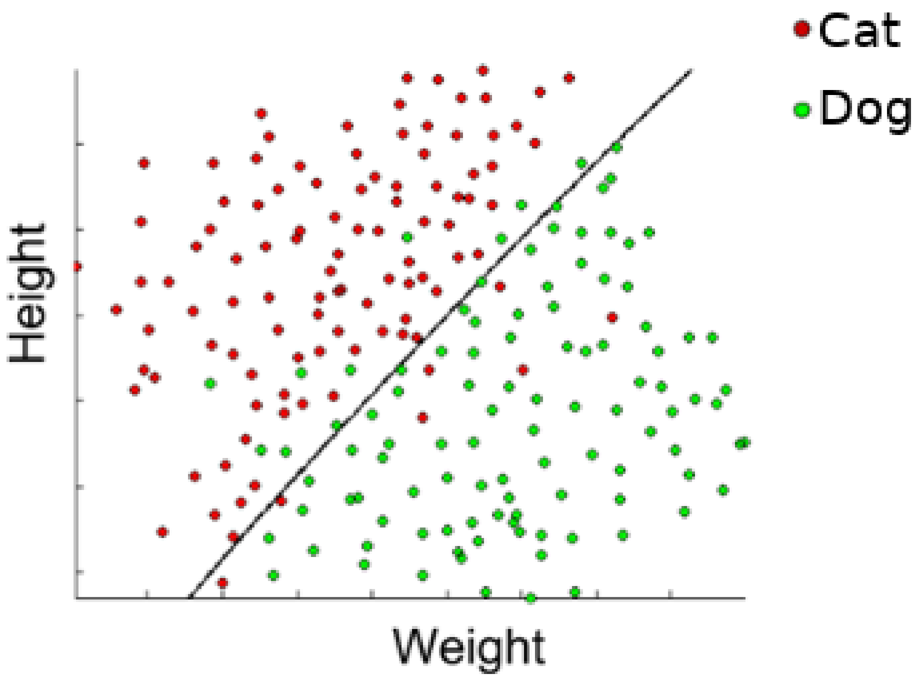
Engineering Fast Multilevel Support Vector Machines

Ehsan SadrFaridpour[†],
Talayeh Razzaghi[‡], and Ilya Safro[†]

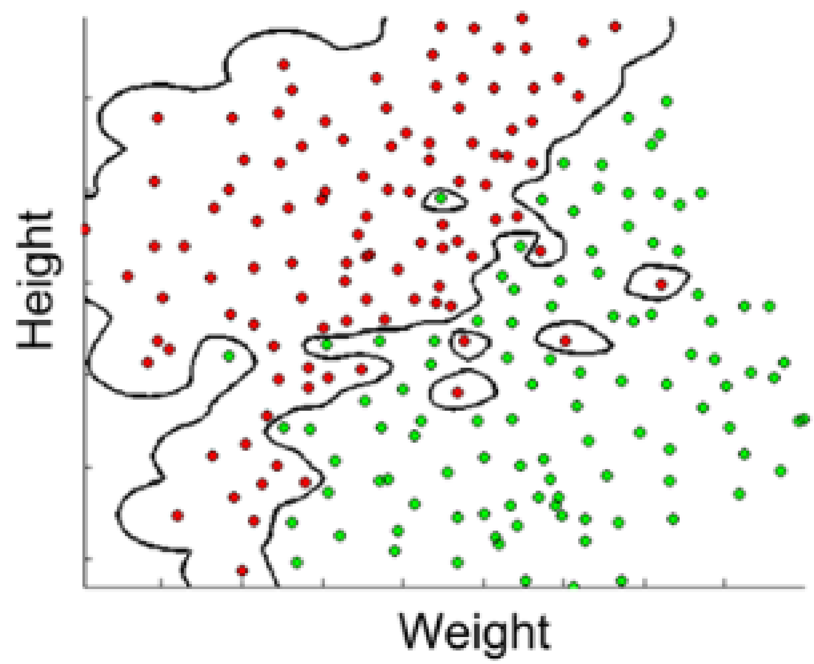
[†] Clemson University
[‡] New Mexico State University

PETSc User Meeting 2019, Georgia Tech, Atlanta, GA, USA, June 2019

Toy Example for Classification



Linear Classifier



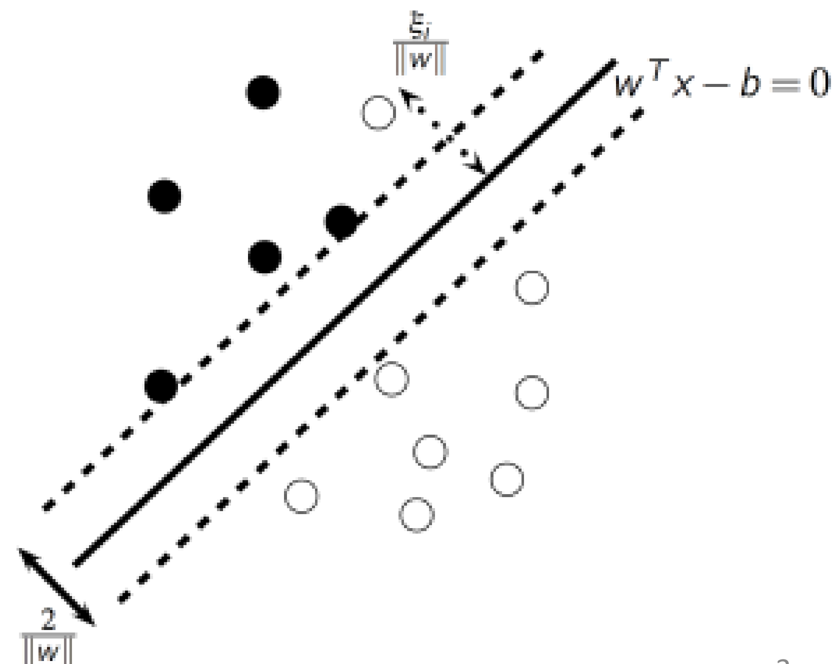
Non-Linear Classifier

Soft Margin SVM

Classification Problem

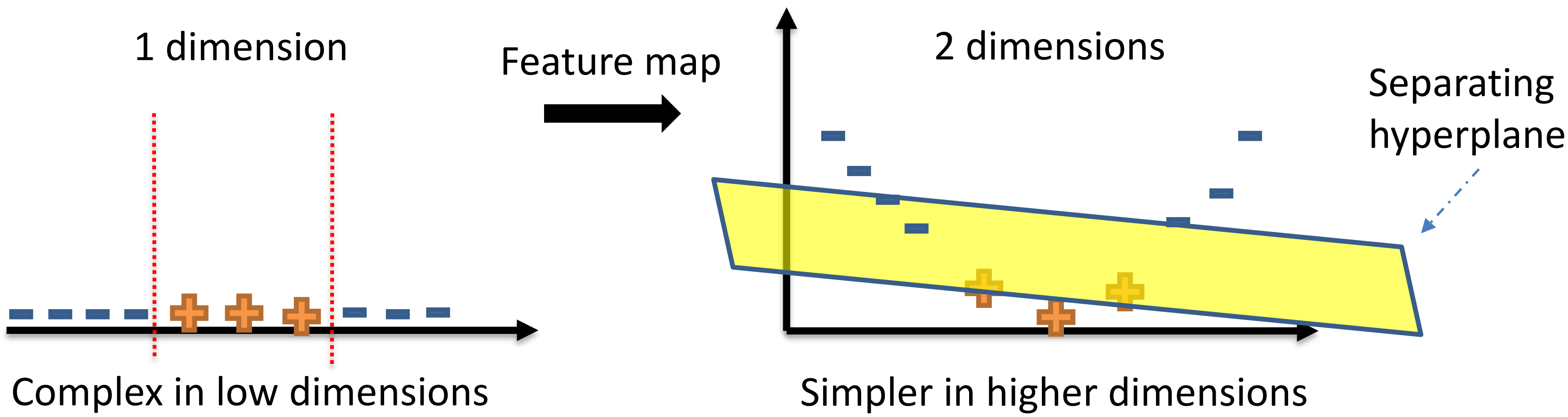
- Data is represented by $(x_i, y_i) \in \mathbb{R}^n \times \{-1, 1\}$
  - x_i: actual *data*
  - y_i: corresponding *label* (binary case)
- Find *classifier* $f : \mathbb{R}^n \mapsto \{-1, 1\}$ to predict the labels y_i^{test} of a group of data samples x_i^{test}

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$
$$\text{s.t.} \quad \forall i \quad y_i(w^T x_i - b) \geq 1 - \xi_i,$$



