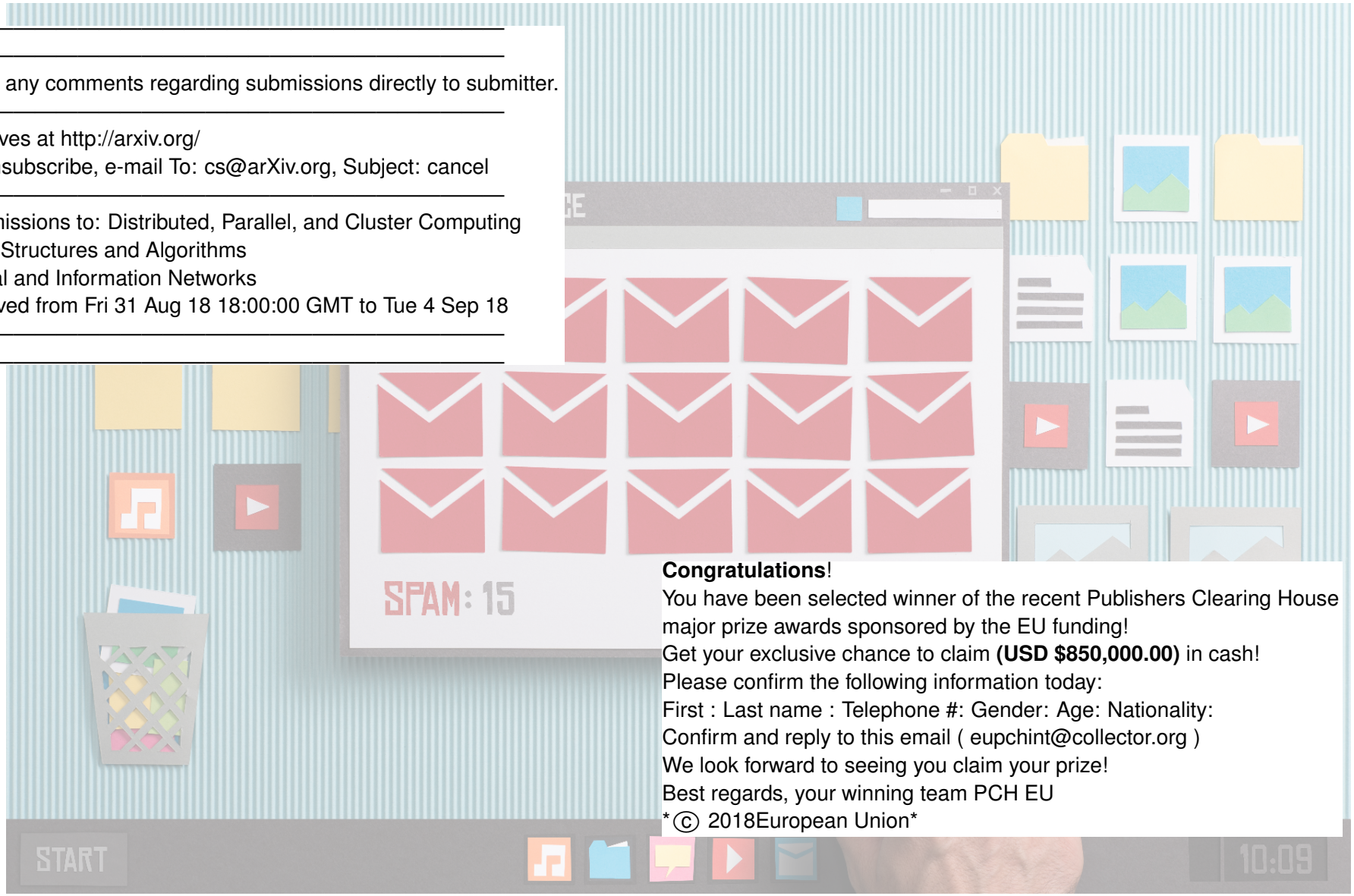


Binary Classification Problem

Send any comments regarding submissions directly to submitter.

Archives at http://arxiv.org/
To unsubscribe, e-mail To: cs@arXiv.org, Subject: cancel

Submissions to: Distributed, Parallel, and Cluster Computing
Data Structures and Algorithms
Social and Information Networks
received from Fri 31 Aug 18 18:00:00 GMT to Tue 4 Sep 18



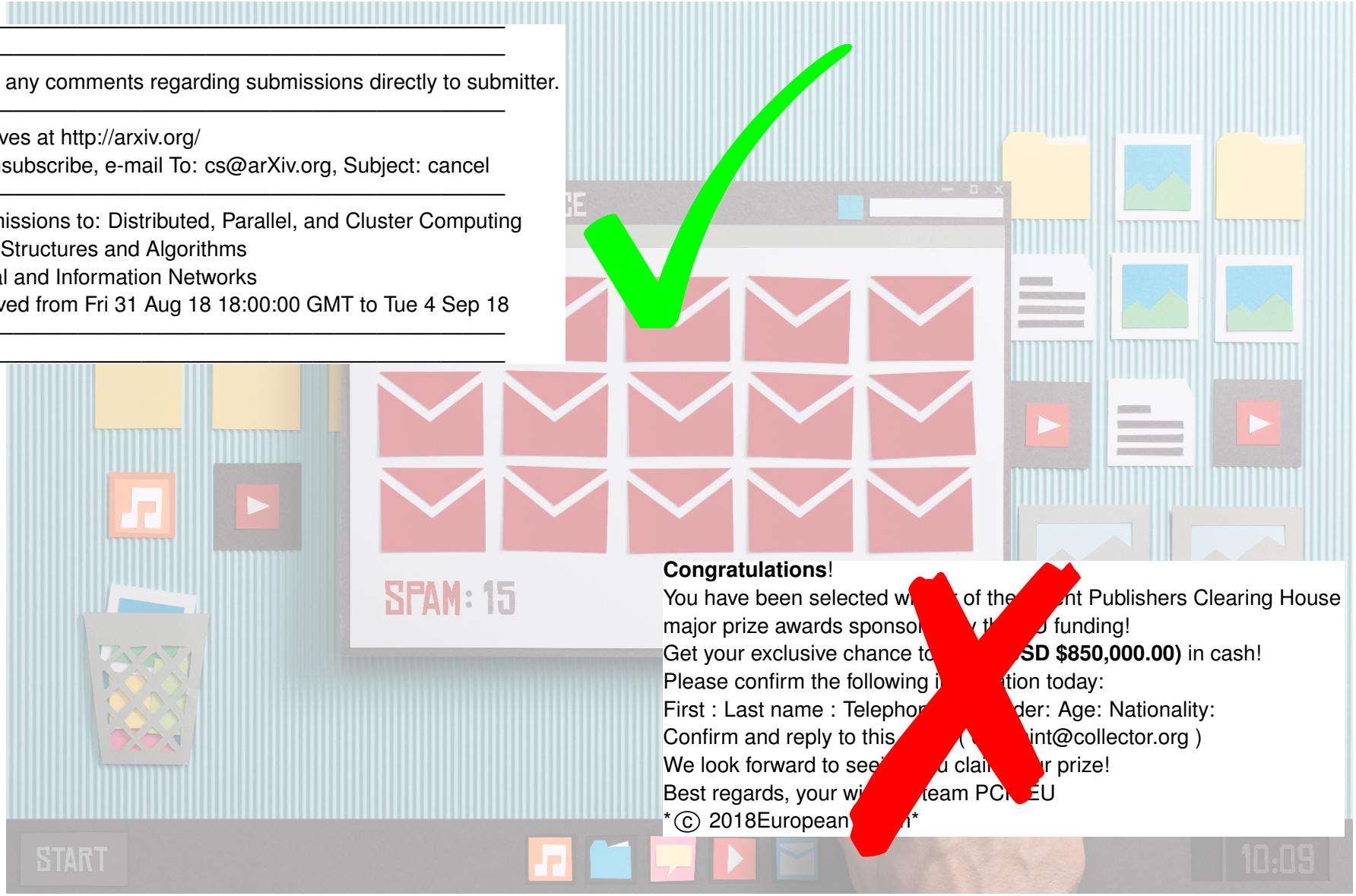
Congratulations!
You have been selected winner of the recent Publishers Clearing House major prize awards sponsored by the EU funding!
Get your exclusive chance to claim **(USD $850,000.00)** in cash!
Please confirm the following information today:
First : Last name : Telephone #: Gender: Age: Nationality:
Confirm and reply to this email (eupchint@collector.org)
We look forward to seeing you claim your prize!
Best regards, your winning team PCH EU
ⓒ 2018European Union

Binary Classification Problem

Send any comments regarding submissions directly to submitter.

Archives at http://arxiv.org/
To unsubscribe, e-mail To: cs@arXiv.org, Subject: cancel

Submissions to: Distributed, Parallel, and Cluster Computing
Data Structures and Algorithms
Social and Information Networks
received from Fri 31 Aug 18 18:00:00 GMT to Tue 4 Sep 18



Binary Classification Problem

Train classifier on n **labeled** data points

$$(x_i, y_i)$$

Data point $x \in \mathbb{R}^d$

Features:

words, text patterns, sender, ...

Label $y \in \{-1, +1\}$

Result:

+1: spam

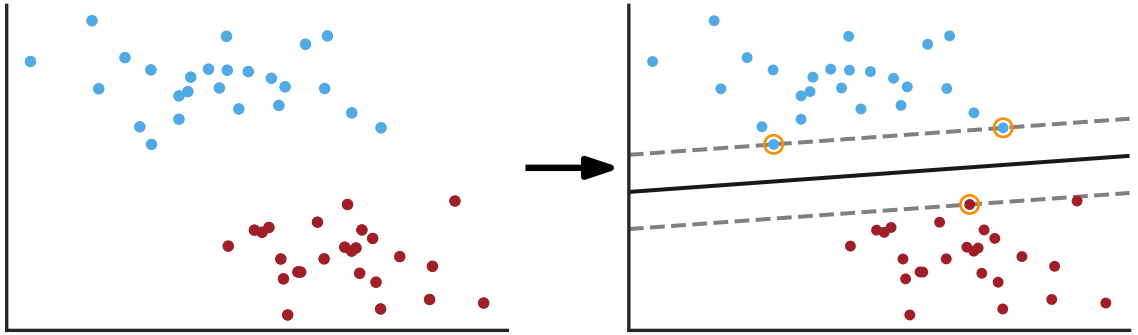
−1: no spam

Goal: Assign label y_{n+1} to new data points x_{n+1}

Support Vector Machines [CV'97]

