Building Sparse Support Vector Machines for Multi-Instance Classification

Zhouyu Fu, Guojun Lu, Kai Ming Ting, Dengsheng Zhang

Gippsland School of IT, Monash University

September 8, 2011

Outline

Problem Definition

Input

A data set of labelled bags $\{(\mathcal{X}_i, y_i)|i = 1, \ldots, I\}$ with $\mathcal{X}_i = \{\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,n_i}\}$ and $y_i \in \{-1, 1\}$ such that $y_i = \max_{p=1}^{n_i} y_{i,p}$

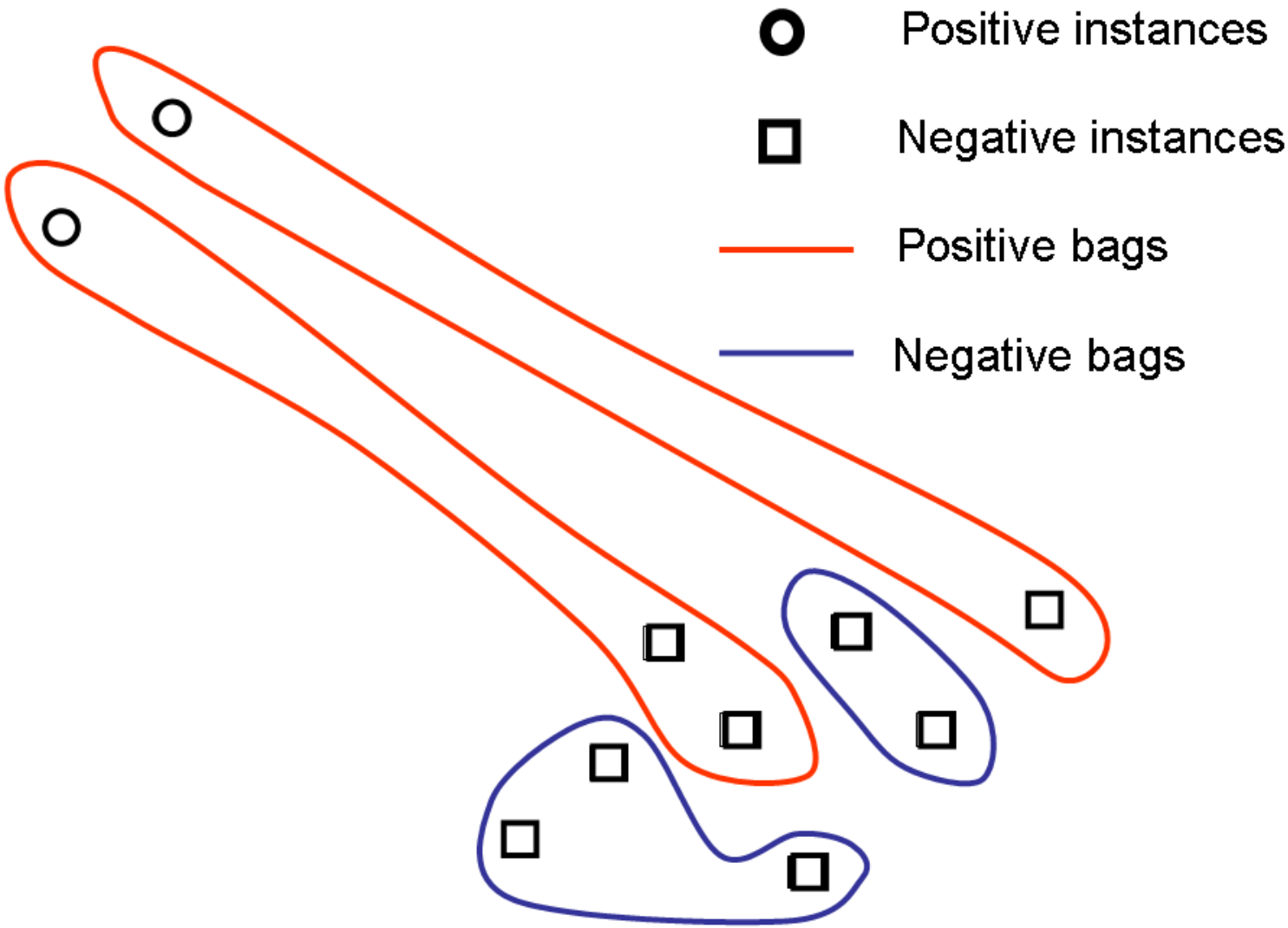
MI Assumption

- A positive bag contains at least a positive instance
- A negative bag contains negative instances only