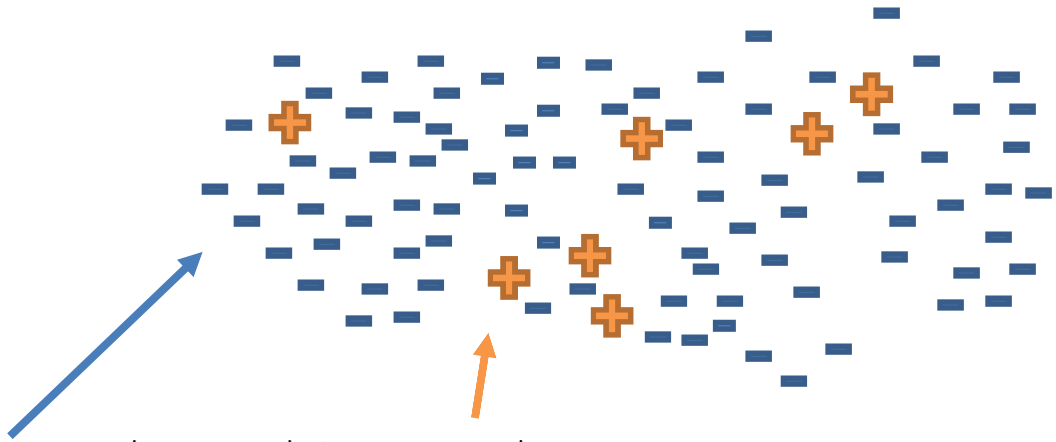
- w and b are hyperplane's parameters
- Parameter C controls misclassification penalty

Kernel & Weighted SVM



- Embed data from **input space** to a higher dimension **feature space**
- For an embedding $\phi(x)$, denote $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$
- We use Radial Basis Function (RBF) kernel
- RBF: $K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$

Weighted SVM for Imbalanced Classification



- $|\text{Class N}| >> |\text{Class P}|$
  (Number of data points)
- **Class P is more important to classify correctly**
- Penalize misclassification of each class with coefficients C^+/C^-

$$\min_{w,b,\xi} \ \frac{1}{2}\|w\|^2 + C^+ \sum_{i=1}^{m^+} \xi_i + C^- \sum_{i=1}^{m^-} \xi_i$$

$$\text{s.t.} \ \ y_i(w^T x_i - b) \geq 1 - \xi_i, \qquad i = 1, \ldots, m$$

What Makes the Nonlinear (W)SVM Slow?

- Complexity of SVM solver is between $O(n \times m^2)$ and $O(n \times m^3)$

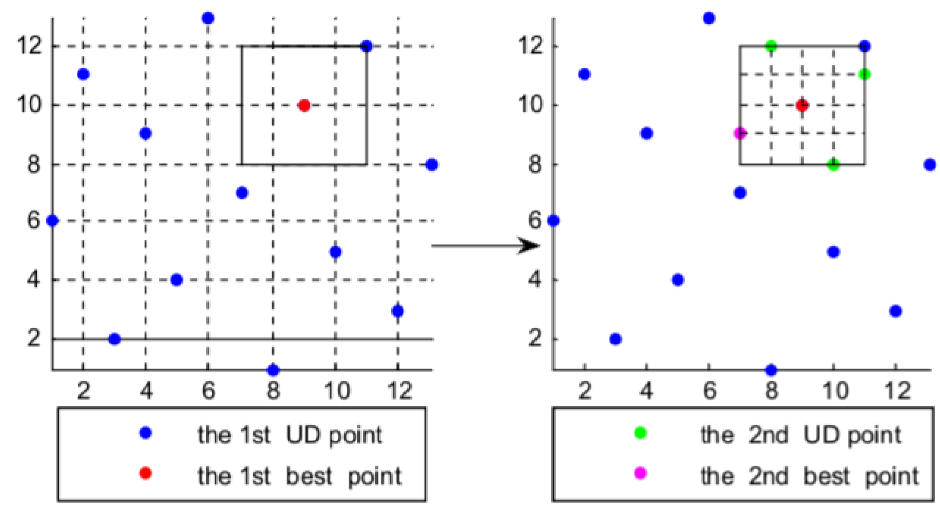
m: number of points in both classes

n: number of features

- Parameter fitting (Model Selection)

