Maximum Margin Clustering Made Practical

Kai Zhang, Ivor W. Tsang, James T. Kwok

Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong

香 港 科 技 大 學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Outline

Outline

Support Vector Machine

$$\textit{Primal} \quad \min_{\mathbf{w},b,\xi_i} \quad \|\mathbf{w}\|^2 + 2C\boldsymbol{\xi}^\top \mathbf{e}$$
$$\text{s.t.} \quad y_i(\mathbf{w}^\top \varphi(\mathbf{x}_i) + b) \geq 1 - \xi_i,\ \xi_i \geq 0.$$

$$\textit{Dual} \quad \max_{\boldsymbol{\lambda}} \quad 2\boldsymbol{\lambda}^\top \mathbf{e} - \boldsymbol{\lambda}^\top (\mathbf{K} \circ \mathbf{y}\mathbf{y}^\top)\boldsymbol{\lambda}$$
$$\text{s.t.} \quad \boldsymbol{\lambda}^\top \mathbf{y} = 0,\ C\mathbf{e} \geq \boldsymbol{\lambda} \geq \mathbf{0}$$

Features

- good generalization by maximizing the margin $\frac{1}{\|w\|^2}$
- non-linear feature mapping through kernel K
- can also be extended to unsupervised scenario

Maximum Margin Clustering

Basic idea: perform clustering by simultaneously finding maximum margin hyper-plane in the data.

- Formulation [Xu et. al.]

$$\begin{aligned} \min_{\mathbf{y}} \max_{\boldsymbol{\lambda}} \quad & 2\boldsymbol{\lambda}^\top \mathbf{e} - \boldsymbol{\lambda}^\top (\mathbf{K} \circ \mathbf{y}\mathbf{y}^\top)\boldsymbol{\lambda} \\ \text{s.t.} \quad & \boldsymbol{\lambda}'\mathbf{y} = 0, \;\; C\mathbf{e} \geq \boldsymbol{\lambda} \geq \mathbf{0}, \\ & y_i = \{\pm 1\}. \end{aligned}$$

- Balance constraint $-\ell \leq \mathbf{e}^\top \mathbf{y} \leq \ell$
- Semi-definite relaxation [Lanckriet et. al.]

$$\begin{aligned} \min_{\mathbf{M},\delta,\boldsymbol{\mu},\boldsymbol{\nu}} \quad & \delta \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{M} \circ \mathbf{K} & \mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\nu} \\ (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\nu})^\top & \delta - 2C\boldsymbol{\nu}'\mathbf{e} \end{bmatrix} \succeq 0, \\ & \text{diag}(\mathbf{M}) = \mathbf{e}, \mathbf{M} \succeq 0, -\ell\mathbf{e} \leq \mathbf{M}\mathbf{e} \leq \ell\mathbf{e}. \end{aligned}$$

Maximum Margin Clustering

Pros and Cons of MMC

- Convex optimization problem
- Cannot include the bias
- Computationally very expensive (number of variables quadratic with sample size)

Extension: Generalized MMC [Valizadegan and Jin]

- Number of variables only linear with sample size
- Incorporate the bias term
- Still rely on SDP and expensive

Iterative Approach

[Xu et. al.] propose a iterative SVM procedure

IterSVM

1: Initialize the label $\mathbf{y}$.
2: Fix $\mathbf{y}$, perform standard SVM training.
3: Compute $\mathbf{w}$ and b by KKT conditions.
4: Assign the labels by $y_i = \text{sign}(\mathbf{w}^\top \varphi(\mathbf{x}_i) + b)$.
5: Repeat steps 2-4 until convergence.

Problem: Empirically, premature convergence, the initial solution only improved slightly.