Find **hyperplane** with **maximum margin** between classes C^- & C^+

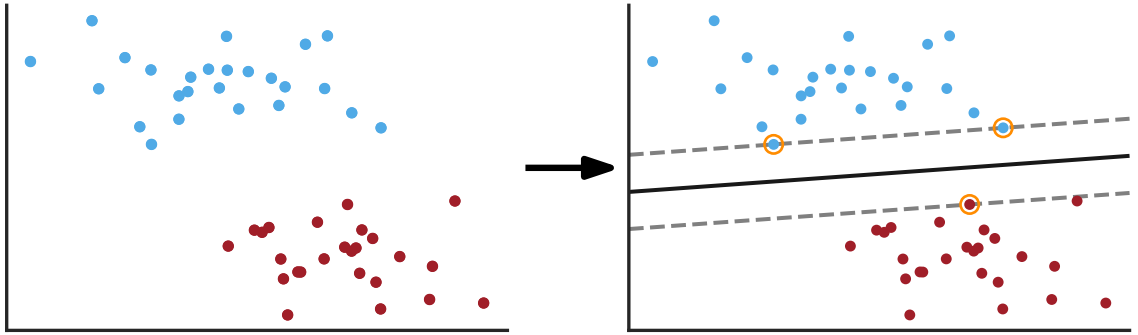
Linear SVM



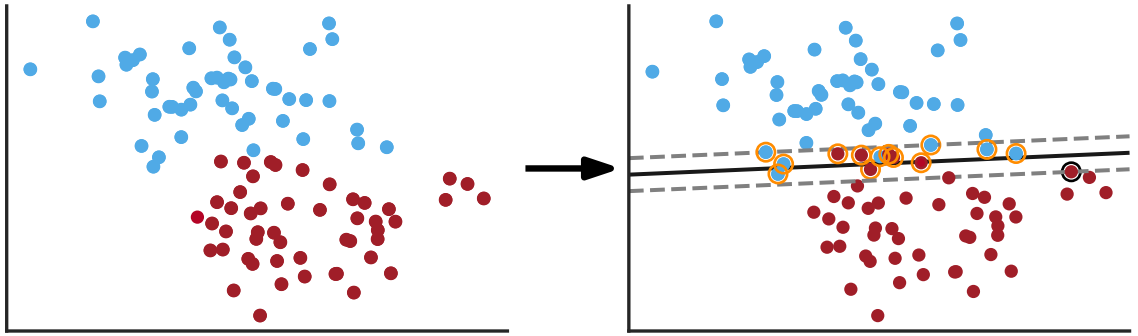
Support Vector Machines [CV'97]

Find **hyperplane** with **maximum margin** between classes C^- & C^+

Linear SVM



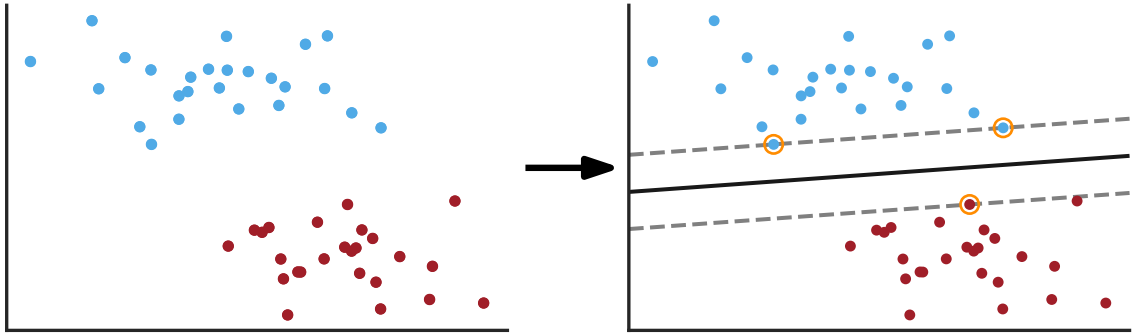
Soft Margin SVM



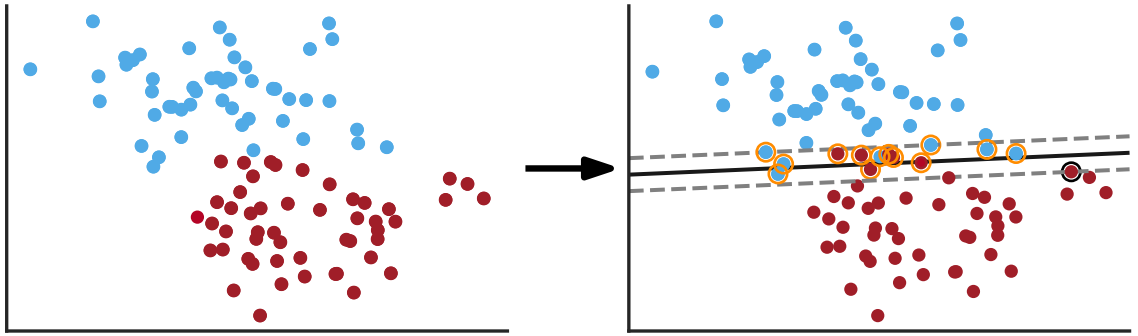
Support Vector Machines [CV'97]

Find **hyperplane** with **maximum margin** between classes C^- & C^+

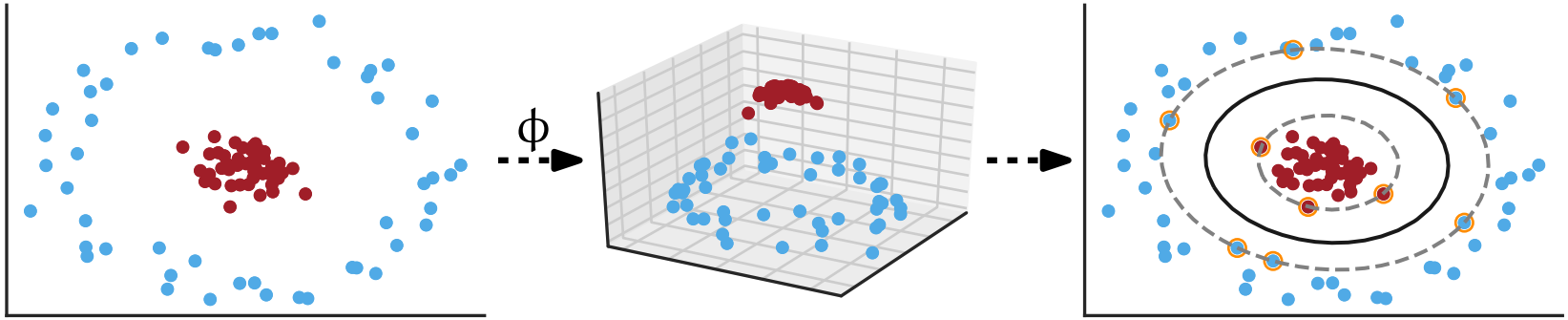
Linear SVM



Soft Margin SVM



Nonlinear SVM

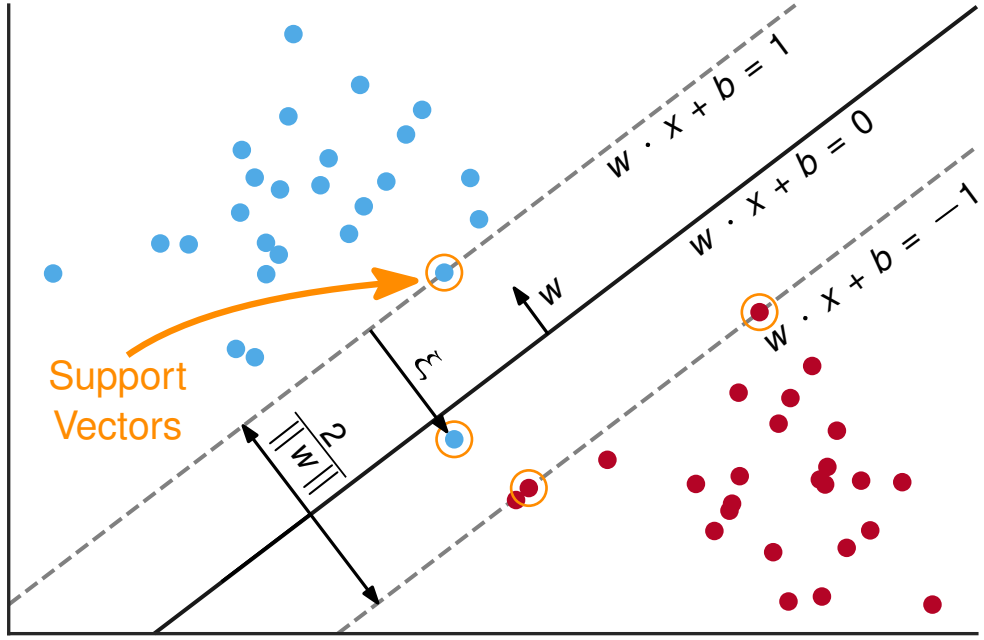


Support Vector Machines [CV'97]

SVM Optimization Problem:

$$\text{minimize} \qquad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \qquad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$



Support Vector Machines [CV'97]

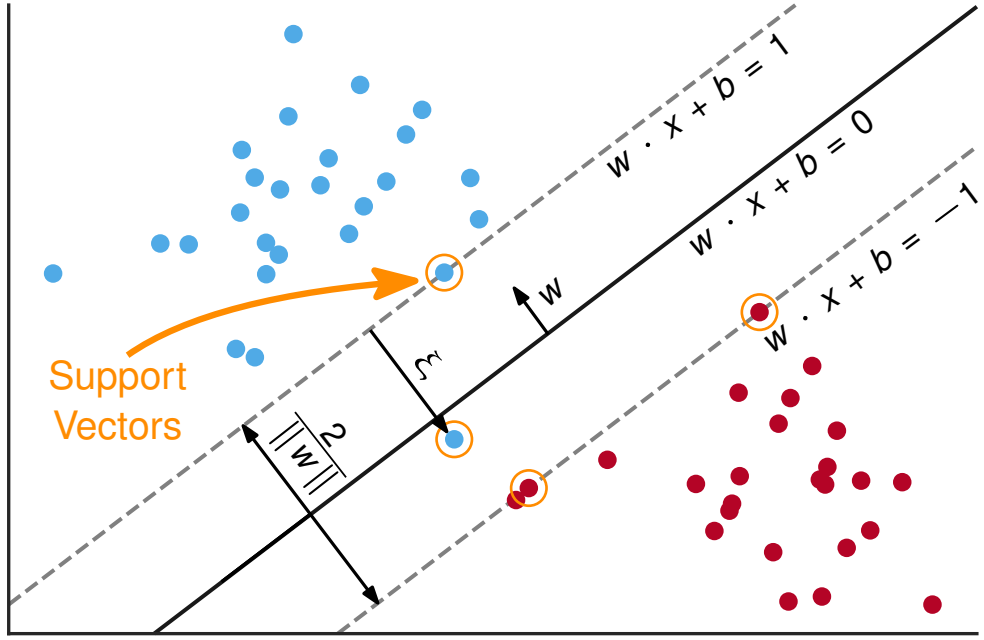
SVM Optimization Problem:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

Slack penalty

$$\text{subject to} \quad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Slack variables



Support Vector Machines [CV'97]

SVM Optimization Problem:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

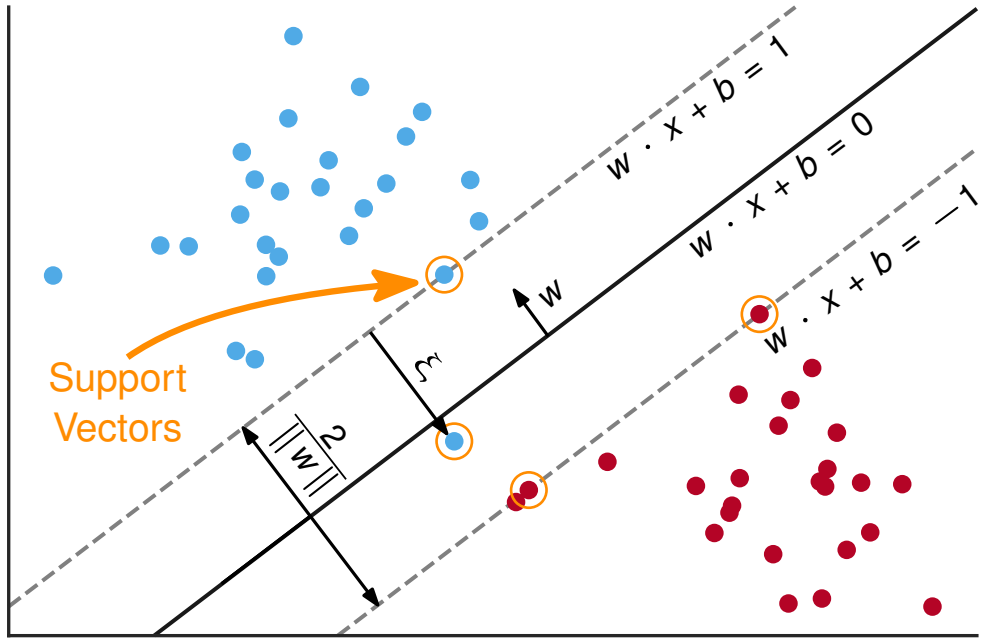
Slack penalty

$$\text{subject to} \quad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Mapping to higher dimensional space

$$\phi : \mathbb{R}^d \to \mathbb{R}^p (d \leq p)$$

Slack variables



Support Vector Machines [CV'97]