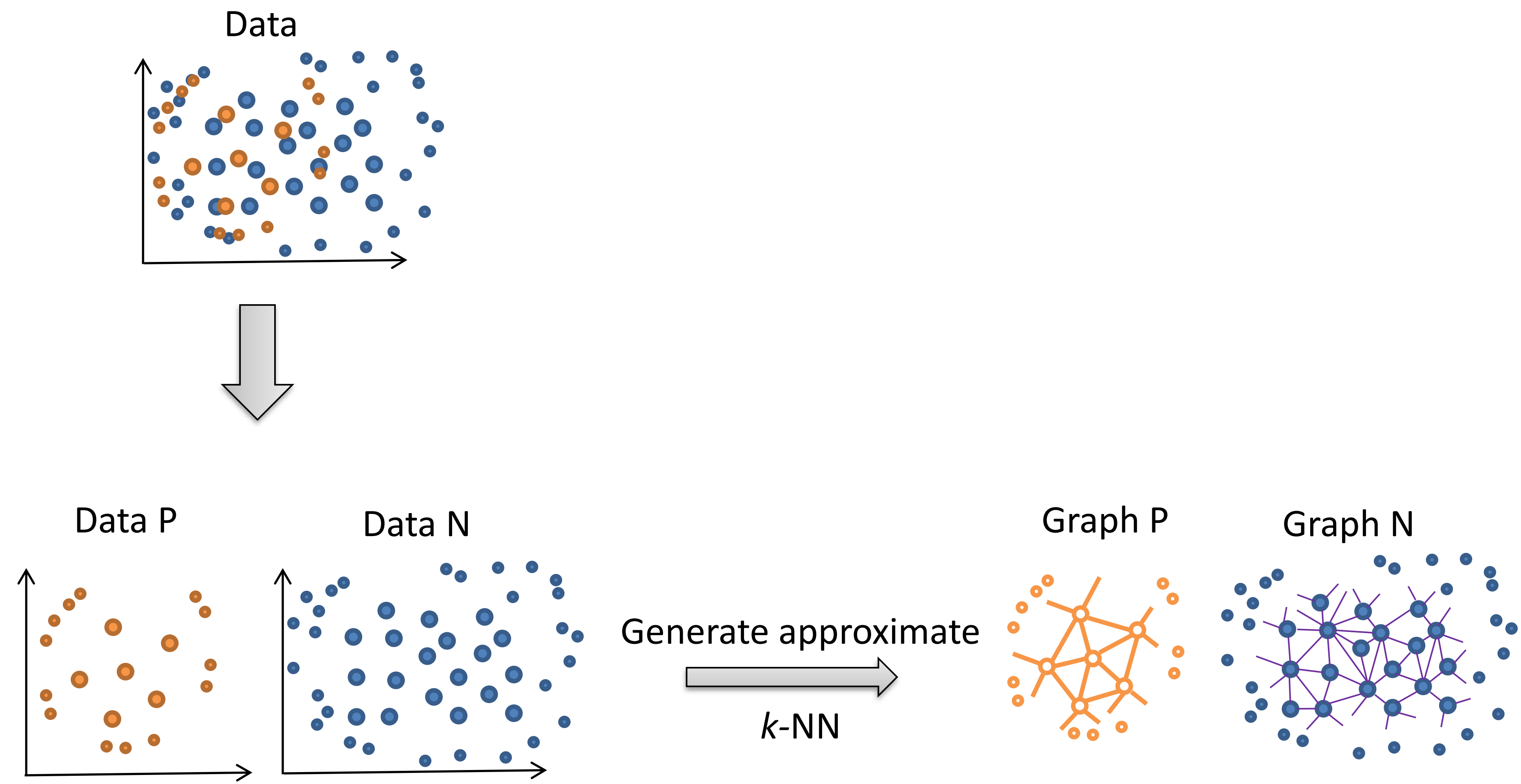
C: Misclassification penalty

γ: RBF Kernel parameter

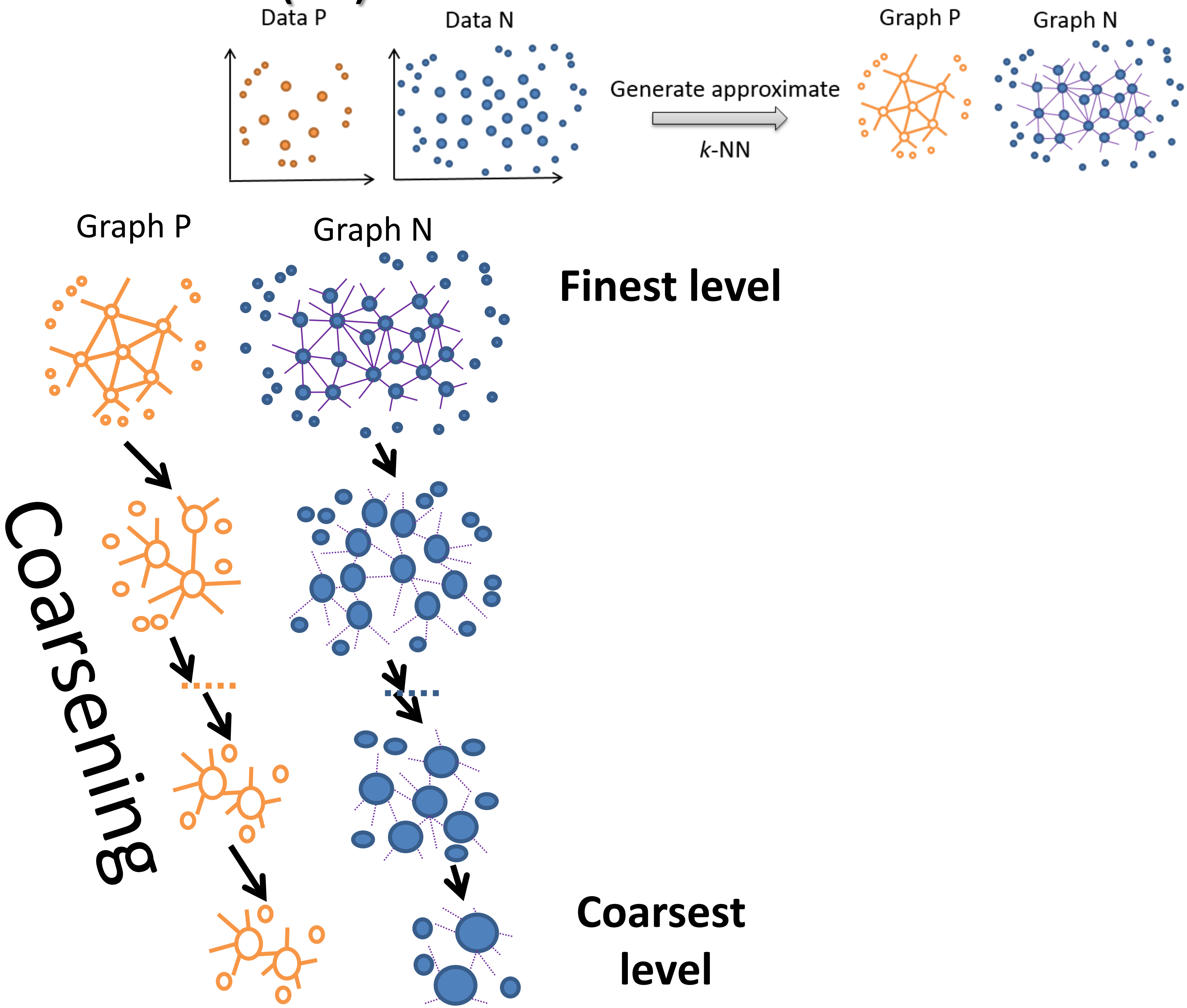


a First stage *b* Second stage

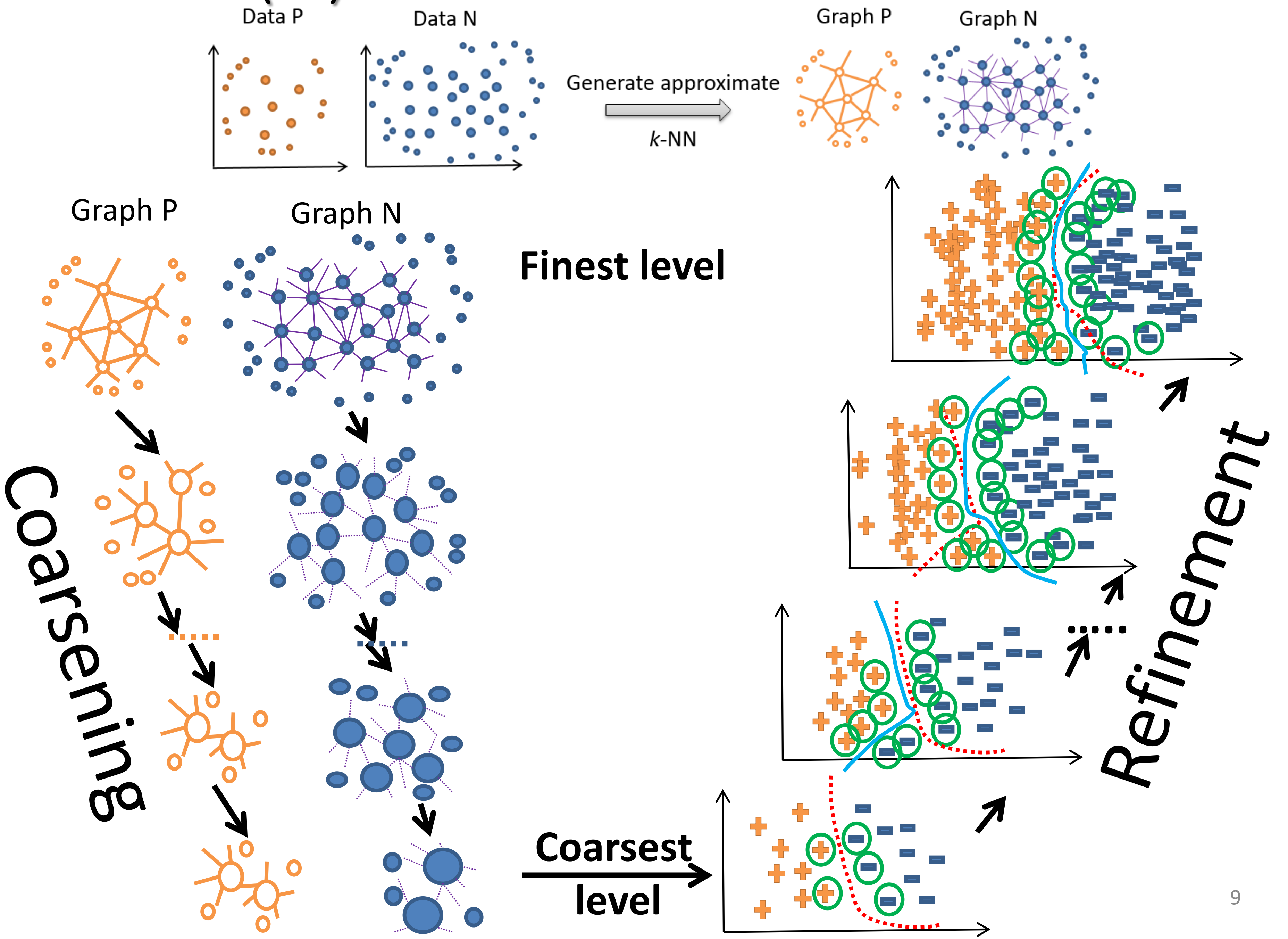
Multilevel (W)SVM



Multilevel (W)SVM

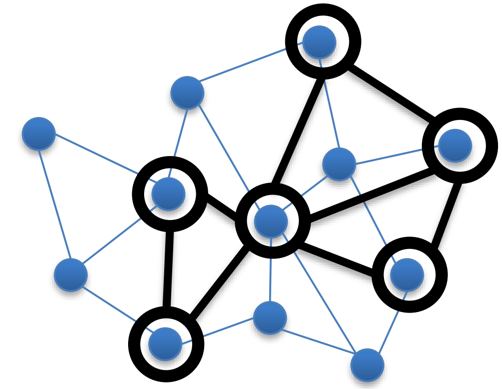


Multilevel (W)SVM

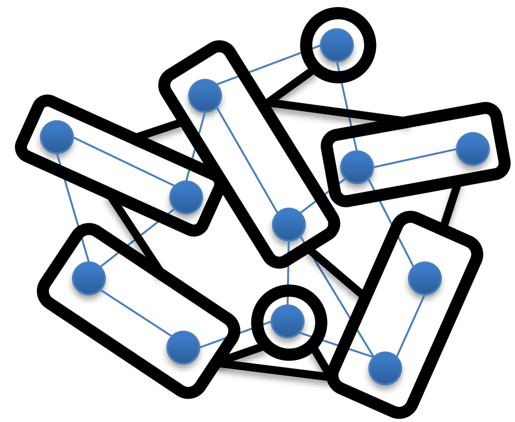


Types of Coarsening

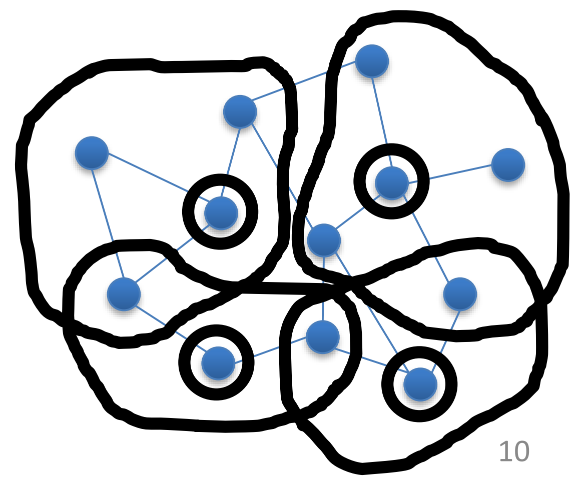
1. Sampling independent sets



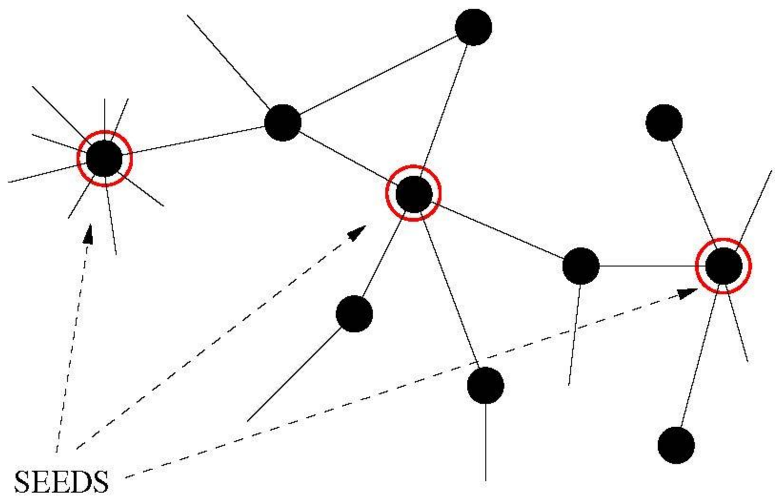