Illustration

- Data set: UCI digits 3 and 9

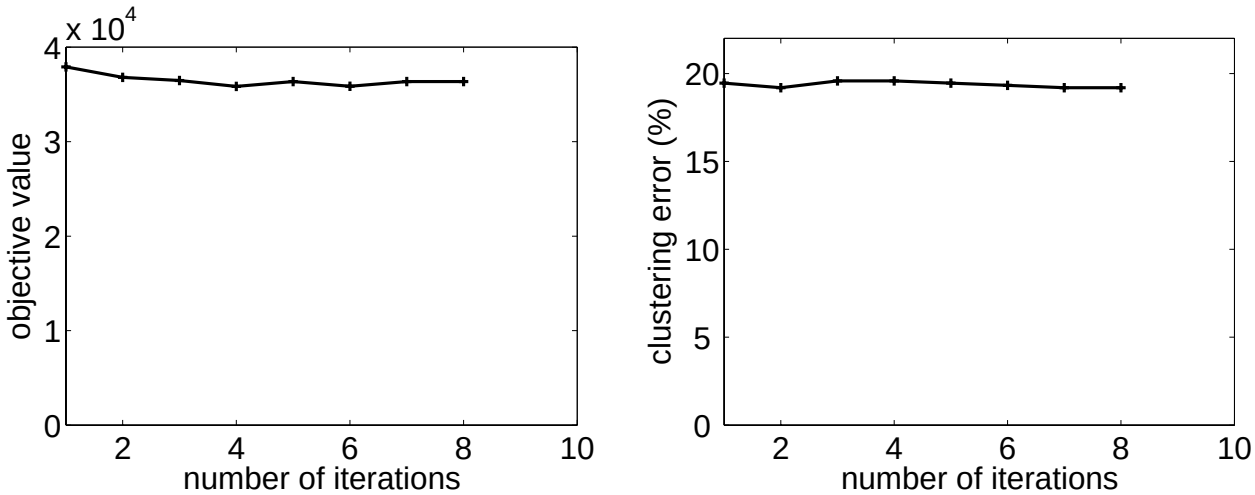


Figure: Iterative SVM procedure. Left: objective value; right: classification error.

Outline

Why Premature Convergence?

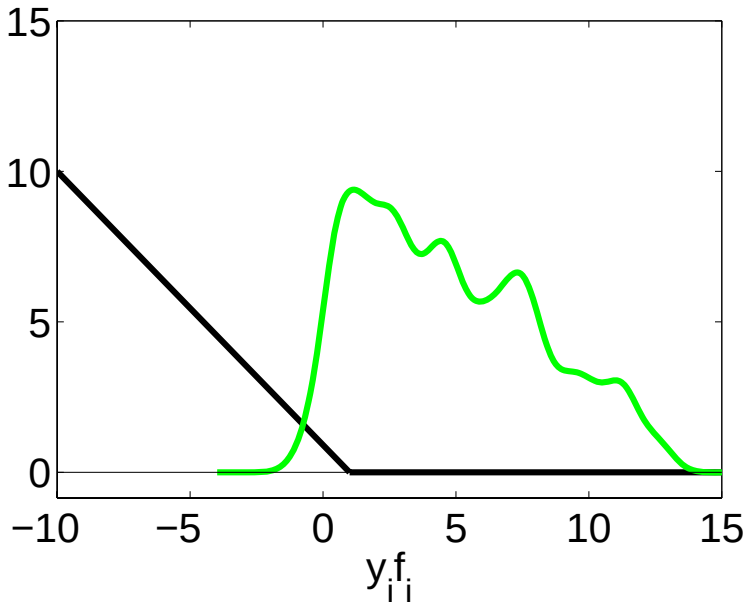
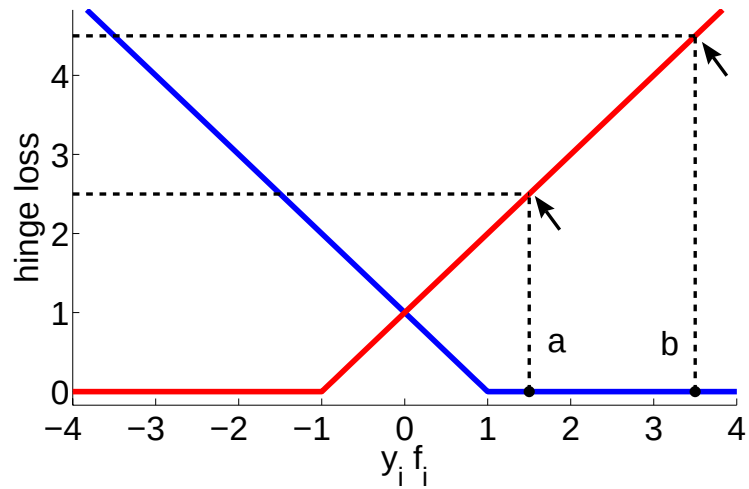


Figure: The hinge loss function and the empirical distribution of $y_i f_i$'s in one classification task.