SVM Optimization Problem:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

Slack penalty

$$\text{subject to} \quad y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Mapping to higher dimensional space

$$\phi : \mathbb{R}^d \to \mathbb{R}^p (d \leq p)$$

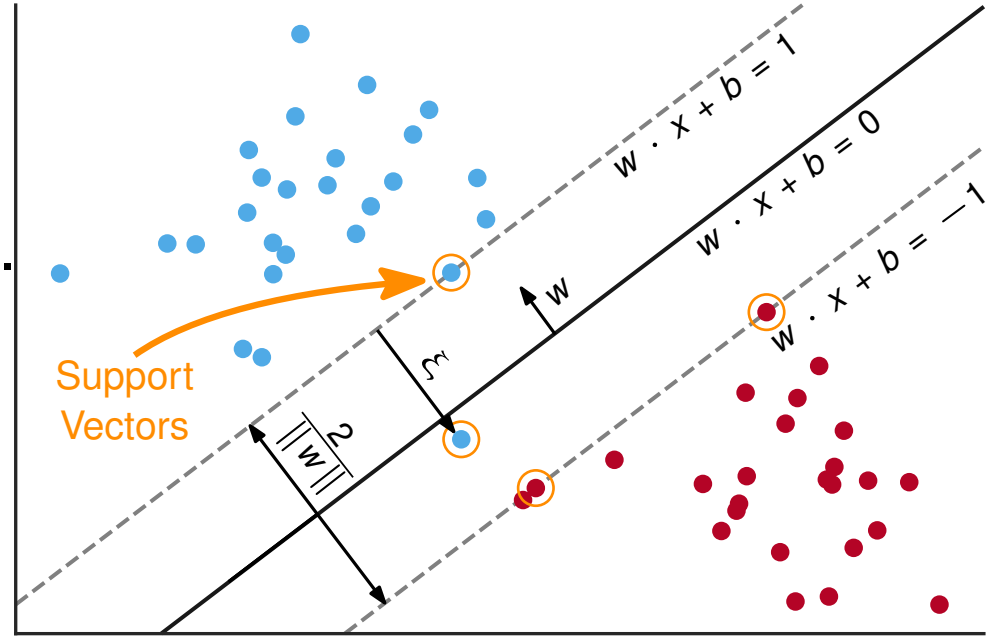
Slack variables

via ...

Kernel Trick:

- Replace inner product by kernel fct.
- $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$
- Here: $k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$

Gaussian kernel (RBF)



Support Vector Machines – Training

Model Selection: instance-specific tuning of several parameters

- Slack penalty C
- Kernel parameters (here: γ)

Complexity:

- Solver running time between $\mathcal{O}\left(n^2\right)$ & $\mathcal{O}\left(n^3\right)$ [GCBDV'04]
- Model selection $\rightsquigarrow$ train many models

Support Vector Machines – Training

Model Selection: instance-specific tuning of several parameters

- Slack penalty C
- Kernel parameters (bandwidth γ)

Training on **large** data sets becomes **infeasible**!

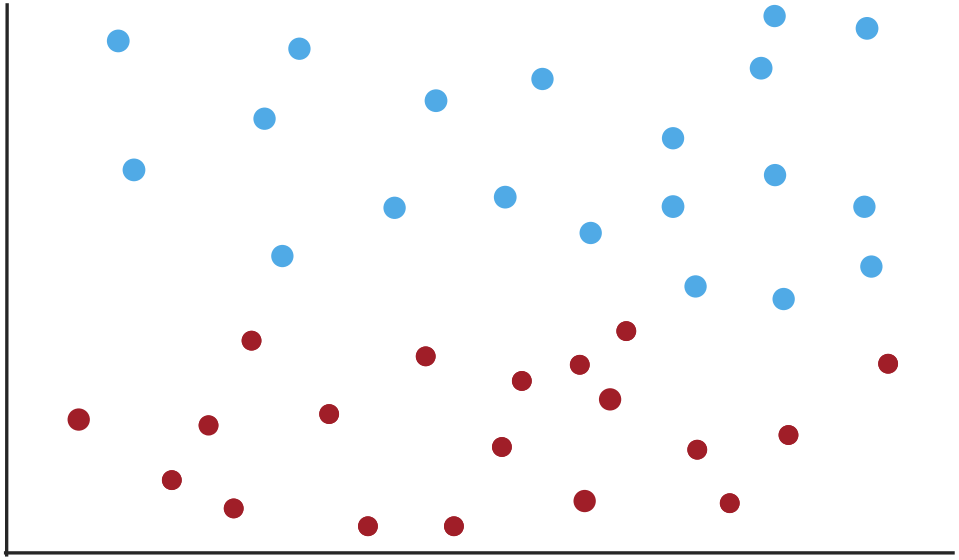
Complexity:

- Solver running time between $\mathcal{O}(n^2)$ & $\mathcal{O}(n^3)$ [GCBDV'04]
- Model selection $\rightsquigarrow$ train many models

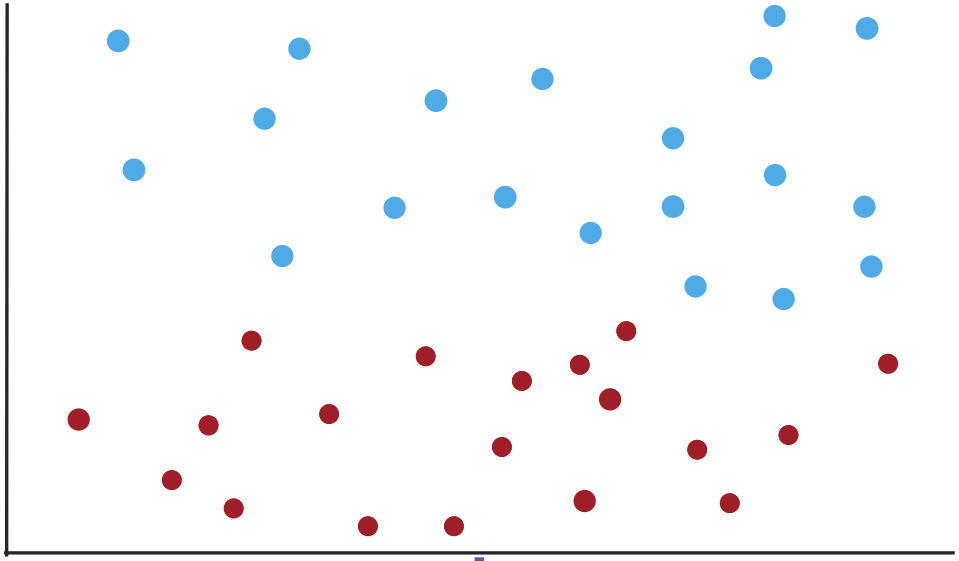
Performance Improvements:

- Sampling [SC'00]
- Parallelization [CZWBLQC'07,ZCWZC'09,CWLTP'17]
- Hierarchical techniques
  - Input space [YYH'03,HSCKCLP'11,HSD'14]
  - Graph representation [RS'15,SJKLRS'17]

From Feature Vectors to Graphs

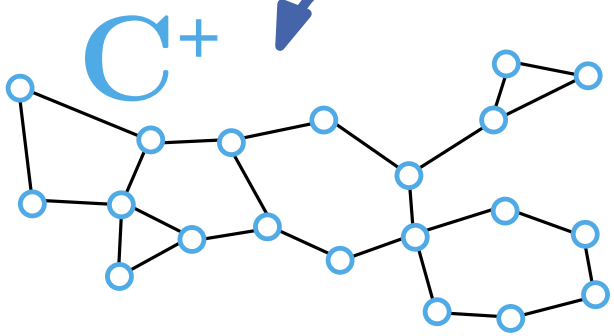


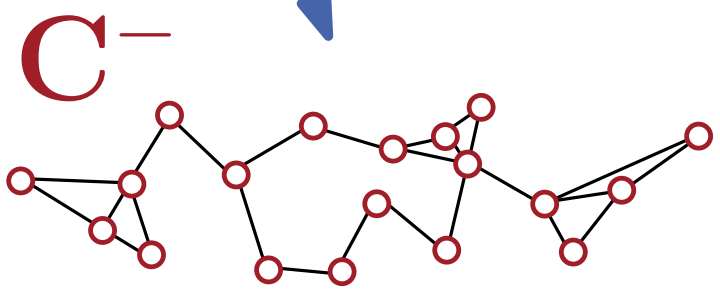
From Feature Vectors to Graphs



Approx. **k-nearest** neighbors

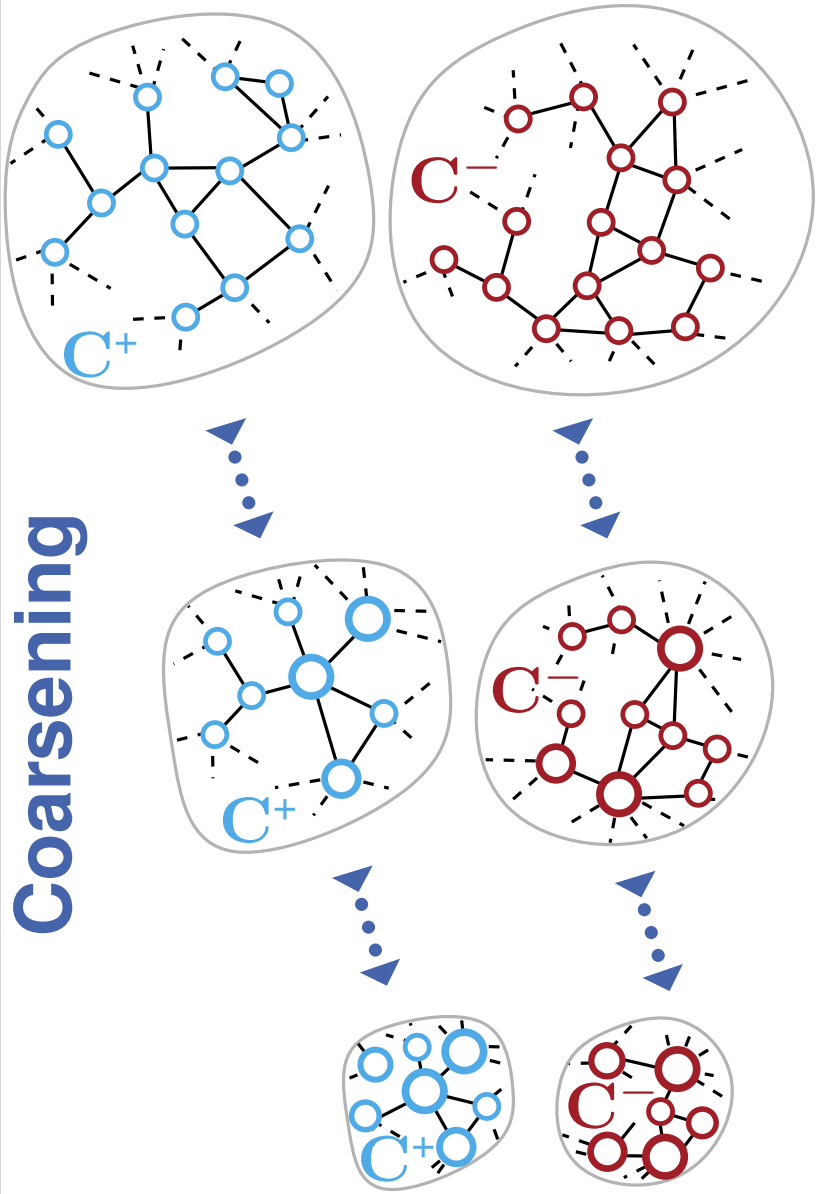
[RS'15]