2. Strict coarsening (merging pairs)



3. AMG coarsening (Galerkin on Laplacian)



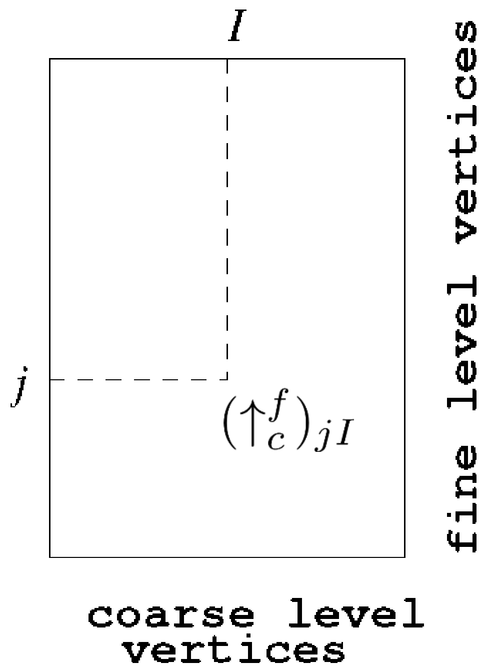
AMG: Coarse Variables



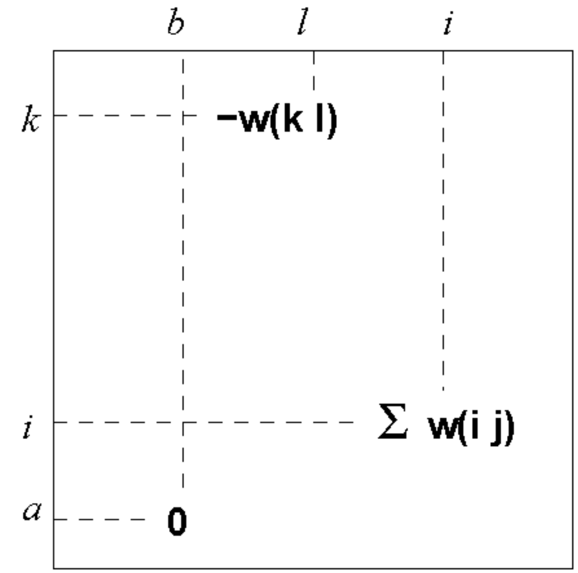
- Choose a dominating set $C \subset V$ s.t. all others from $F = V \setminus C$ are "strongly coupled" to C
- "Strongly coupled" $=$ Kernel coupling $+$ algebraic distances ρ_{ij}
  Safro, Chen "Algebraic distance on graphs", 2012
- Restriction matrix is sparsified by algebraic distances