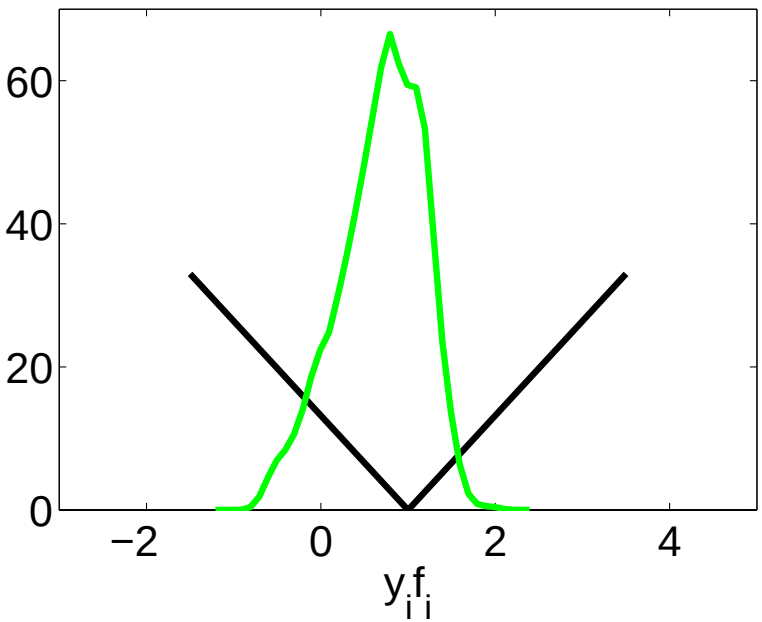
Observation

1. most $y_i f_i$'s are pushed to the right
2. flipping of y_i's is discouraged



Discouraging Premature Convergence

- Basic idea: apply the laplacian loss $L_s = |f_i - y_i|$ that is symmetric around $y_i f_i = 1$.
  - squared L_2 loss has similar effects.
- Advantage: flipping of y_i's no longer induces large increase of the objective.



The laplacian loss function and the resultant empirical distribution of the yifi's in one classification task.

Formulation

Primal (SVR with 0-sensitive loss)

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i,\xi_i^*} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{s.t.} \quad & y_i - (\mathbf{w}^\top \varphi(\mathbf{x}_i) + b) \le \xi_i, \\ & -y_i + (\mathbf{w}^\top \varphi(\mathbf{x}_i) + b) \le \xi_i^*, \\ & \xi_i \ge 0, \quad \xi_i^* \ge 0, \end{aligned}$$

Dual

$$\begin{aligned} \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \quad & (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^\top \mathbf{y} - \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^\top \mathbf{K}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \\ \text{s.t.} \quad & (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^\top \mathbf{e} = 0, \quad C\mathbf{e} \ge \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \ge \mathbf{0}, \end{aligned}$$

Comparison with SVM Classification

- By $\mathbf{u} = \alpha - \alpha^*$ our dual becomes

$$\begin{aligned} \max_{\mathbf{u}} \quad & \mathbf{u}^\top \mathbf{y} - \frac{1}{2}\mathbf{u}^\top \mathbf{K}\mathbf{u} \\ \text{s.t.} \quad & \mathbf{u}^\top \mathbf{e} = 0, \;\; C\mathbf{e} \geq \mathbf{u} \geq -C\mathbf{e}. \end{aligned}$$

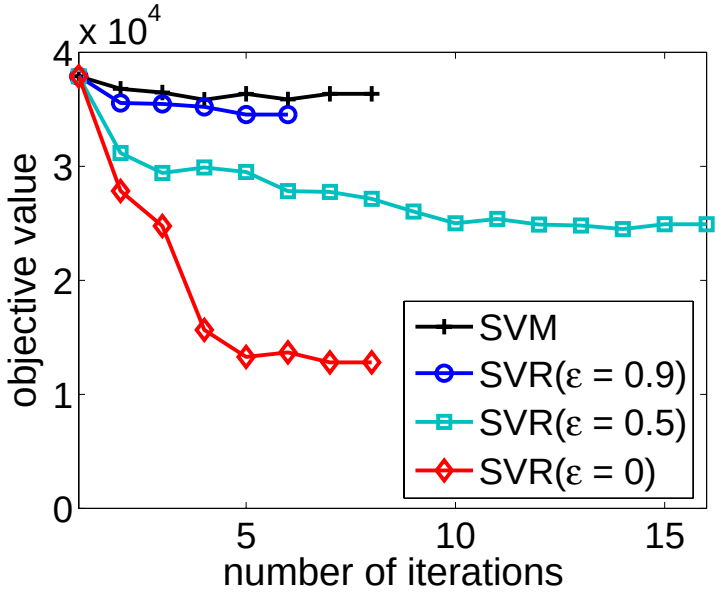
- Since $y_i \in \{\pm 1\}$, factorize as $u_i = u_i y_i^2 = \lambda_i y_i$, with $\lambda_i = u_i y_i$, the dual is