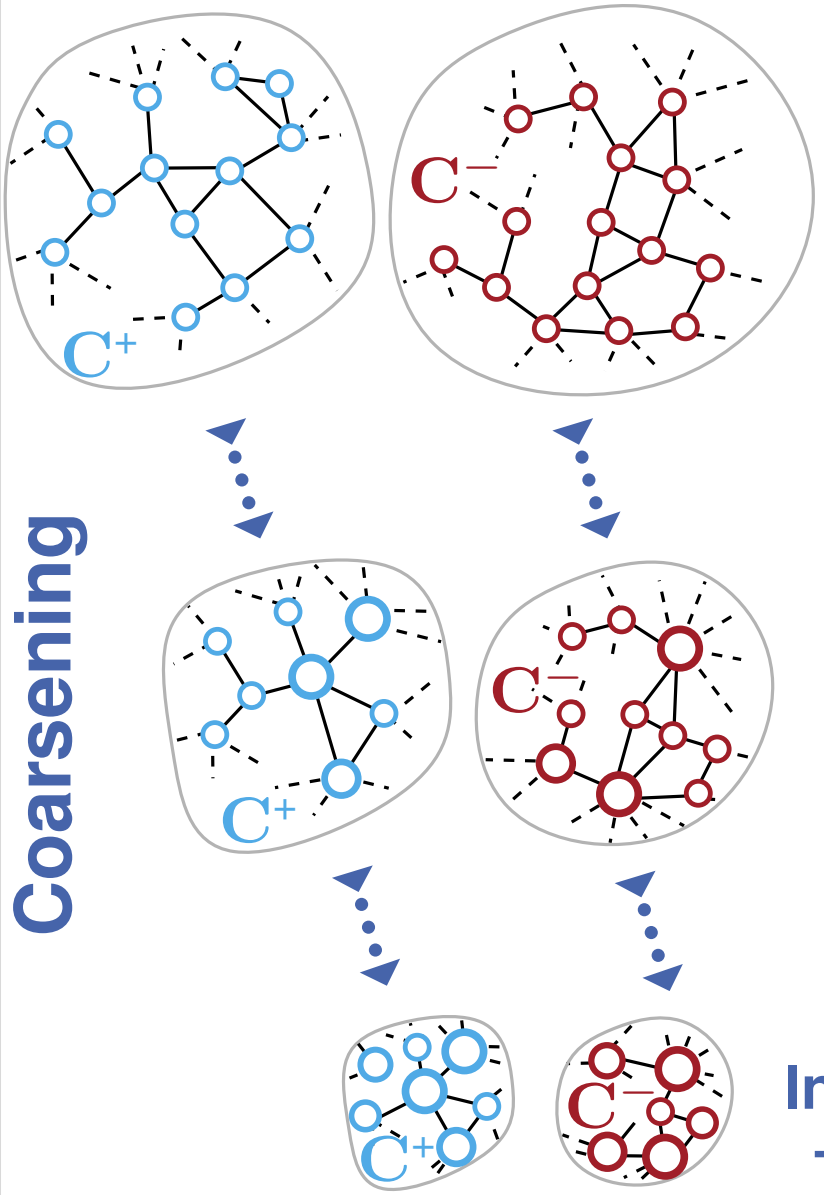


$\omega(e) = \frac{1}{\text{dist}(p,q)} \Rightarrow$ encode **proximity** information into **edge weights**

Multilevel Support Vector Machines [RS'15]

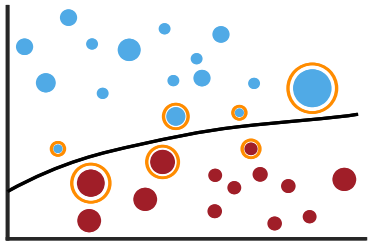


Multilevel Support Vector Machines [RS'15]

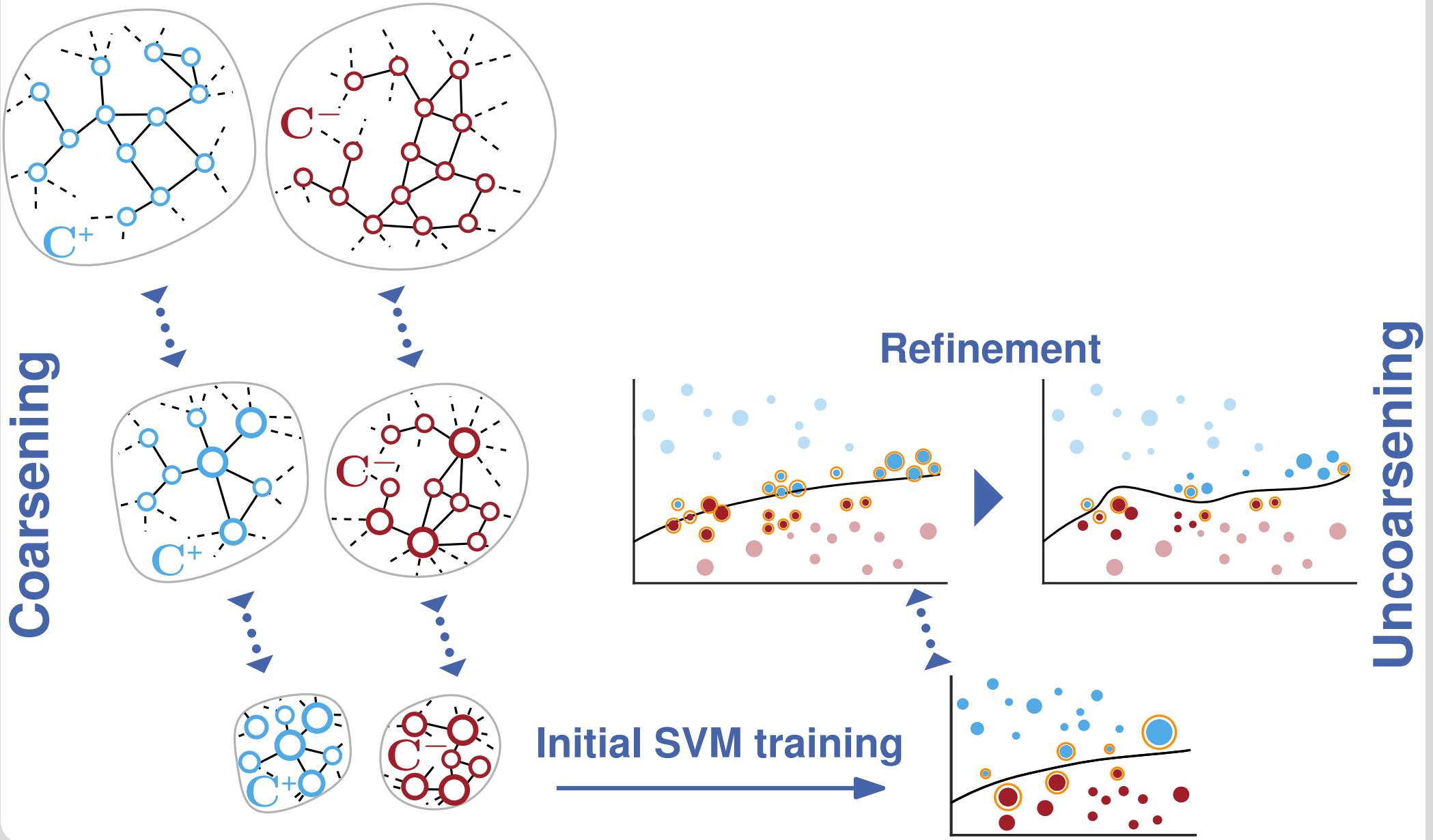


Coarsening

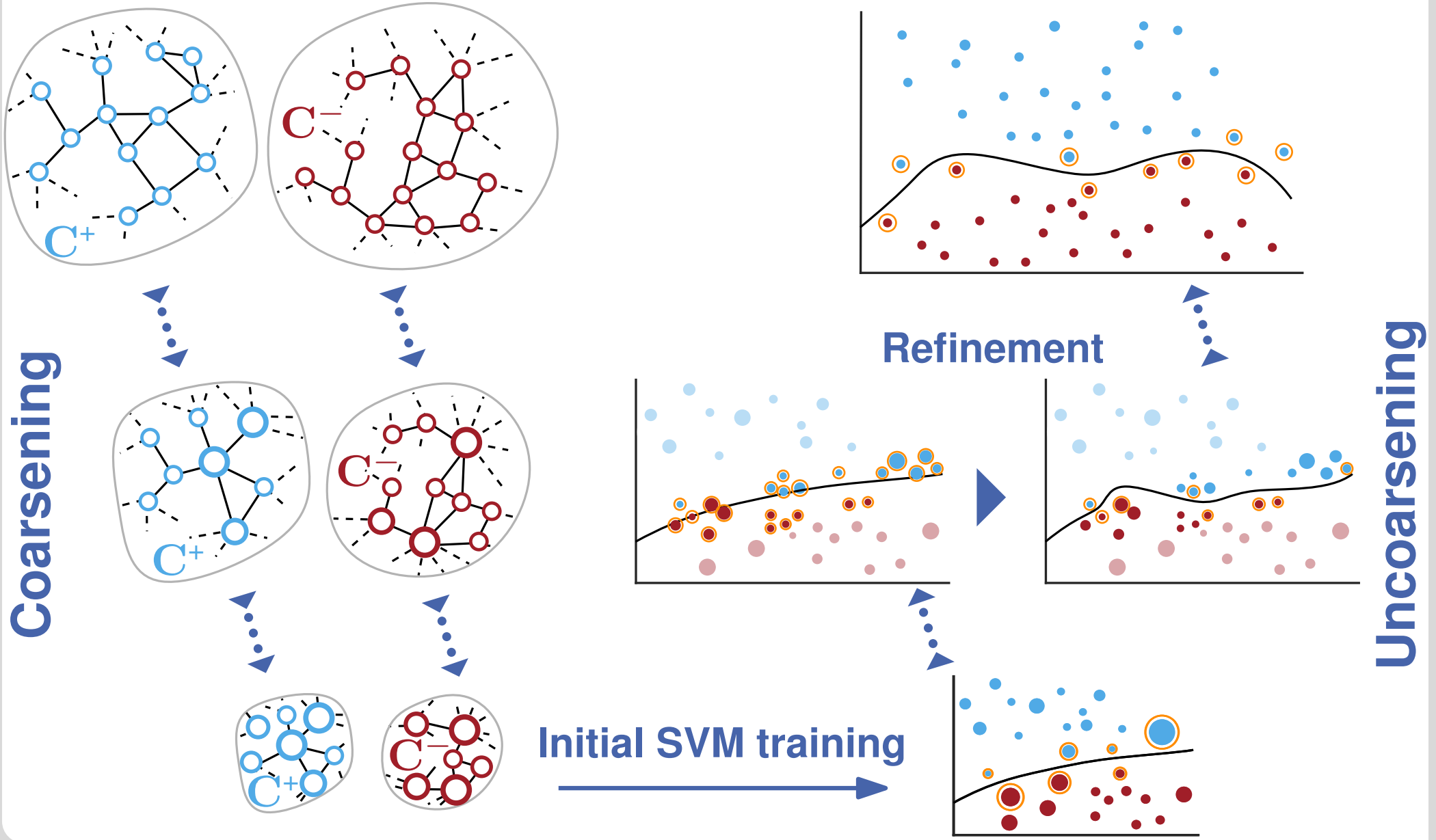
Initial SVM training



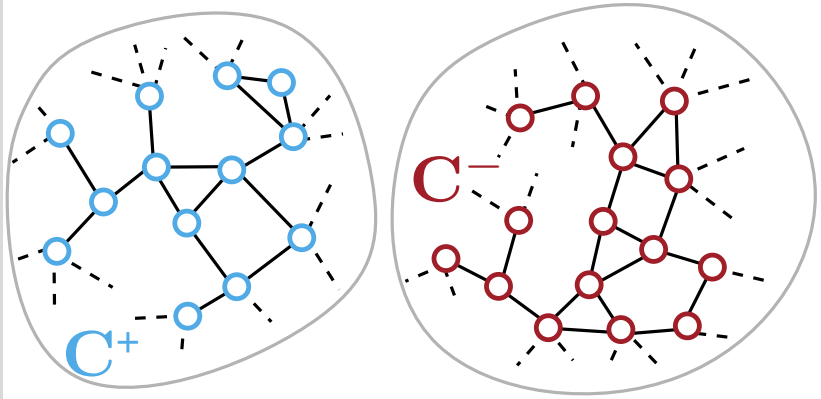
Multilevel Support Vector Machines [RS'15]