AMG: Coarse Laplacian

$\uparrow_c^f$ - restriction operator



L_f - weighted Laplacian at level f, where weights are kernel distances



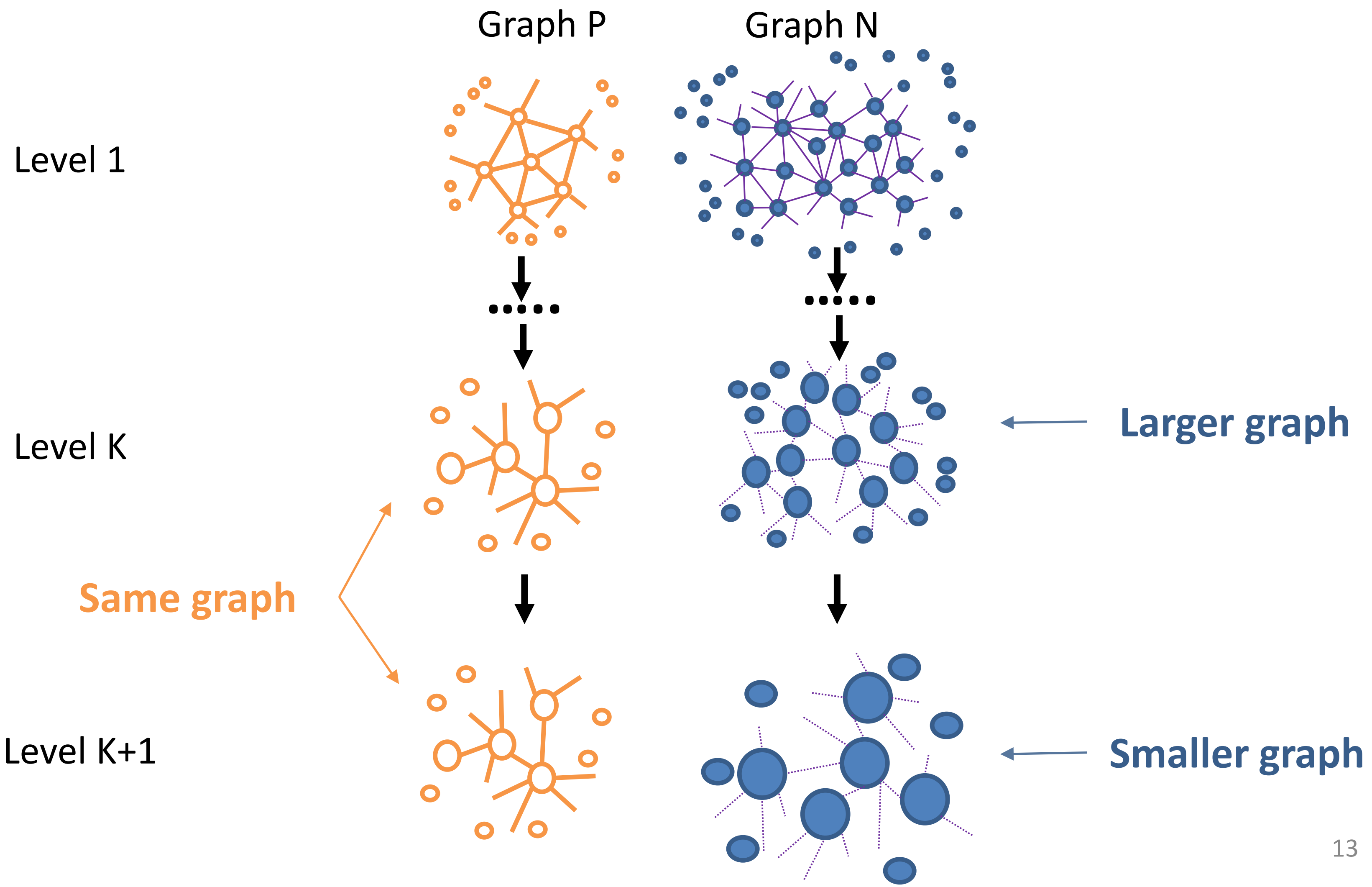
Coarse graph Laplacian

$$L_c \leftarrow (\uparrow_c^f)^T L_f \uparrow_c^f$$

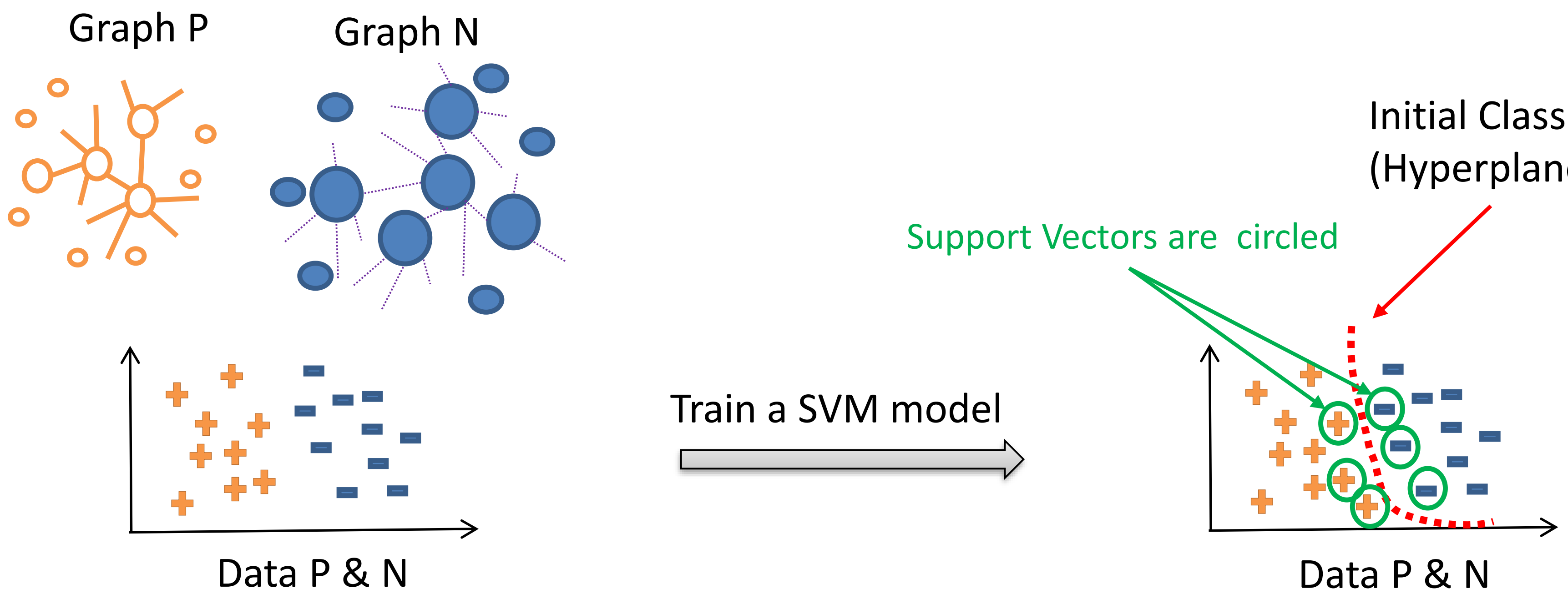
$$w_{IJ} = \sum_{l,k} (\uparrow_c^f)_{Il} \cdot w_{lk} \cdot (\uparrow_c^f)_{kJ}$$

Coarsening for Imbalance Classes

- Aggregate multiple points or their fractions
- For imbalance classes, stop coarsening for the smaller class (P)



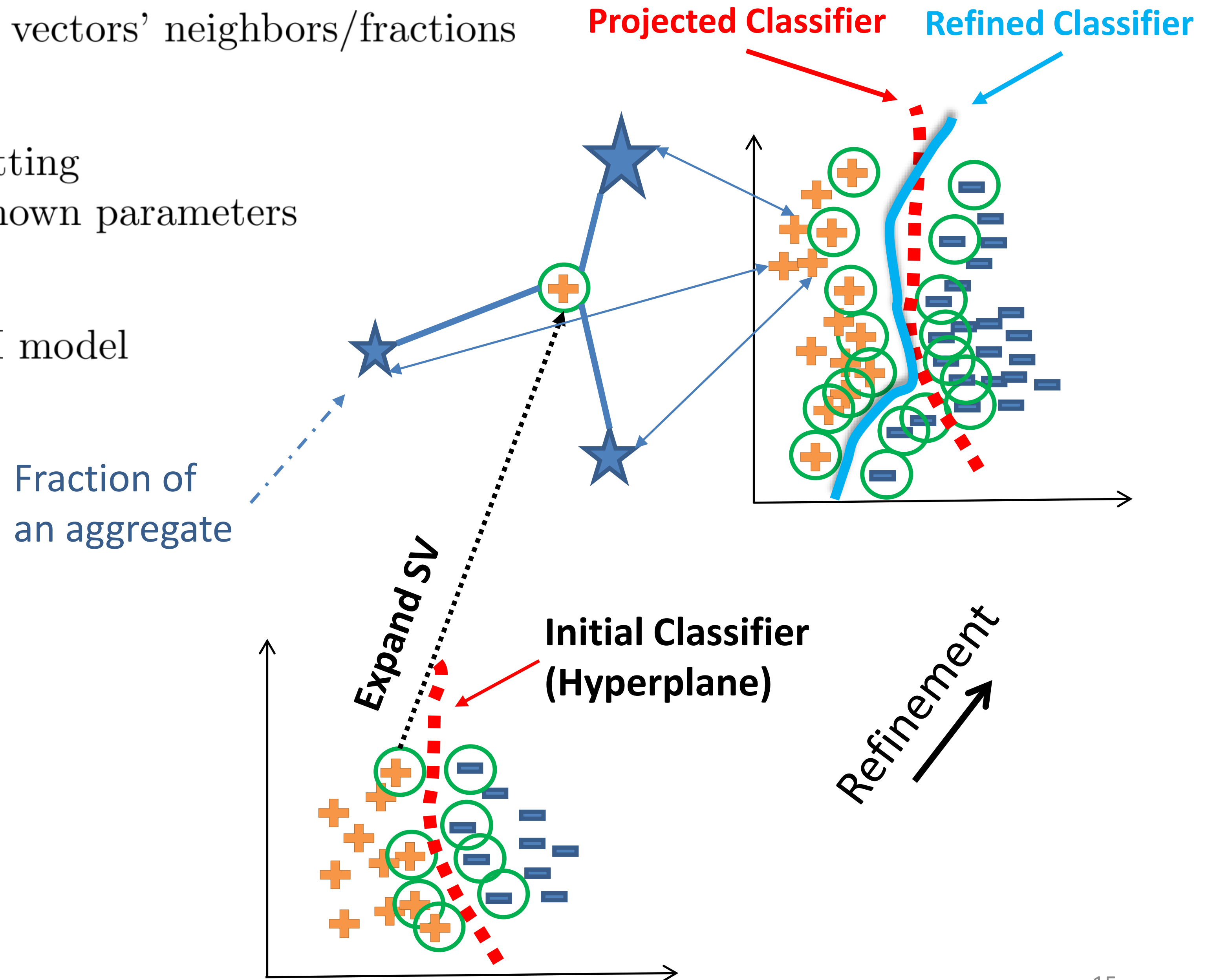
Train a SVM on the Coarsest Data



- Apply complex parameter fitting
- Fast running time (small data)
- Balanced class sizes