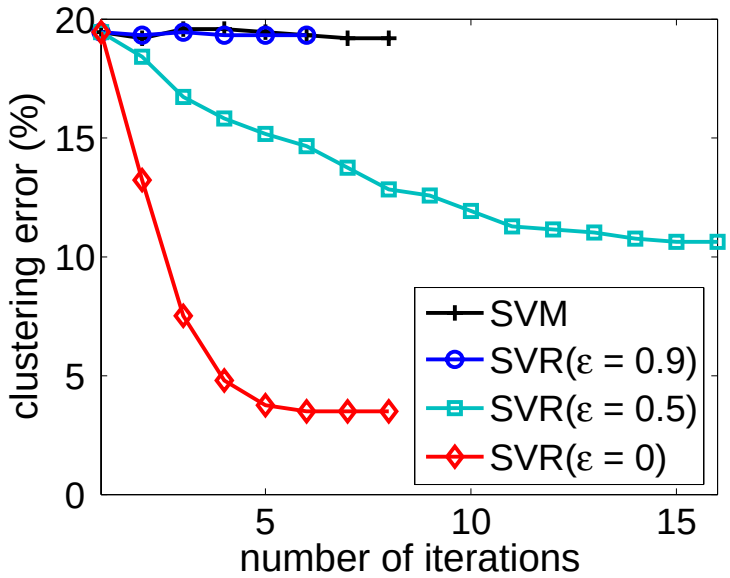
$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & 2\boldsymbol{\lambda}^\top \mathbf{e} - \boldsymbol{\lambda}^\top (\mathbf{K} \circ \mathbf{y}\mathbf{y}^\top)\boldsymbol{\lambda} \\ \text{s.t.} \quad & \boldsymbol{\lambda}^\top \mathbf{y} = 0, \;\; C\mathbf{e} \geq \boldsymbol{\lambda} \geq -C\mathbf{e}. \end{aligned}$$

- Similar to standard SVM dual except the constraint.

Illustrations

- Comparison of the iterative SVM and SVR with different ε's.





Observation: SVR with small ε (e.g. Laplacian loss) works better than those with larger ε's and SVM.

Balance Constraint

The balance constraint, e.g., $-\ell \leq \mathbf{ey} \leq \ell$ is intrinsic in unsupervised/semi-supervised problems.

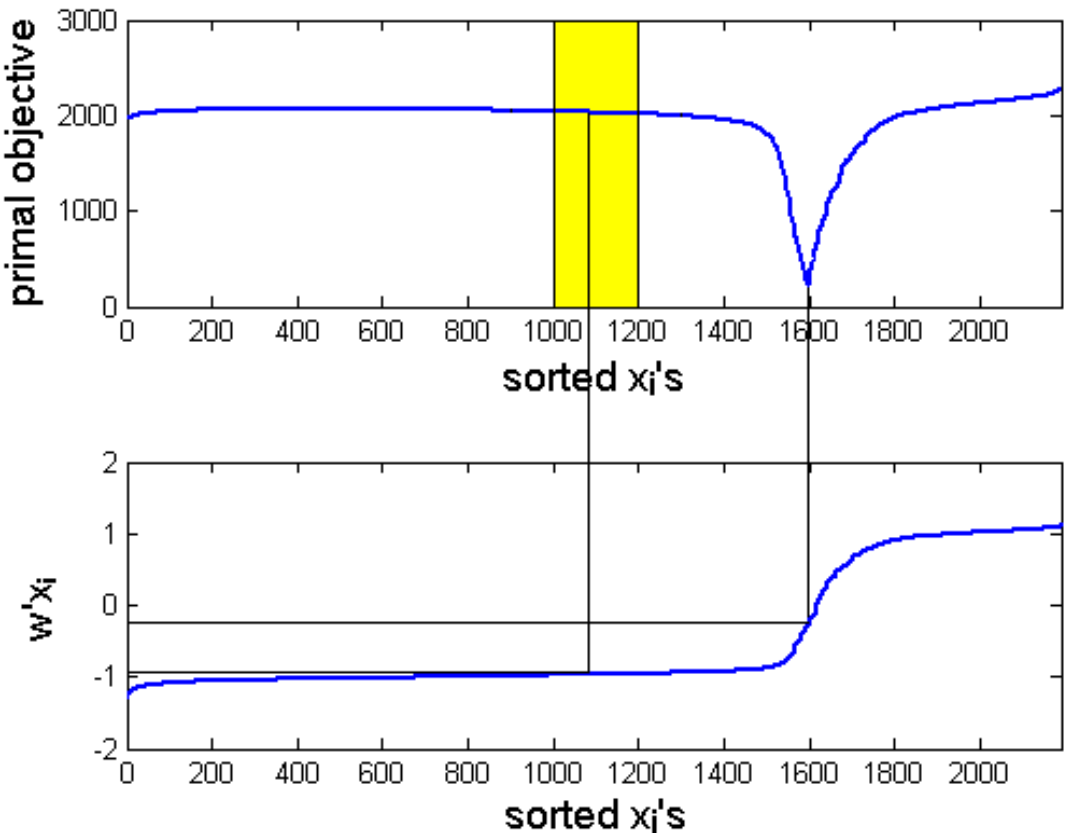
- How to compute b with this constraint?
  1. First solve $\mathbf{w}$ by $\boldsymbol{\lambda}$ from SVR dual;
  2. Fix $\mathbf{w}$, minimize the primal w.r.t. b, i.e.,