Multilevel Support Vector Machines [RS'15]

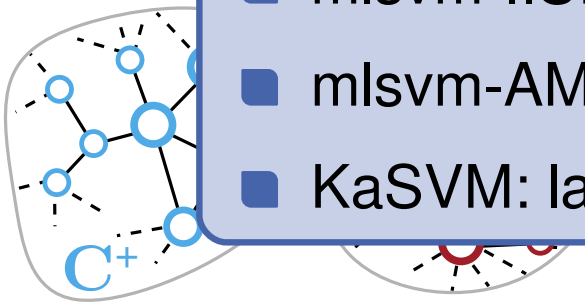


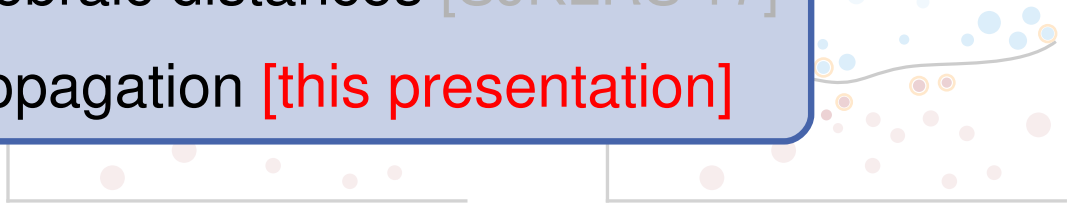
Multilevel Support Vector Machines [RS'15]



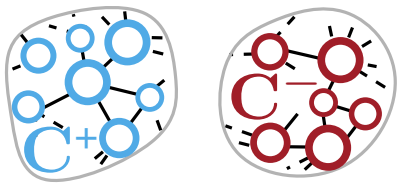
Coarsening

- mlsvm-IIS: independent sets [RS'15]
- mlsvm-AMG: algebraic distances [SJKLRS'17]
- KaSVM: label propagation [this presentation]





Uncoarsening



Initial SVM training

KaSVM - Algorithm Outline

Algorithm 1: KaSVM

`preprocess` data
`build` k-nearest neighbor graphs for C^+ and C^-
`contract` graphs recursively, build hierarchy
`train` initially on coarsest problem
while *levels in the hierarchy* **do**
    `train` model on uncontracted support vectors of prev. level
return *best model of all levels*

KaSVM - Algorithm Outline

Algorithm 1: KaSVM

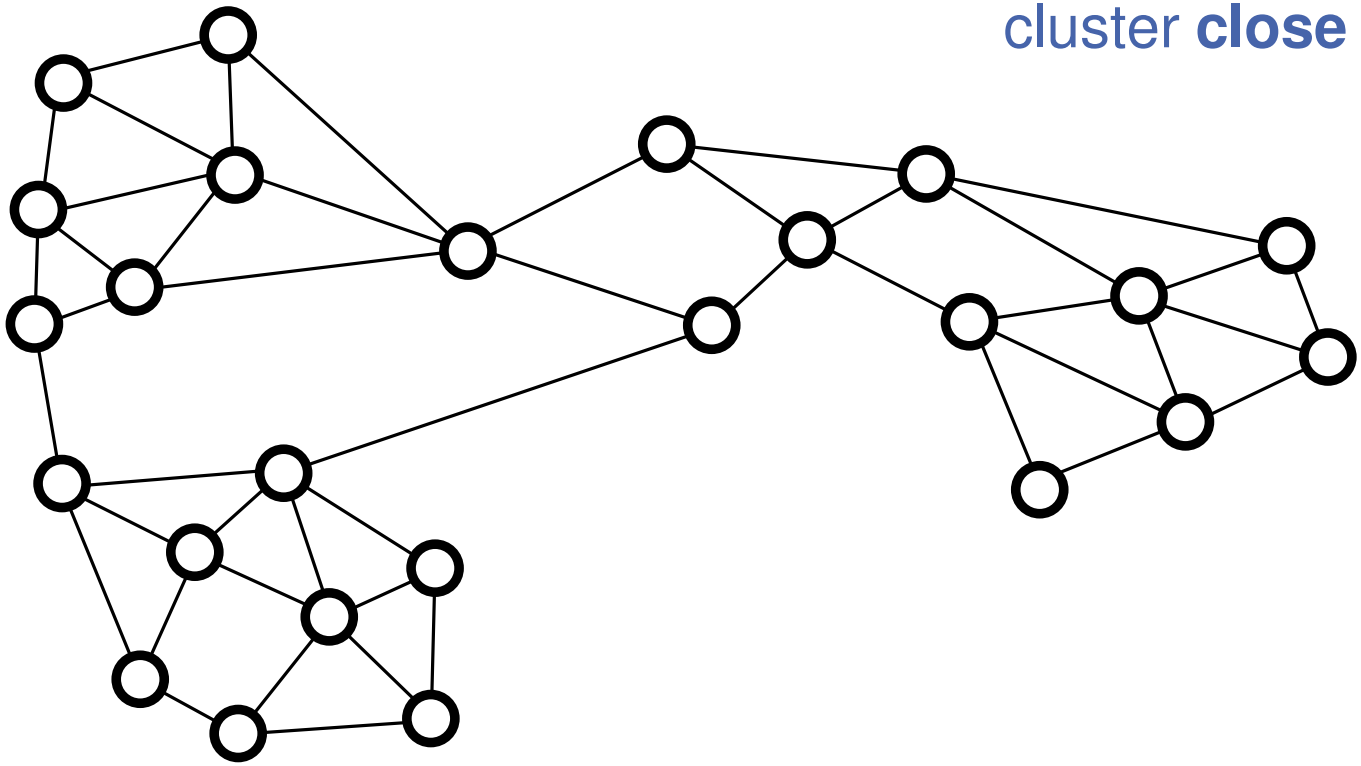
`preprocess` data
`build` k-nearest neighbor graphs for C^+ and C^-
`contract` graphs recursively, build hierarchy
`train` initially on coarsest problem
while *levels in the hierarchy* **do**
    `train` model on uncontracted support vectors of prev. level
return *best model of all levels*

Label Propagation [RAK'07]

Cut-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster V_i having **strongest** connection
  $\Rightarrow c[v] = \text{argmax}_{V_i} \omega(\{(v,u) | u \in N(v) \cap V_i\})$

cluster **close** nodes

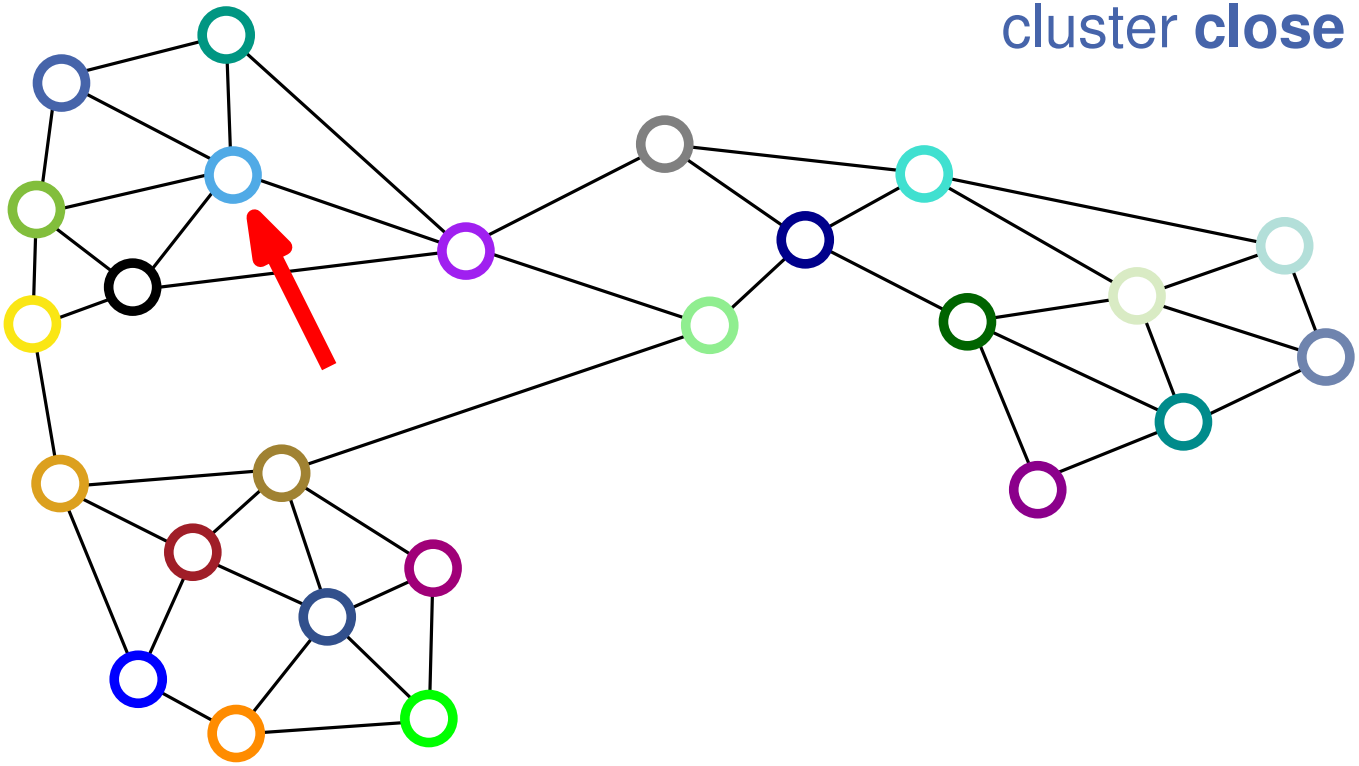


Label Propagation [RAK'07]