Output

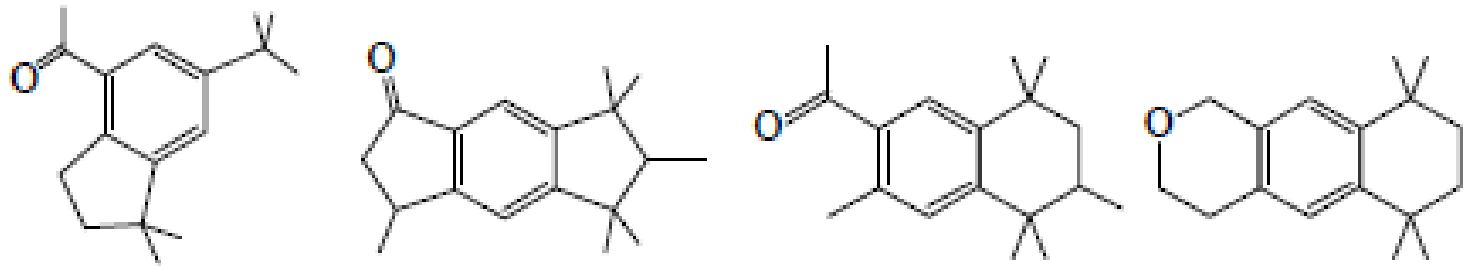
A classifier $f : 2^{\mathbb{R}^d} \rightarrow \mathbb{R}$ for bag-level label prediction that minimizes the bag classification error

An Example of MI Classification



Applications of MI Classification

Drug Activity Prediction



Region-based Image Classification

Prediction Function for Kernel SVM

$$f(\mathbf{x}) = \sum_{i=1,\alpha_i \neq 0}^{N} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) + b$$

- Proportional to the number of SVs ($\mathbf{x}_i$'s with $\alpha_i \neq 0$)
- Number of SVs further depends on the training data size and the complexity of decision boundary
- Desirable to have SVM classifiers with a few SVs when prediction speed is a major concern

Why Sparsity is Important for MI Classification?

- Larger number of feature vectors to deal with as compared to standard classification
- Bag-level prediction usually involves going through all instances in the bag and accumulating the results

Challenge

To build a sparse SVM classifier for MI classification comprising fewer SVs without compromising on performance

"Label-Max" and Existing MI Formulations

MI Assumption

$$y_i = \max_p \ y_{i,p} \qquad y_i \in \{-1, 1\}, y_{i,p} \in \{-1, 1\}$$

This naturally translates to the "Label-Max" constraint in existing MI formulations (MI-SVM, mi-SVM, etc)

$$F(\mathcal{X}_i) = \max_p \ f(\mathbf{x}_{i,p})$$

- Difficult optimization problem (non-differentiable, non-convex)
- Learning sparse MI classifier further complicates the issue, with additional constraints on f

The "Label-Mean" Surrogate

Target Function

$$\min_f \ \|f\|^2 + C \sum_i \ell(F(\mathcal{X}_i), y_i)$$

"Label-Mean" Constraints

$$F(\mathcal{X}_i) = \frac{1}{n_i} \sum_p f(\mathbf{x}_{i,p})$$

Pros and Cons

- Simple optimization problem with optimality guarantee
- Violation of MI assumption, wrong model?

The "Label-Mean" Surrogate

Target Function

$$\min_f \ \|f\|^2 + C \sum_i \ell(F(\mathcal{X}_i), y_i)$$

"Label-Mean" Constraints

$$F(\mathcal{X}_i) = \frac{1}{n_i} \sum_p f(\mathbf{x}_{i,p})$$

Pros and Cons

- Simple optimization problem with optimality guarantee
- Violation of MI assumption, wrong model?
  Not necessarily the case.