$$\begin{aligned} \min_{\mathbf{y},b} \quad & \sum_{i=1}^{n} \left| \mathbf{w}^\top \varphi(\mathbf{x}_i) + b - y_i \right| \\ \text{s.t.} \quad & y_i \in \{\pm 1\}, \ -\ell \leq \mathbf{e}^\top \mathbf{y} \leq \ell. \end{aligned}$$

  3. This can be easily solved by a linear search.

Balance Constraint

- Illustration of choosing the bias under balance constraint



Complete Procedure

Iterative SVR

1: Initialize the labels $\mathbf{y}$.
2: Fix $\mathbf{y}$, and perform SVR with Laplacian loss.
3: Compute $\mathbf{w}$ from the KKT condition.
4: Compute the bias b under balance constraint.
5: Assign the labels as $y_i = \text{sign}(\mathbf{w}^\top \varphi(\mathbf{x}_i) + b)$.
6: Repeat steps 2-5 until convergence.

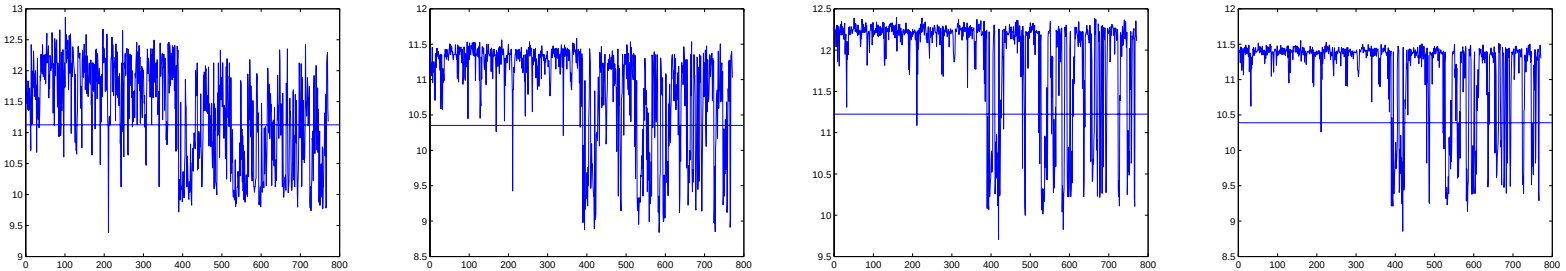
The complexity is much lower than the MMC based on SDP.

complexity	SDP	iterSVR
time	$O(n^7)$	$O(ln^{2.3})$
space	$O(n^{4.5})$	$O(n)$

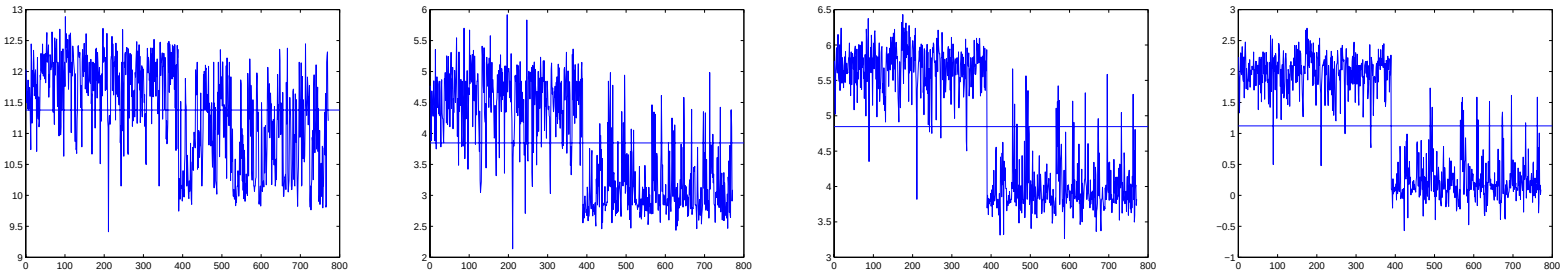
(l: number of iterations; n: sample size)

Illustration of Balance Constraint

- The balance constraint improves the clustering.