Cut-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster V_i having **strongest** connection
  $\Rightarrow c[v] = \text{argmax}_{V_i} \omega(\{(v,u) | u \in N(v) \cap V_i\})$

cluster **close** nodes

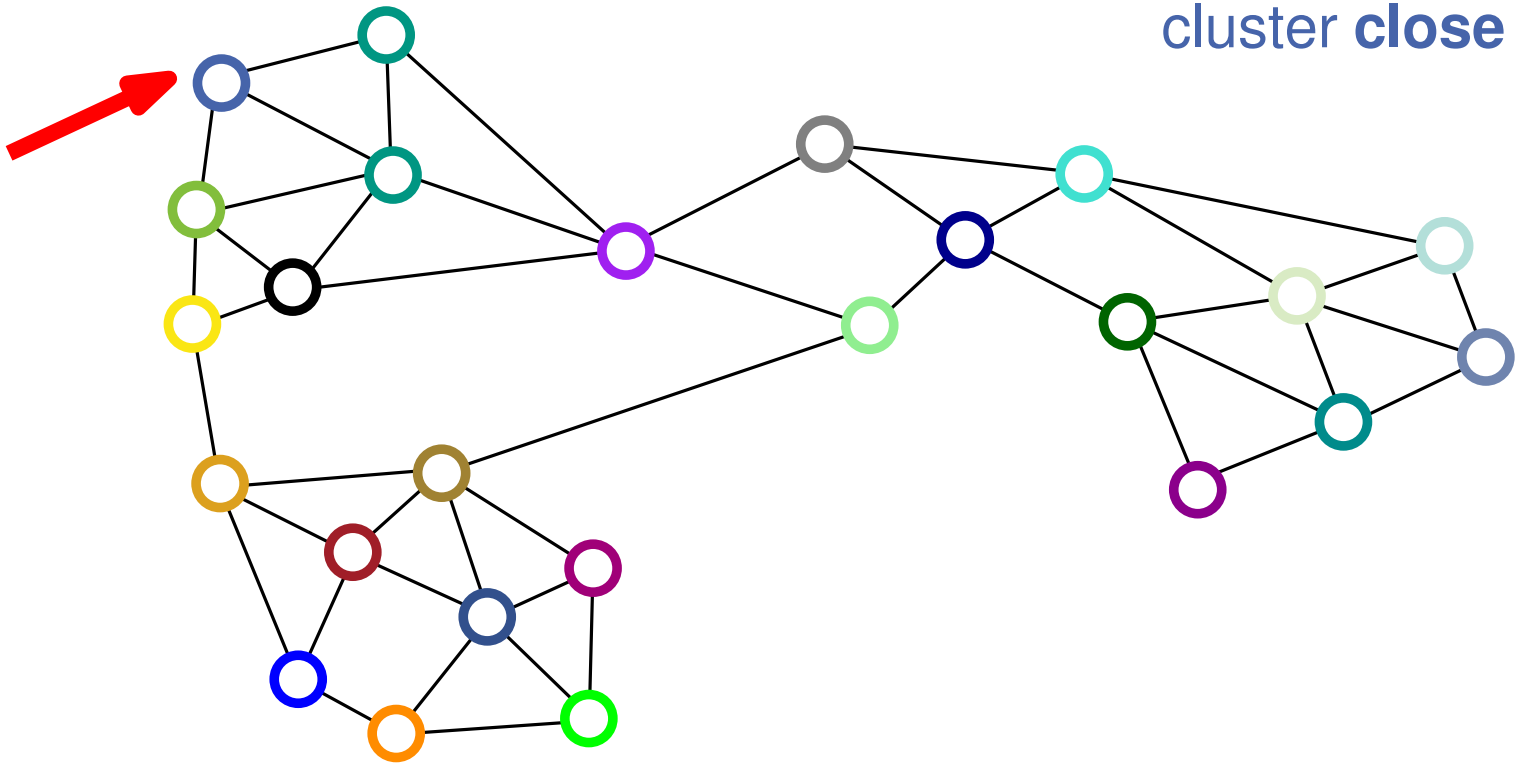


Label Propagation [RAK'07]

Cut-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster V_i having **strongest** connection
  $\Rightarrow c[v] = \text{argmax}_{V_i} \omega(\{(v,u) | u \in N(v) \cap V_i\})$

cluster **close** nodes

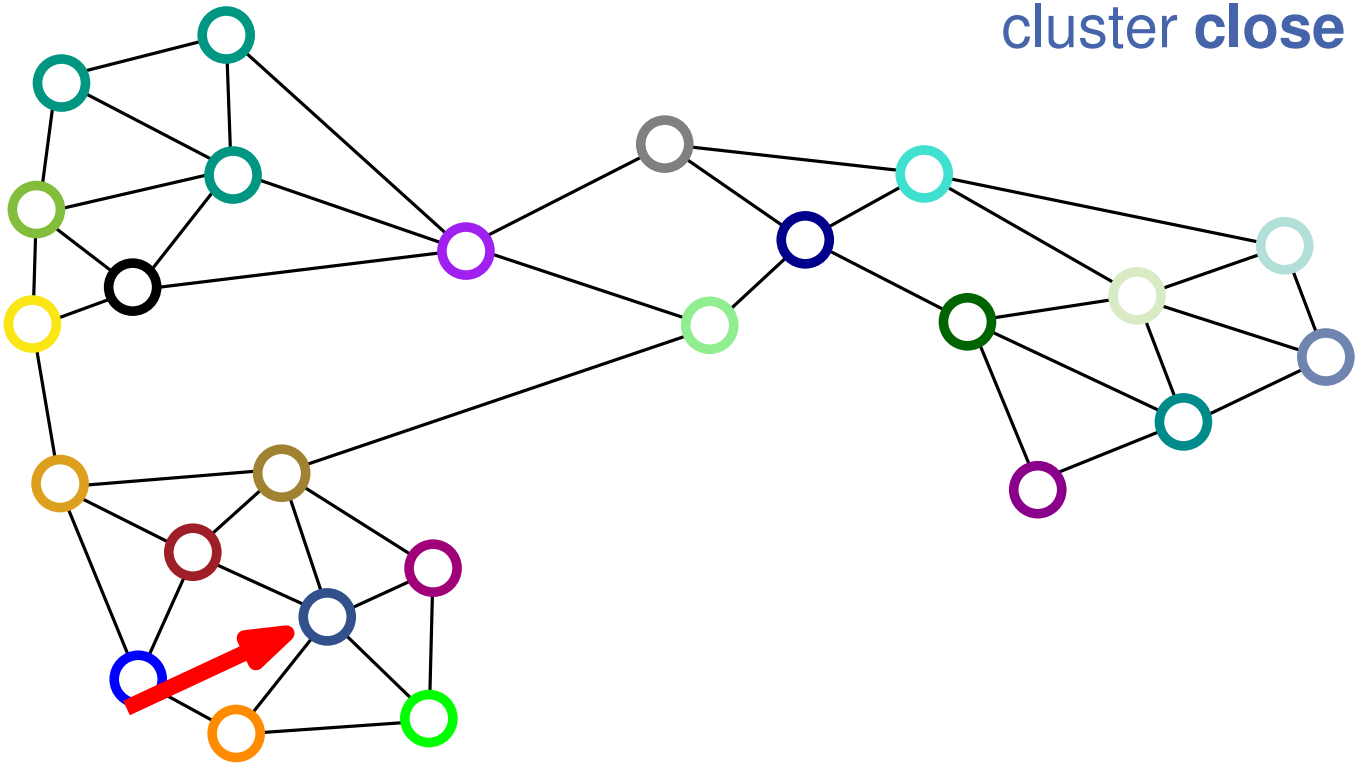


Label Propagation [RAK'07]

Cut-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster V_i having **strongest** connection
  $\Rightarrow c[v] = \text{argmax}_{V_i} \omega(\{(v,u) | u \in N(v) \cap V_i\})$

cluster **close** nodes

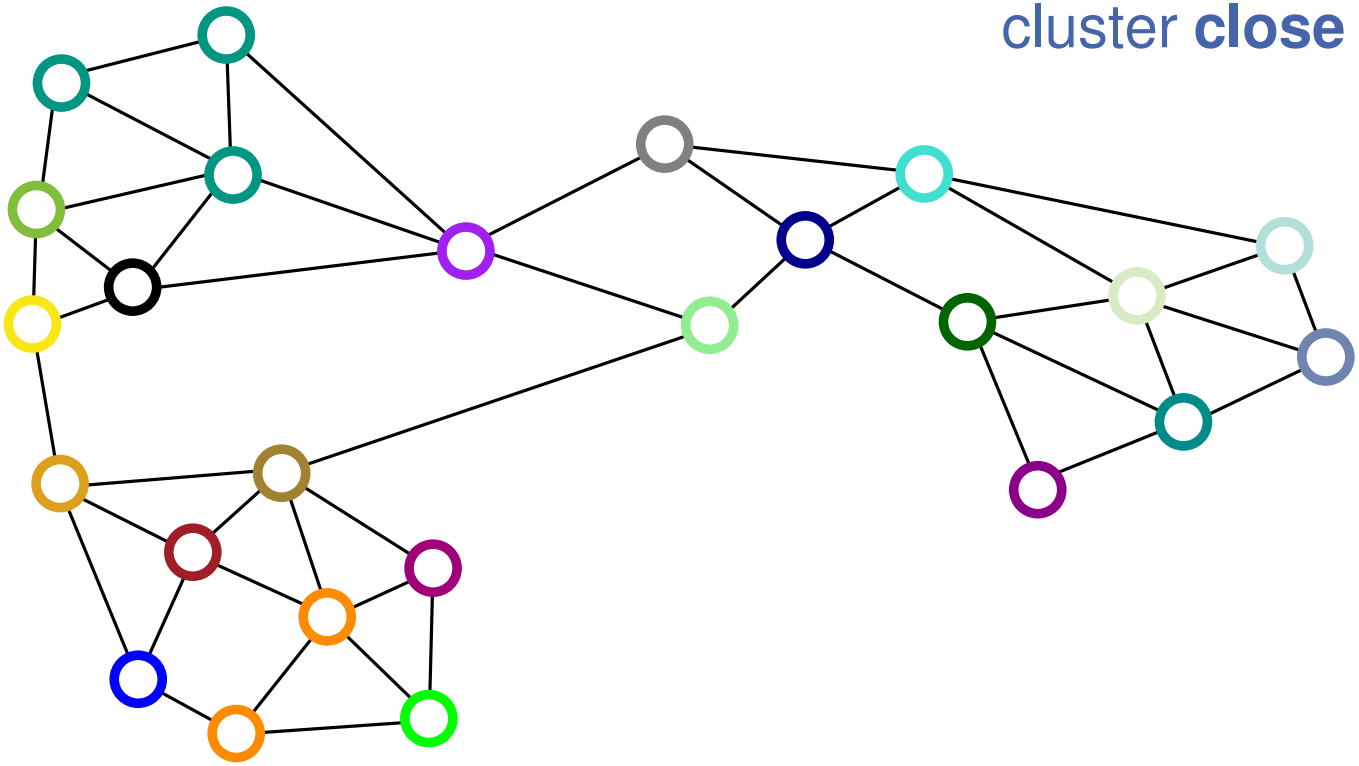


Label Propagation [RAK'07]