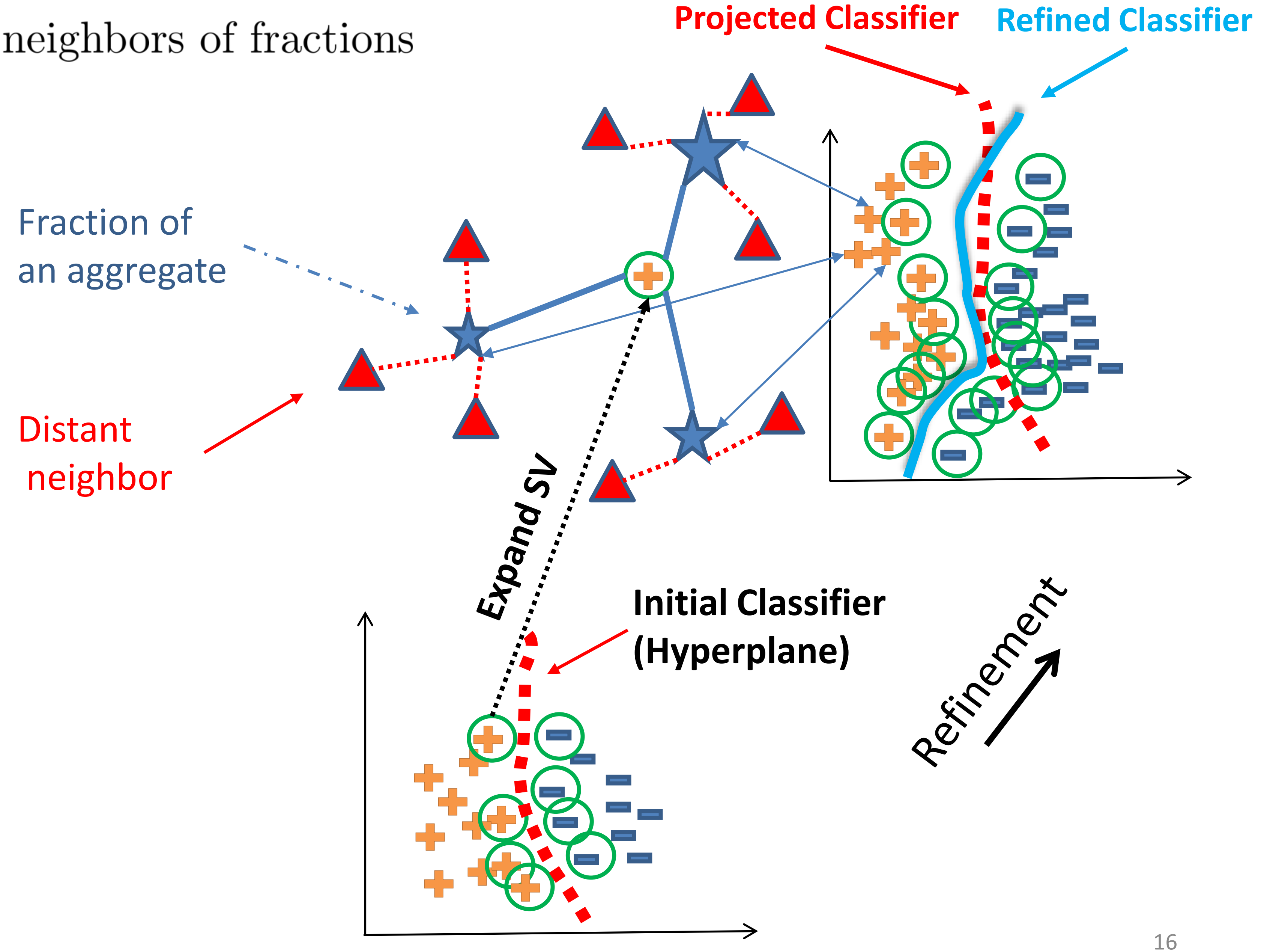
Refinement

- Add support vectors' neighbors/fractions
- Parameter fitting
  using best known parameters
- Train a SVM model



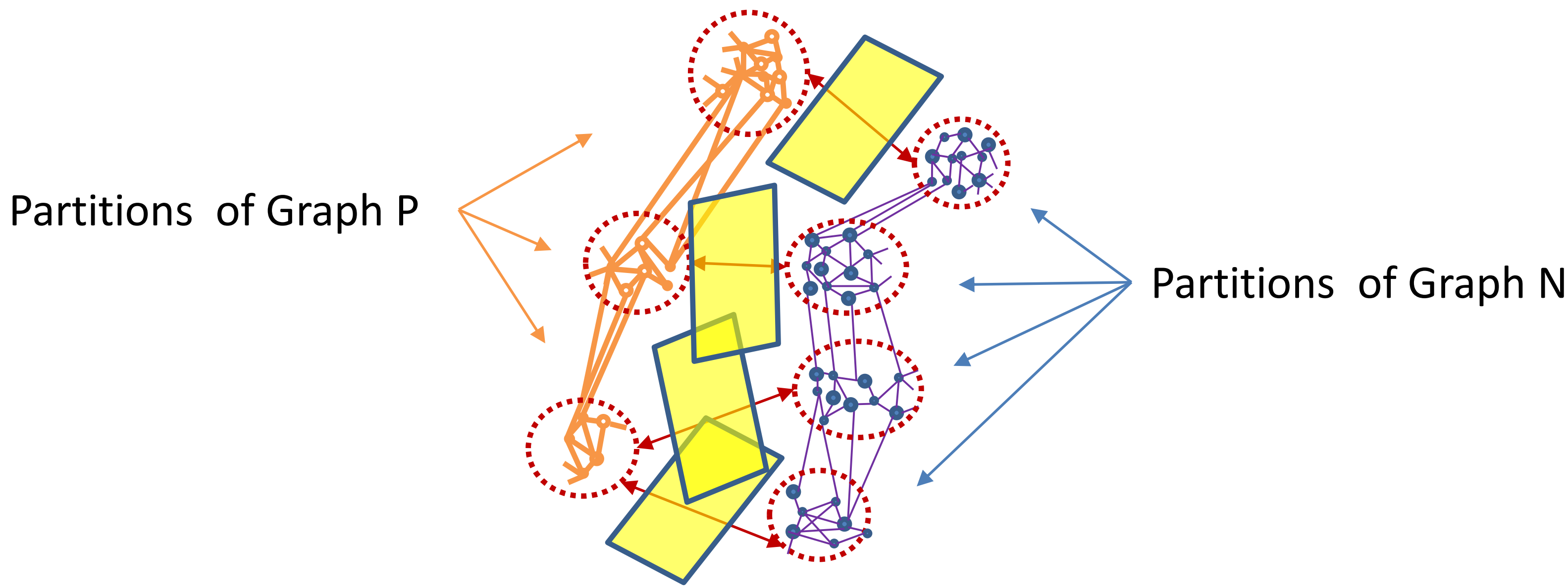
Distant Neighbors in Refinement

Adding neighbors of fractions

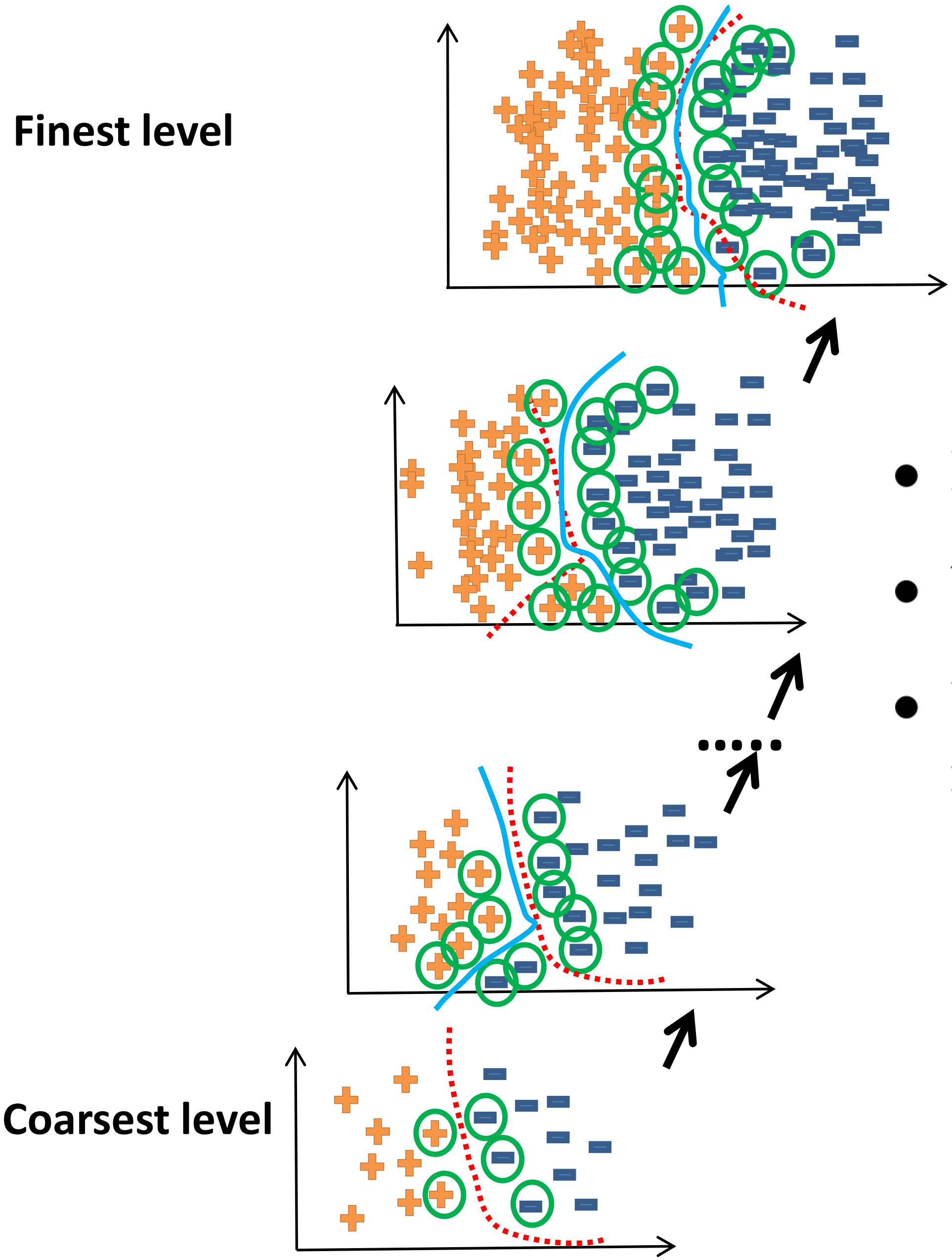


What If the Number of SVs is Too Big?