Iterative SVR without balance constraint



Iterative SVR with balance constraint

Outline

Experimental Setting

Data set

- UCI repository (ionosphere, digits, letter and satellite);
- LIBSVM data (svmguide1-a and w1b)
- Boost data (ringnorm and image)

Approaches under comparison

1. K-means
2. Normalized cut
3. MMC by [Xu et. al.]
4. GMMC by [Valizadegan and Jin]
5. iterative LS-SVM
6. iterative SVR

Experimental Setting

- K-means as initialization for iterative LS-SVM and SVR.
- Each task repeated 10 times.
  - Seeds of K-means are chosen far away.
  - This makes the initialization more stable.
- Iterative SVR: $C = 500$, $\varepsilon = 0.05$, $\sigma = 2 \sim 5D$ where D is an estimate of the maximum pairwise distance.
- Other approaches: optimal parameter chosen from some candidates.

Clustering Errors (%)

Table: Clustering errors (%) on different data sets/methods.

Data	KM	NC	MMC	GMMC	iterSVM	iterLS-SVM	iterSVR
3-8	5.32 ± 0	35	10	5.6	4.2 ± 0	3.92 ± 0	$\mathbf{3.36 \pm 0}$
1-7	$\mathbf{0.55} \pm 0$	45	31.25	2.2	$\mathbf{0.55} \pm 0$	0.55 ± 0	$\mathbf{0.55} \pm 0$
2-7	3.09 ± 0	34	1.25	0.5	3.09 ± 0	2.25 ± 0	$\mathbf{0.0} \pm 0$
8-9	9.32 ± 0	48	3.75	16.0	9.6 ± 0	8.76 ± 0	$\mathbf{3.67} \pm 0$
iono	32 ± 17.9	25	**21.25**	23.5	31.6 ± 24.9	24.4 ± 2.8	32.3 ± 16.6
w1b	44.1 ± 0	41.8	-	-	39.4 ± 0	-	$\mathbf{36} \pm 0$
svmgd1a	23.5 ± 0	12.5	-	-	23.6 ± 0	$\mathbf{6.8} \pm 0$	$\mathbf{6.8} \pm 0$
letter	17.94 ± 0	23.2	-	-	14.6 ± 0	7.33 ± 0	$\mathbf{7.2} \pm 0$
satellite	4.07 ± 0	4.21	-	-	4.05 ± 0	3.7 ± 0	$\mathbf{3.18} \pm 0$
image	43.5 ± 0	41.2	-	-	38.08 ± 0	41.2 ± 0	$\mathbf{28.6} \pm 0$
ringnorm	24 ± 5.83	22.3	-	-	27.2 ± 6.43	29.8 ± 11	$\mathbf{9.3 \pm 5.87}$