Cut-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster V_i having **strongest** connection
  $\Rightarrow c[v] = \text{argmax}_{V_i} \omega(\{(v,u) | u \in N(v) \cap V_i\})$

cluster **close** nodes

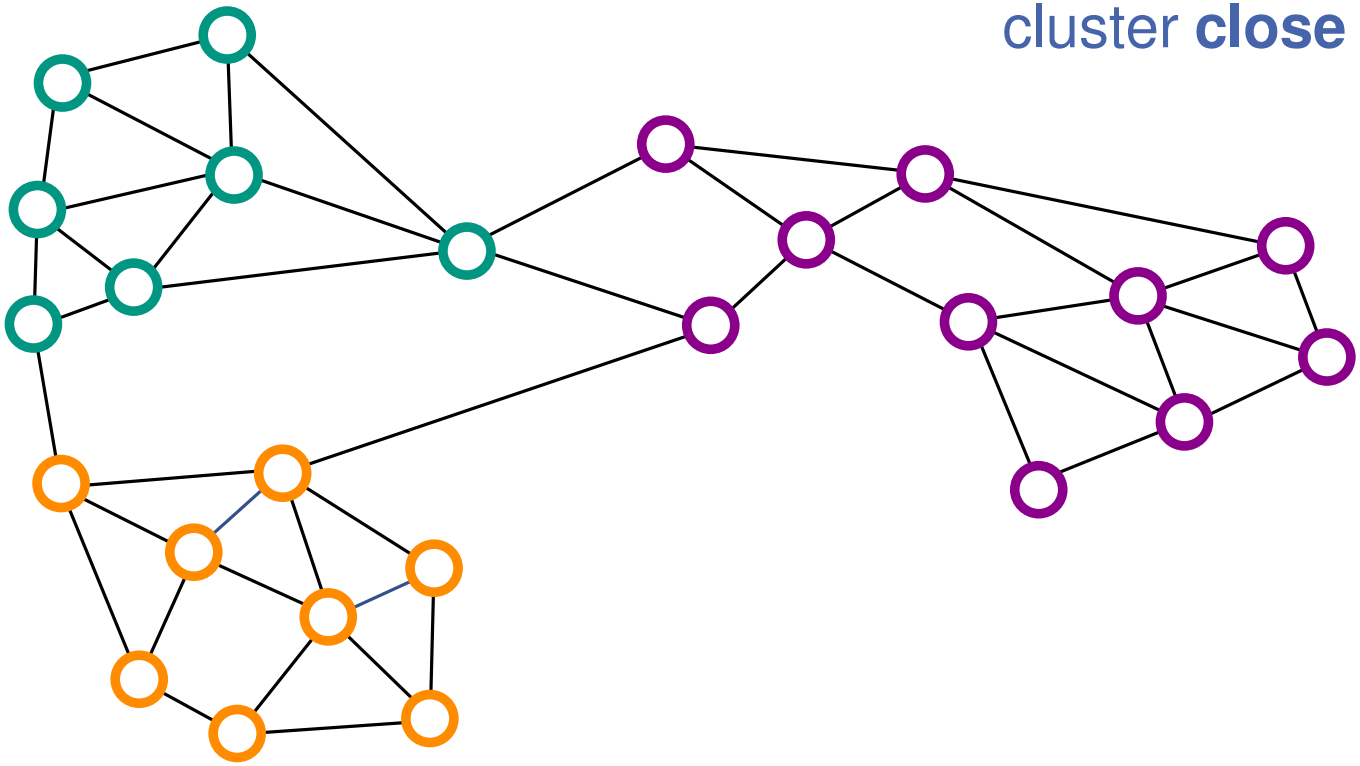


Label Propagation [RAK'07]

Cut-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster V_i having **strongest** connection
  $\Rightarrow c[v] = \text{argmax}_{V_i} \omega(\{(v,u) | u \in N(v) \cap V_i\})$

cluster **close** nodes

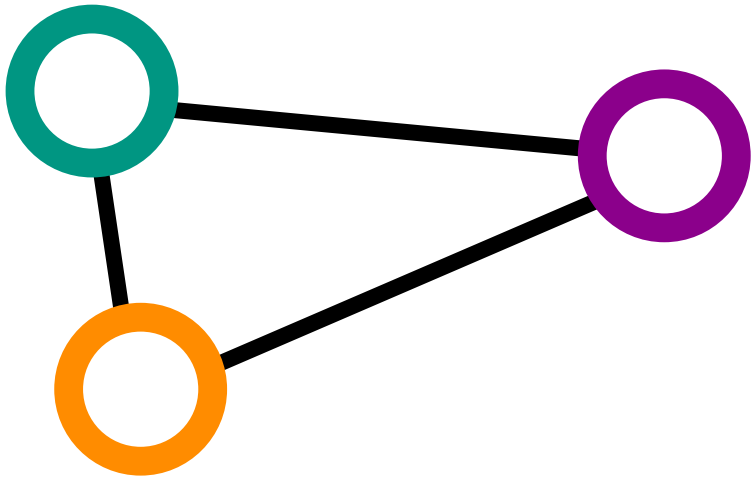


Label Propagation [RAK'07]

Cut-based, **linear** time clustering algorithm

- Start with singletons
- Traverse nodes in random order or smallest degree first
- Move to cluster V_i having **strongest** connection
  $\Rightarrow c[v] = \text{argmax}_{V_i} \omega(\{(v,u) | u \in N(v) \cap V_i\})$

cluster **close** nodes

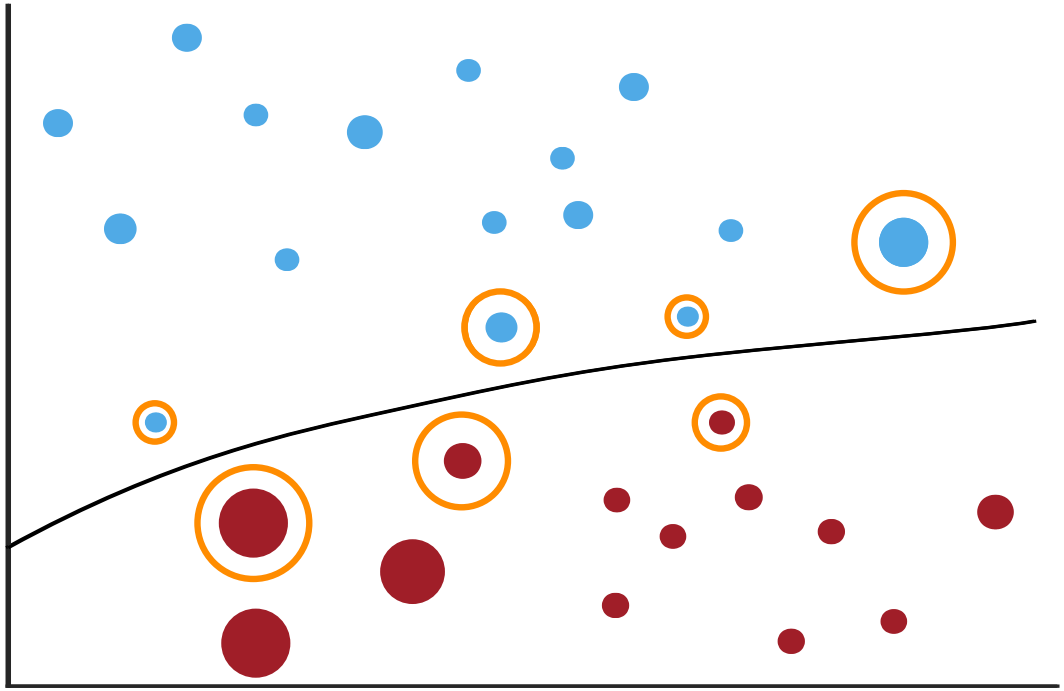


$\Rightarrow$ average feature values of clustered nodes: $x_{\text{coarse}} = \frac{1}{c(V_i)} \sum_{v_i \in V_i} c(v_i) x_i$

Initial Training

Train on **coarsest** problem

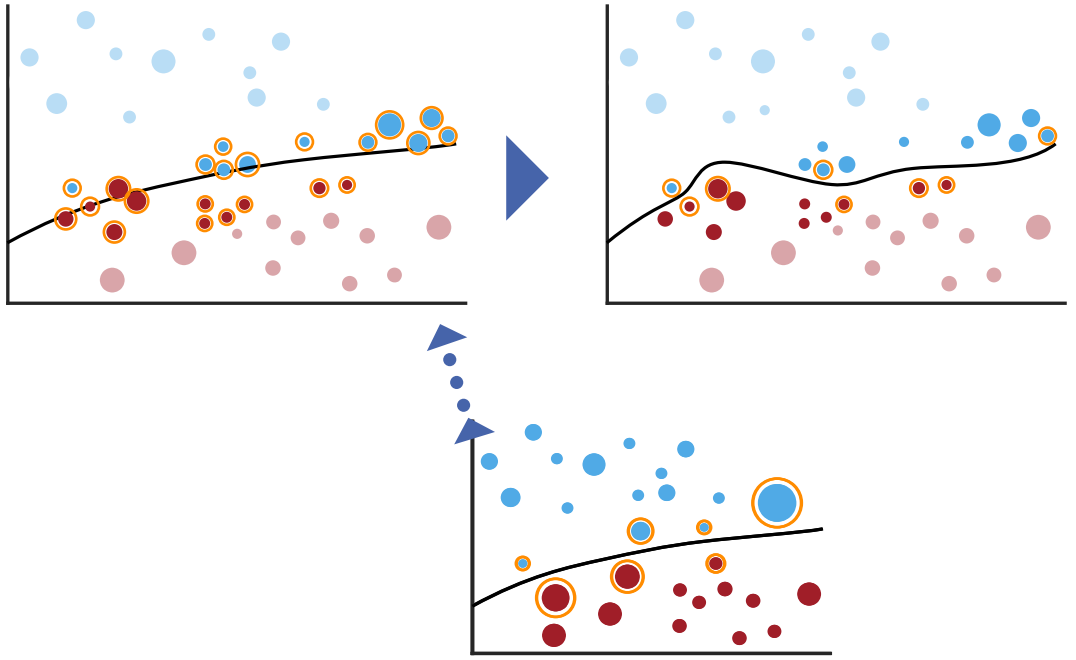
- Model selection (C, γ) via uniform design (UD) [HLLH'07]
- Solver: LibSVM
- Validation on random 10% of training set



Uncoarsening/Refinement

Try to improve model

- Uncontract support vectors
- Ensure **similar** size for C^- & C^+
- **Reuse** C, γ from prev. level $\Rightarrow$ 2nd UD sweep around old params.
- Validation on random 10% of training set



Experimental Setup

Machine: AMD Opteron 6168 with 1.9 GHz, 256 GB of RAM

Implementation:

- approx. k-nearest neighbors: FLANN 1.8.4 [ML'09]
- SVM training: LibSVM 3.22

Configuration:

- $k = 10$ nearest neighbors
- $\ell = 10$ label propagation iterations
- stop coarsening $\left|C^{+/-}\right| \approx 500$ nodes

Algorithms:

- KaSVM / KaSVM$_{\text{fast}}$
- mlsvm-AMG (outperforms DC-SVM & EnsembleSVM) [SJKLRS'17]
- LibSVM

Experimental Setup

Machine: AMD Opteron 6168 with 1.9 GHz, 256 GB of RAM

Implementation:

- approx. k-nearest neighbors: FLANN 1.8.4 [ML'09]
- SVM training: LibSVM 3.22

Configuration:

- $k = 10$ nearest neighbors
- $\ell = 10$ label propagation iterations
- stop coarsening $|C^{+/-}| \approx 500$ nodes

Algorithms:

- KaSVM / KaSVM$_{\text{fast}}$ ← initially trained model / **no** refinement
- mlsvm-AMG (outperforms DC-SVM & EnsembleSVM) [SJKLRS'17]
- LibSVM

Instances

mlsvm-AMG [SJKLRS'17]

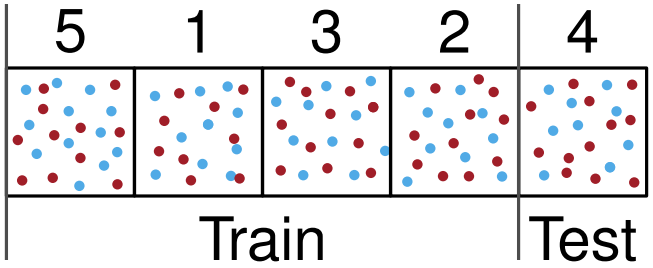
UCI Machine Learning Repository

Name	Size	Feat.	C^+	C^-
Advertisement	3 279	1 558	459	2 820
Buzz	140 707	77	27 775	112 932
Clean (Musk)	6 598	166	1 017	5 581
Cod-rna	59 535	8	19 845	39 690
EEG Eye State	14 980	14	6 723	8 257
Forest (Class 3)	581 012	54	35 754	369 172
Forest (Class 5)	581 012	54	9 493	571 519
Forest (Class 7)	581 012	54	20 510	560 502
Hypothyroid	3 919	21	240	3 679
Isolet (Class A)	6 919	617	240	5 998
Letter (Class Z)	20 000	16	734	19 266
Nursery	12 960	8	4 320	8 640
Protein	145 751	74	1 296	144 455
Ringnorm	7 400	20	3 664	3 736
Twonorm	7 400	20	3 703	3 697
APS failure	76 000	170	1 375	74 625
Census	299 285	41	18 568	280 717
Letter (Class A)	20 000	16	786	19 266
Letter (Class B)	20 000	16	766	19 266
Letter (Class H)	20 000	16	734	19 266
Skin	245 057	3	50 859	194 198
Sleep (Class 1)	105 908	13	9 052	96 856