- Partition the graph
- Training is performed by pairs of data partitions
- Generate multiple classification models



The Benefits of a Multilevel Framework



- Multiple models at various scale
- Variety of classification qualities
- Use large validation data from finest level

Performance Measures

		Ground Truth	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

- **Accuracy** $= \frac{TP+TN}{FP+TN+TP+FN}$
- **Sensitivity(recall)** $= \frac{TP}{TP+FN}$
- **Specificity** $= \frac{TN}{TN+FP}$
- **G-mean** $= \sqrt{\text{Sensitivity} * \text{Specificity}}$
- **G-mean is the best performance measurement** on both small and large classes using one value

Computational Results

- Multilevel SVM vs LIBSVM and DC-SVM
  - On small datasets
  - Validation of proposed framework
- Multilevel SVM vs LIBLINEAER
  - On large datasets
  - Benchmark the performance and quality
  - **The state-of-the-art non-linear solver (LIBSVM) and DC-SVM are impractical**

Small and Large Imbalanced Datasets

Dataset	r_{imb}	|features|	Size	|Class P|	|Class N|
Advertisement	0.86	1558	3279	459	2820
Buzz	0.80	77	140707	27775	112932
Clean (Musk)	0.85	166	6598	1017	5581
Cod-rna	0.67	8	59535	19845	39690
Forest (Class 5)	0.98	54	581012	9493	571519
Letter	0.96	16	20000	734	19266
Nursery	0.67	8	12960	4320	8640
Ringnorm	0.50	20	7400	3664	3736
Twonorm	0.50	20	7400	3703	3697

Dataset	r_{imb}	|features|	Size	|Class P|	|Class N|
SUSY	0.54	18	5M	2.29M	2.71M
MNIST	0.90	784	4M	0.4M	3.65M
HIGGS	0.53	28	11M	5.17M	5.83M

Imbalance ratio between size of classes ($r_{imb} = \frac{|ClassN|}{|ClassN|+|ClassP|}$)
Size: Number of data points in both classes
|Class P|: Number of data points in class P

Running Time on Small Datasets

MLSVM has significant improvement on performance

Dataset	LIBSVM	DC-SVM	MLSVM (AMG)
Advertisement	231	610	**213**
Buzz	26026	2524	**31**
Clean (Musk)	**82**	95	94
Cod-RNA	1857	420	**13**
Forest	353210	19970	**948**
Letter	139	38	**30**
Nursery	192	49	**2**
Ringnorm	26	38	**2**
Twonorm	28	30	**1**

G-mean on Small Datasets

Multilevel WSVM (MLSVM) produces higher G-mean than the state-of-the-art WSVM (LIBSVM) and DC-SVM almost on all datasets

Datasets	LIBSVM	DC-SVM	MLSVM (AMG)
Advertisement	0.67	0.90	**0.91**
Buzz	0.89	0.92	**0.95**
Clean (Musk)	**0.99**	0.94	**0.99**
Cod-RNA	**0.96**	0.93	0.94
Forest	0.92	**0.94**	0.88
Letter	0.99	**1.00**	0.99
Nursery	**1.00**	**1.00**	**1.00**
Ringnorm	**0.98**	0.95	**0.98**
Twonorm	**0.98**	0.97	**0.98**

Running Time on Large Datasets

- MLSVM is faster of normal large dataset
- LIBLINEAR is faster over high dimensional datasets

Dataset	LIBLINEAR	LIBSVM or DC-SVM	MLSVM (AMG)
SUSY	1300	No	**1116**
MNIST	**1840**	results	14724
HIGGS	4406	in 3 days	**3283**

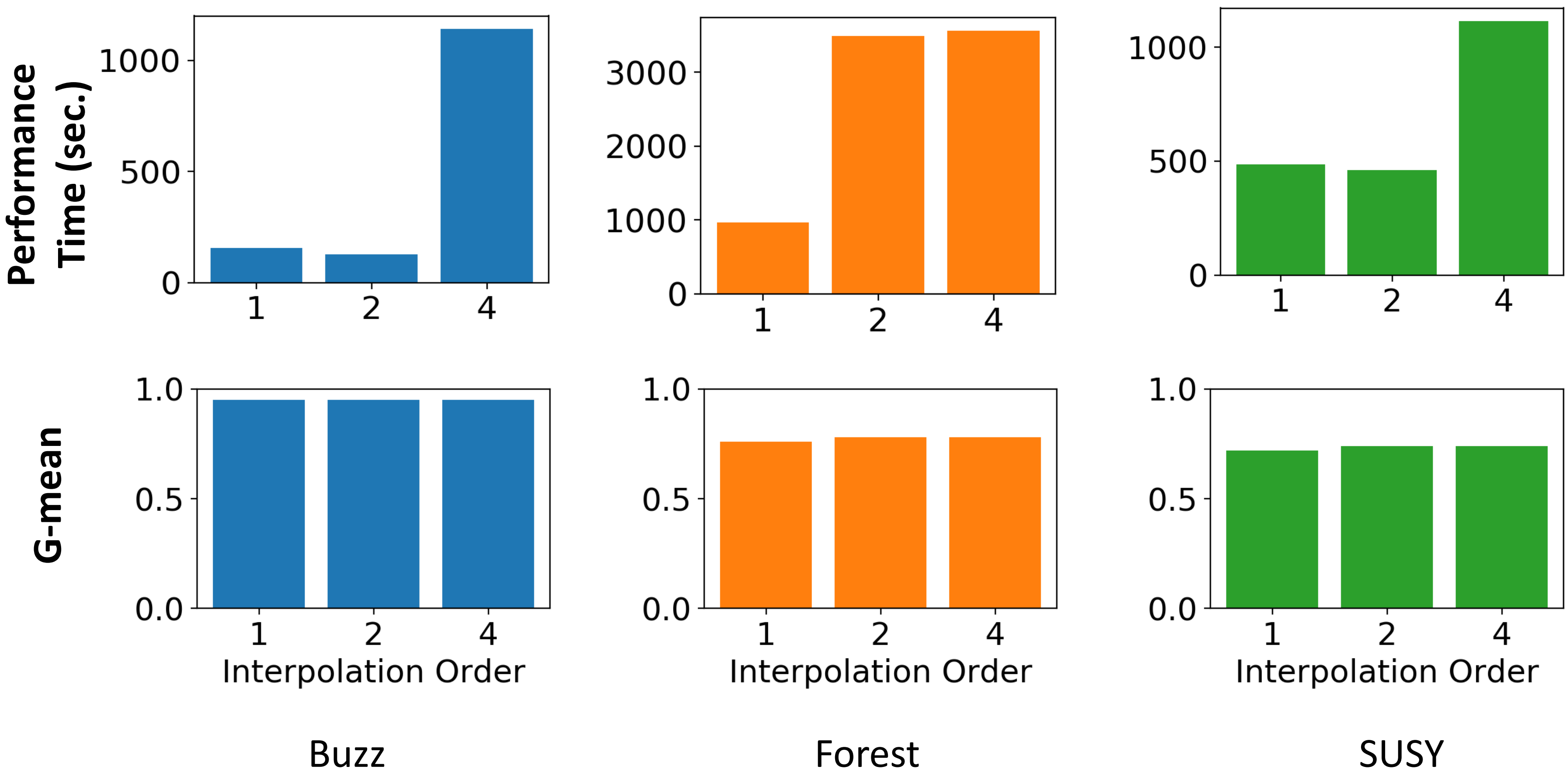
G-mean on Large Datasets

- **MLSVM produces higher G-mean**

Dataset	LIBLINEAR	LIBSVM& DC-SVM	MLSVM (AMG)
SUSY	0.68	No	**0.74**
MNIST	0.85	results	**0.90**
HIGGS	0.54	in 3 days	**0.62**