Digits Classification

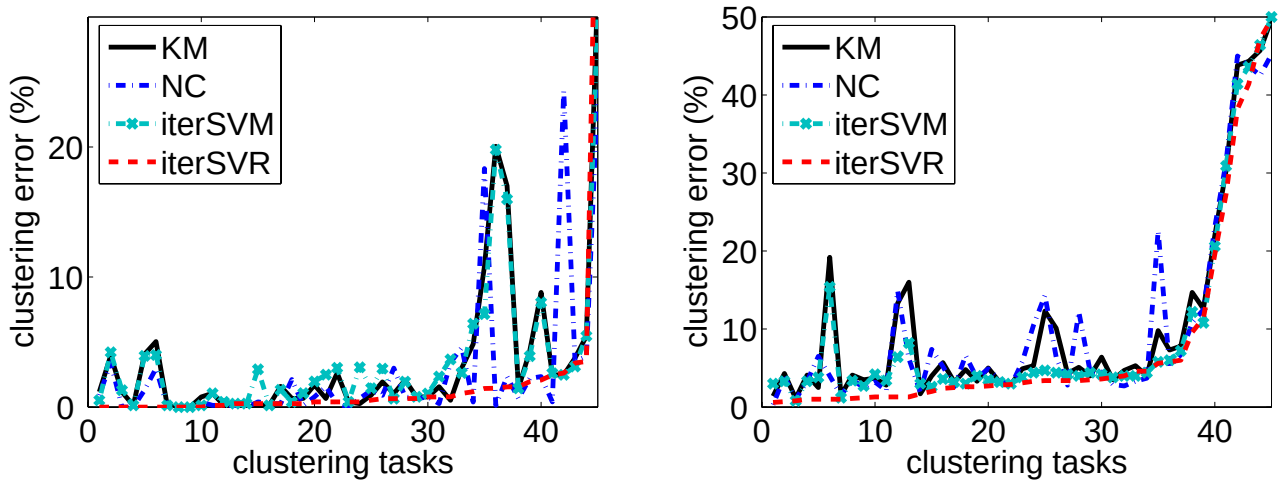


Figure: Comparison using all pairs of UCI and MNIST data sets.

Data	KM	NC	iterSVM	iterSVR
UCI	3.62	2.43	3.49	**1.82**
MNIST	10.79	10.08	9.49	**7.59**

Time Comparison

Table: Wall clock time (in seconds) and clustering error (%).

	Data	NC	MMC	iterSVR
Time	digits 3–9	0.05	1218	0.2 (6090)
	digits 8–9	0.09	957	0.3 (3190)
	digits 2–7	0.08	1079	0.14 (7707)
	ionosphere	0.11	764	0.12 (6367)
Error (%)	digits 3–9	21	7	3
	digits 8–9	9	3	3
	digits 2–7	6	15	8
	ionosphere	29	25	25

Segmentation Results (1)