Experimental Methodology

k-fold cross validation:

- Shuffle data set $\rightarrow$ split into $k = 5$ parts
- k training repetitions:
  $\Rightarrow$ **Training** set: k-1 parts
  $\Rightarrow$ **Test** set: 1 part
- 5 k-folds per instance



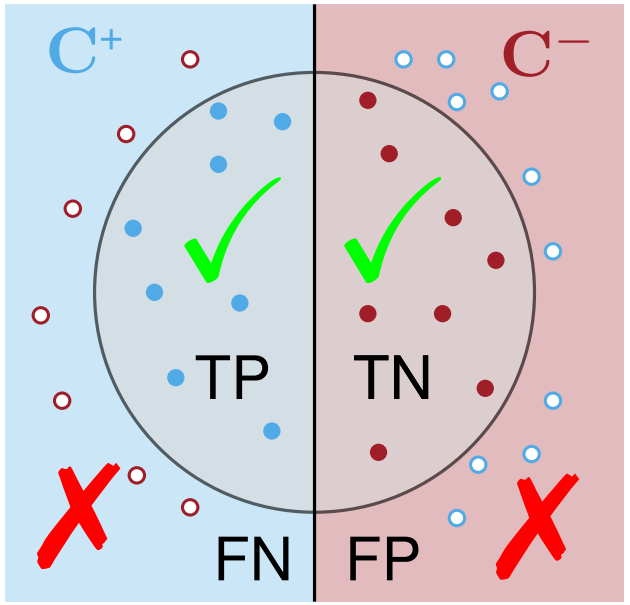
Performance Measures:

$$\text{Accuracy } (ACC) = \frac{TP + TN}{FP + TN + TP + FN}$$

$$\text{Sensitivity } (SN) = \frac{TP}{TP + FN}$$

$$\text{Specificity } (SP) = \frac{TN}{TN + FP}$$

$$\text{Geometric mean} = \sqrt{SP \cdot SN}$$



Running Time

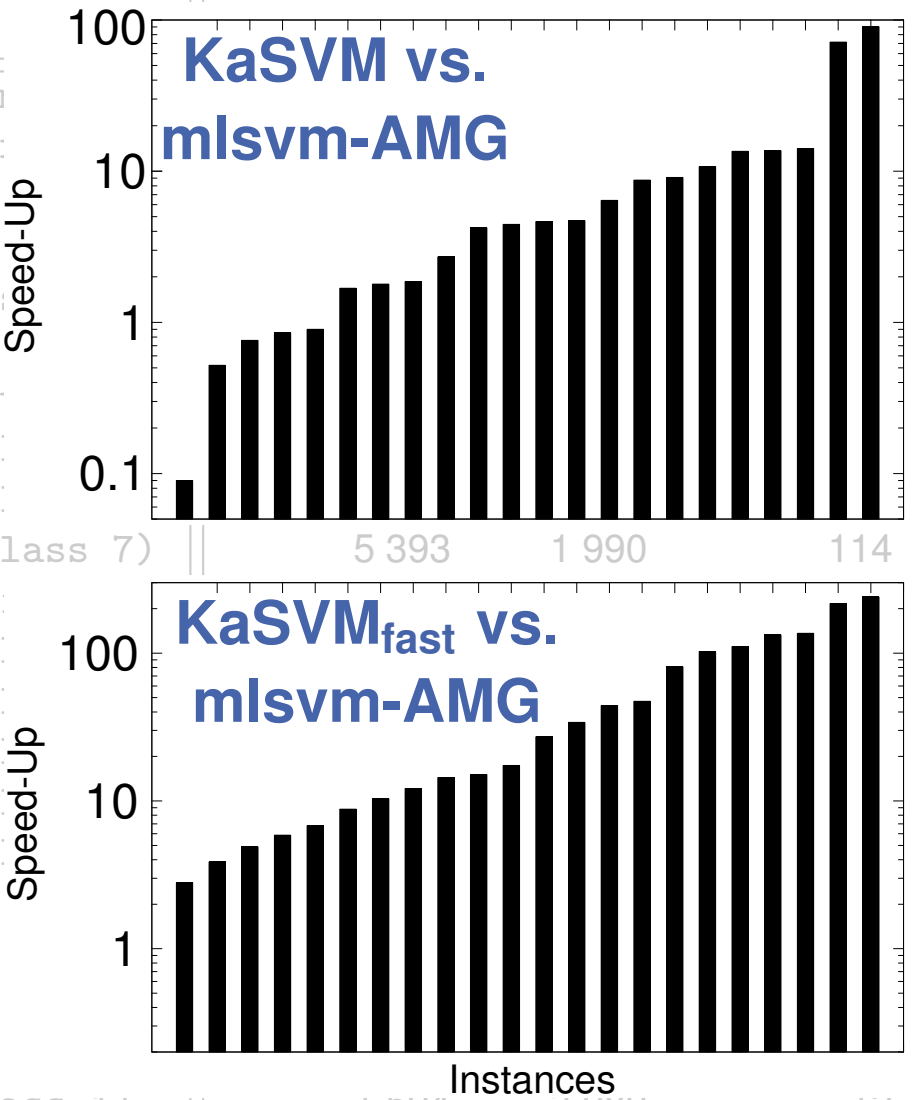
	mlsvm-AMG	KaSVM	$\text{KaSVM}_{\text{fast}}$	LibSVM
	running time [s]			
Advertisement	343	192	70	557
APS failure	1 473	109	13	-
Buzz	110	121	19	-
Census	3 047	657	38	-
Clean (Musk)	14	8	4	320
Cod-rna	80	43	7	15 700
EEG Eye State	123	1 320	0.9	2 700
Forest (Class 3)	10 156	744	99	-
Forest (Class 5)	6 986	1 090	158	-
Forest (Class 7)	5 393	1 990	114	-
Hypothyroid	2	3	0.9	13
Isolet (Class A)	1 627	23	7	856
Letter (Class A)	17	4	2	1 930
Letter (Class B)	55	4	2	1 590
Letter (Class H)	74	9	2	1 970
Letter (Class Z)	31	3	2	1 710
Nursery	7	2	0.7	998
Protein	3 654	41	17	-
Ringnorm	10	13	0.6	161
Skin	81	18	12	38 200
Sleep (Class 1)	1 594	3 080	12	-
Twonorm	7	0.7	0.5	109

Running Time

	mlsvm-AMG	KaSVM	$\text{KaSVM}_{\text{fast}}$	LibSVM
	running time [s]			
Advertisement	343	192	70	557
APS failure	1 473	109	13	-
Buzz	110	121	19	-
Census	3 047	657	38	-
Clean (Musk)	14	8	4	320
Cod-rna	80	43	7	15 700
EEG Eye State	123	1 320	0.9	2 700
Forest (Class 3)	10 156	744	99	-
Forest (Class 5)	6 986	1 090	158	-
Forest (Class 7)	5 393	1 990	114	-
Hypothyroid	2	3	0.9	13
Isolet (Class A)	1 627	23	7	856
Letter (Class A)	17	4	2	1 930
Letter (Class B)	55	4	2	1 590
Letter (Class H)	74	9	2	1 970
Letter (Class Z)	31	3	2	1 710
Nursery	7	2	0.7	998
Protein	3 654	41	17	-
Ringnorm	10	13	0.6	161
Skin	81	18	12	38 200
Sleep (Class 1)	1 594	3 080	12	-
Twonorm	7	0.7	0.5	109

Running time $> 24h$

Running Time



	mlsvm-AMG	KaSVM	$KaSVM_{fast}$	LibSVM
Advertise				557
APS failu				-
Buzz				-
Census				-
Clean (Mu				320
Cod-rna				15 700
EEG Eye S				2 700
Forest (C				-
Forest (C				-
Forest (Class 7)	5 393	1 990	114	-
Hypothyro				13
Isolet (C				856
Letter (C				1 930
Letter (C				1 590
Letter (C				1 970
Letter (C				1 710
Nursery				998
Protein				-
Ringnorm				161
Skin				38 200
Sleep (Class 1)	1 594	3 080	12	-
Twonorm	7	0.7	0.5	109

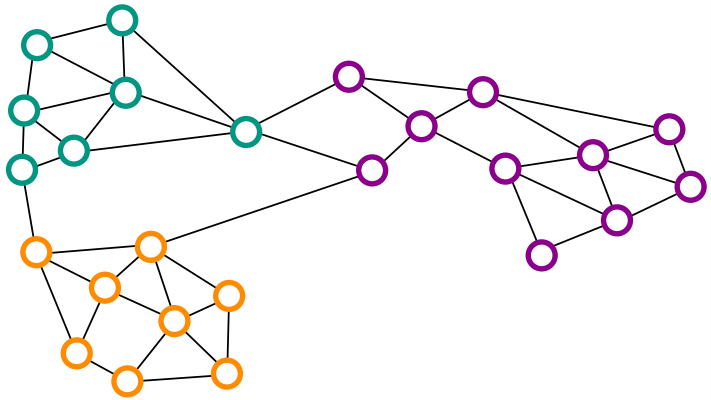
Classification Quality

Instance	mlsvm-AMG G-mean	KaSVM G-mean	KaSVM$_{fast}$ G-mean	LibSVM G-mean
Advertisement	**0.91**	**0.91**	0.87	0.941
APS failure	**0.94**	0.92	**0.94**	-
Buzz	**0.95**	0.94	0.93	-
Census	0.81	**0.83**	0.81	-
Clean (Musk)	0.88	**0.96**	0.88	0.996
Cod-rna	**0.94**	**0.94**	0.93	0.855
EEG Eye State	0.75	**0.83**	0.63	0.929
Forest (Class 3)	0.94	**0.95**	0.94	-
Forest (Class 5)	0.79	**0.90**	**0.90**	-
Forest (Class 7)	0.86	**0.93**	0.91	-
Hypothyroid	**0.94**	0.90	**0.94**	0.927
Isolet (Class A)	0.00	**0.99**	0.88	0.990
Letter (Class A)	0.94	**0.96**	0.95	0.995
Letter (Class B)	0.88	**0.92**	**0.92**	0.979
Letter (Class H)	0.76	**0.89**	0.86	0.970
Letter (Class Z)	0.92	**0.95**	**0.95**	0.991
Nursery	**1.00**	**1.00**	**1.00**	1.000
Protein	0.92	**0.93**	0.91	-
Ringnorm	**0.98**	0.97	0.82	0.987
Skin	0.99	**1.00**	**1.00**	1.000
Sleep (Class 1)	0.69	**0.70**	0.41	-
Twonorm	**0.97**	0.96	0.96	0.981

Conclusion & Discussion

KaSVM – multilevel SVM using **label propagation**

- **Comparable** classification quality
- Training: up to **two** orders of magnitude faster



Future Work:

- shared/distributed-memory parallelization
- small-diameter clustering
- different solvers for training

KaSVM - Open Source:
`https://algo2.iti.kit.edu/kasvm`