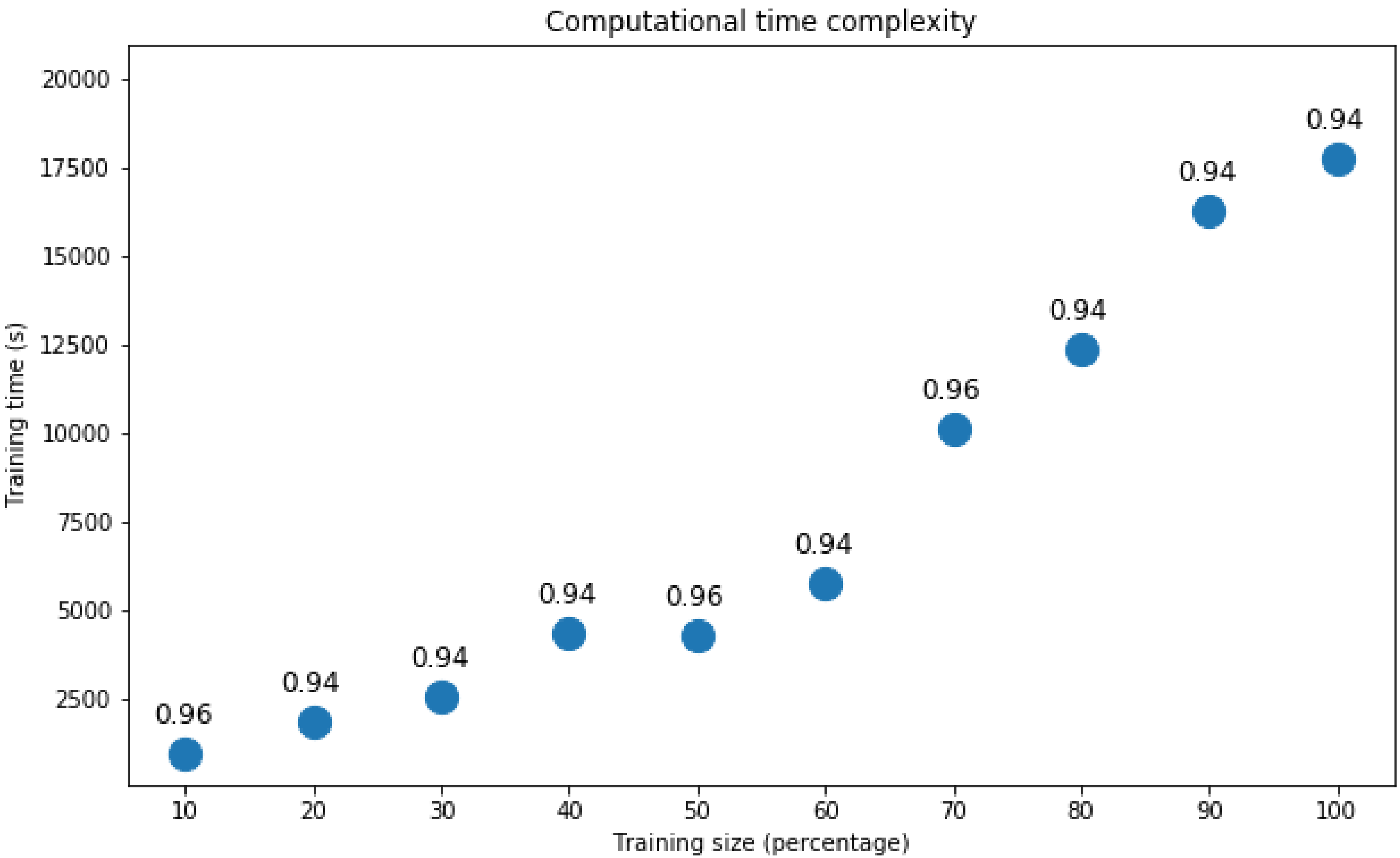
Interpolation Order vs Performance Time and G-mean

- The performance dropped on larger interpolation orders
- The G-mean is not significantly improved



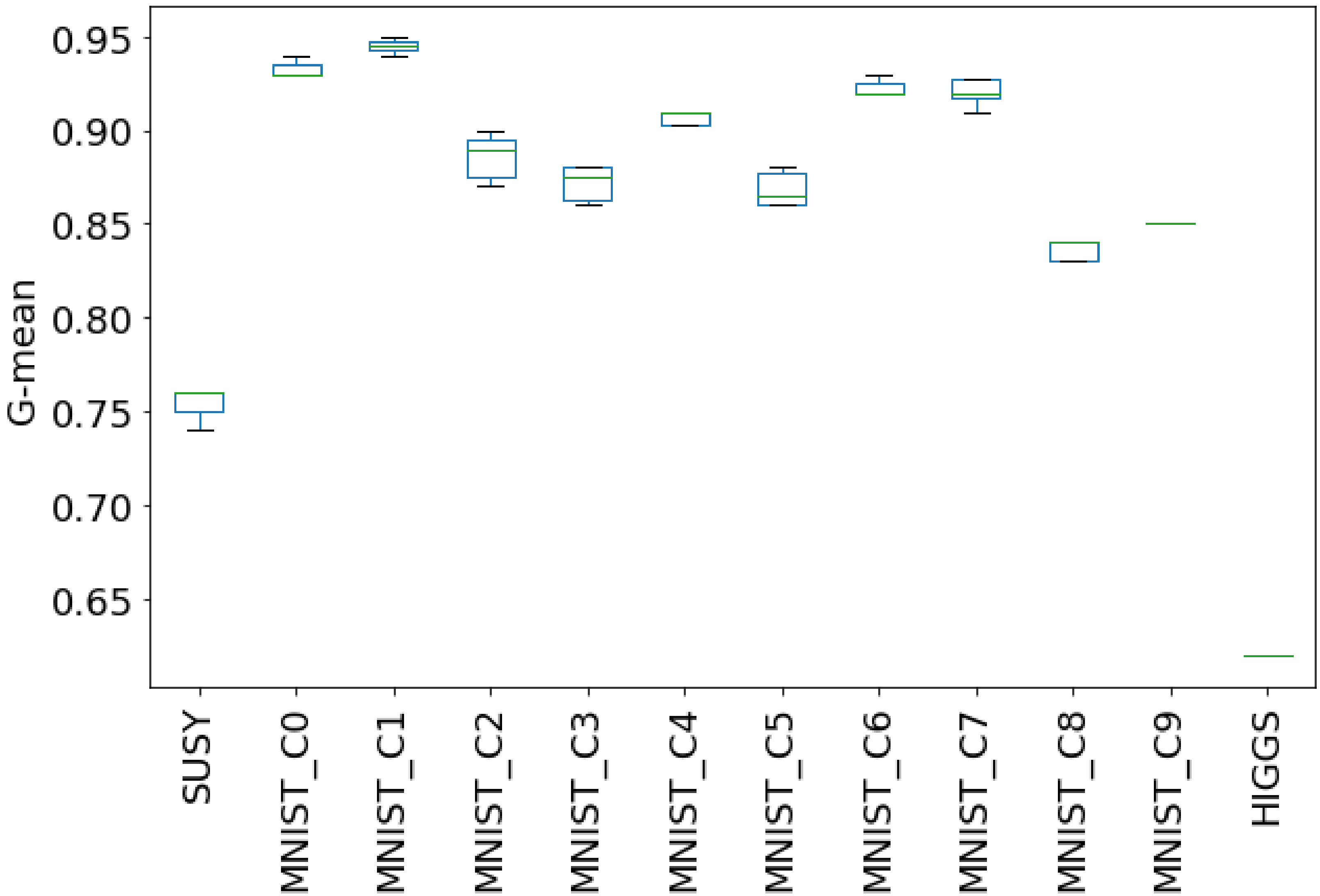
Scalability: Size vs Time

- Scalability on growing training size
- Better than linear scalability



Statistics of G-mean Variability

- Low G-mean variance confirms the robustness of MLSVM



Why We Used PETSc?

- Fast linear algebra operation for vectors and sparse matrices
- Variety of methods for constructing, slicing and summarizing matrices
- Convenient integration of other libraries such as Metis, LIBSVM, and FLANN with PETSc
- Allow us to make a parallel and distributed version using PETSc data structures and methods
- Compatible with other scientific packages like HYPRE which we can use to try other multigrid methods

Future Direction

- Looking for collaboration in scientific computing
  Machine learning

- Build a parallel and distributed version of MLSVM

- Leverage other multigrid methods for coarsening

- Source Code: https://github.com/esadr/mlsvm

References

- Sadrfaridpour, Razzaghi, Safro “Engineering fast multilevel support vector machines”, 2019
- Sadrfaridpour, et al. "Algebraic multigrid support vector machines." , 2017
- Razzaghi, Roderick, Safro, Marko “Multilevel Weighted Support Vector Machine for Classification on Healthcare Data with Missing Values”, 2016
- Razzaghi, Safro “Scalable Multilevel Support Vector Machines”, 2015
- Ron, Safro, Brandt “Relaxation-based Coarsening and Multiscale Graph Organization”, 2011

https://ehsan.faridpour.com

esadrfa@g.clesmon.edu