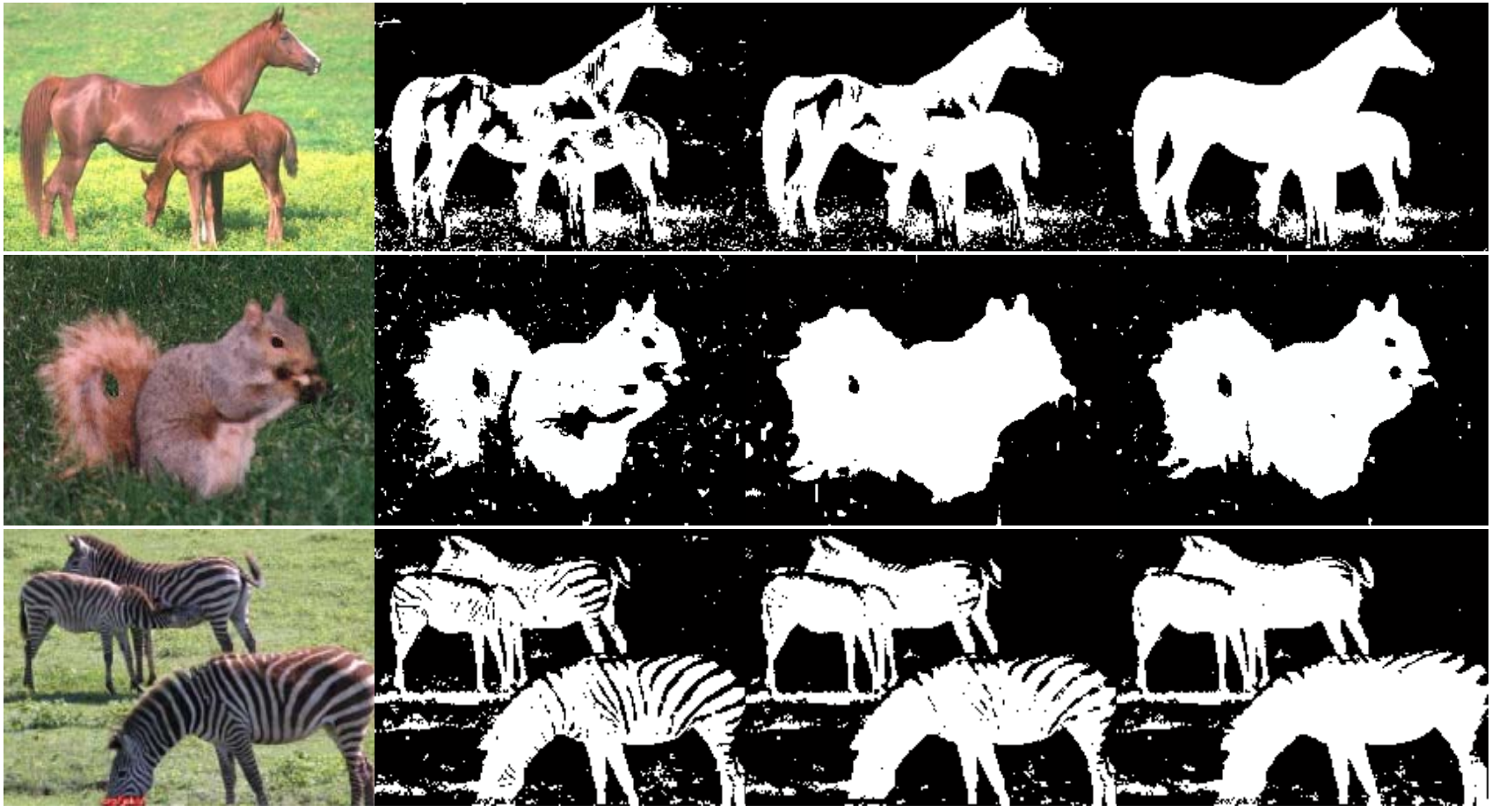


Figure: Recursive segmentation results by iterSVR.

Segmentation Results (2)

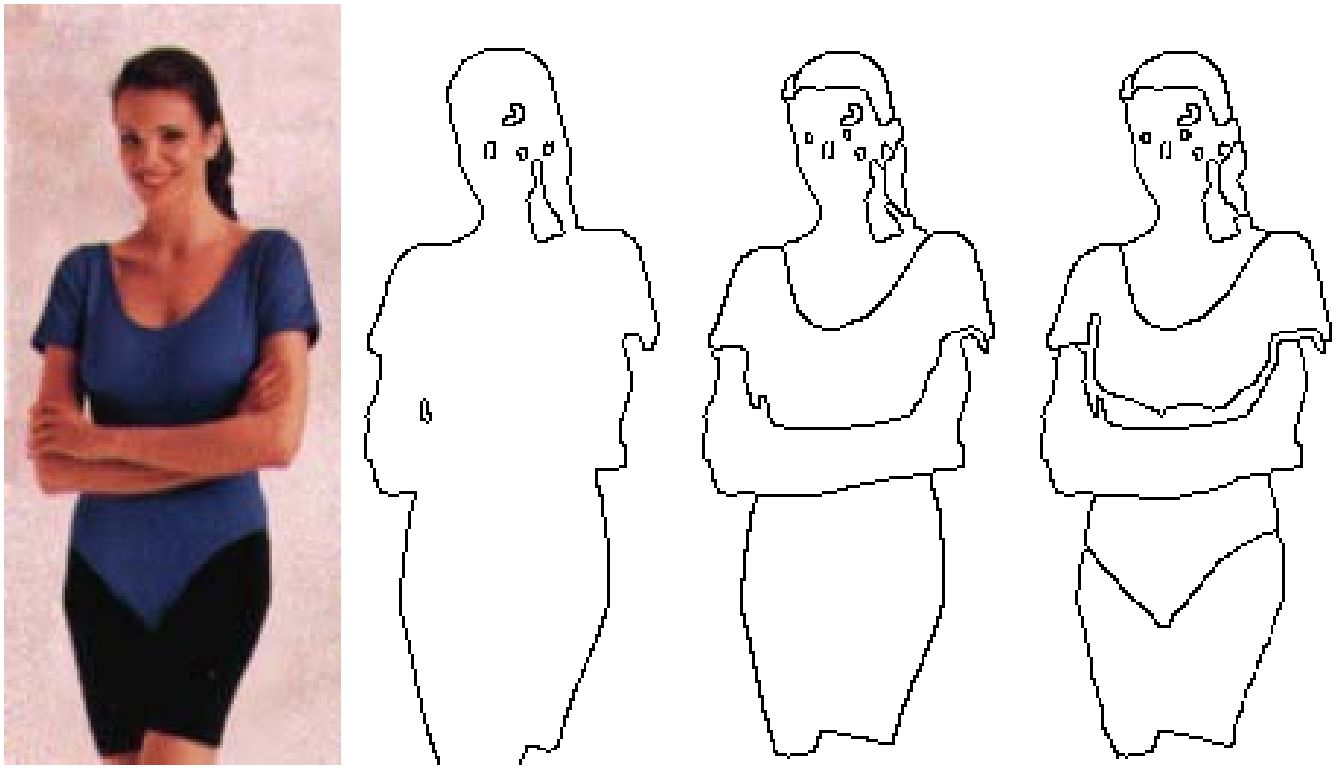


Figure: Recursive segmentation results by iterSVR.

Outline

Conclusions

Summary

- Proposed an efficient approach for maximum margin clustering
- The computation only involves a series of QP's
- The use of Laplacian loss leads to better local optimum

Future Directions

- Extension to multi-class setting
- How to learn the kernel parameters in a principled way