Connections with MI-Kernel

Key results

- **Lemma 1**: Training MI-SVM with Label-Mean is equivalent to training a standard SVM at bag level with the normalized set kernel.
- **Theorem 1**: If positive and negative instances are separable with respect to feature map ϕ in the RKHS induced by kernel κ, then for sufficiently large integer r, positive and negative bags are separable using the Label-Mean prediction function with instance-level kernel κ^r.

Intuitions

- If instances are linearly separable, then bags are separable by a polynomial kernel.
- If instances are separable by a Gaussian kernel, then bags are separable by a Gaussian kernel with larger bandwidth.
- Note instance prediction function f is a kernel classifier (a linear classifier is unlikely to work at bag level)
- We used Gaussian kernel in our work

Prediction with "Label-Mean"

Instance classifier

$$f(\mathbf{x}) = \sum_{i=1}^{N} \sum_{p=1}^{|\mathcal{X}_i|} \alpha_i y_i \kappa(\mathbf{x}_{i,p}, \mathbf{x}) + b$$

Bag classifier

$$F(\mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$

The complexity depends on

- size of testing bag
- nonzero coefficients α_i's
- size of training bags with nonzero coefficients

Prediction with "Label-Mean"

$$F(\mathcal{X}) = \frac{1}{|\mathcal{X}|}\sum_{q=1}^{|\mathcal{X}|}\sum_{i=1}^{I}\alpha_i y_i \frac{1}{n_i}\sum_{p=1}^{n_i}\kappa(\mathbf{x}_{i,p},\mathbf{x}_q) + b$$

$$= \frac{1}{|\mathcal{X}|}\sum_{q=1}^{|\mathcal{X}|}\sum_{j=1}^{N}\beta_j\kappa(\mathbf{z}_j,\mathbf{x}_q) + b$$

$$\{\mathbf{z}_1,\ldots,\mathbf{z}_{n_1+1},\ldots,\mathbf{z}_N\} \Longleftrightarrow \{\mathbf{x}_{1,1},\ldots,\mathbf{x}_{2,1},\ldots,\mathbf{x}_{I,n_I}\}$$

$$\{\beta_1,\ldots,\beta_{n_1+1},\ldots,\beta_N\} \Longleftrightarrow \{\frac{1}{n_1}y_1\alpha_1,\ldots,\frac{1}{n_2}y_2\alpha_2,\ldots,\frac{1}{n_I}y_I\alpha_I\}$$

We want to reduce $\mathbf{z}$'s in the expansion!

The Sparse-MI Problem

Objective

Learn $f(.) = \sum_{j=1}^{N_{sv}} \beta_j \kappa(\mathbf{z}_j, .)$ with small N_{sv} for MI classification

Alternatives

- **RSVM** - select N_{sv} $\mathbf{z}_j$'s randomly from training set
- **RS** - approximate a learned classifier $\hat{f}$ by minimizing $\|\hat{f} - f\|^2$
- **MILES** - enforce sparsity on the coefficients using L1 norm (cannot explicitly specify N_{sv})

Proposed Approach

$$\min_{\beta,b,\mathbf{Z}} \; Q(\beta, b, \mathbf{Z}) = \beta^T \mathbf{K}_{\mathbf{Z}} \beta + C \sum_i \ell(y_i, F(\mathcal{X}_i))$$

$$F(\mathcal{X}_i) = \frac{1}{n_i} \sum_{p=1}^{n_i} \sum_{j=1}^{N_{sv}} (\beta_j \kappa(\mathbf{z}_j, \mathbf{x}_{i,p}) + b)$$

- Squared Hinge loss is used $\ell(y, F) = \max(0, 1 - yF)^2$
- Joint learning of classifier weights and SVs in a discriminative fashion
- SVs not necessarily overlap with training instances
- Non-convex, but convex and differentiable in β and b given $\mathbf{Z}$

Optimization Scheme

Reformulation of the optimization problem

$$\min_{\mathbf{Z}} g(\mathbf{Z})$$

$$g(\mathbf{Z}) = \min_{\beta, \boldsymbol{b}} Q(\beta, b, \mathbf{Z})$$

- What's special about $g(\mathbf{Z})$ -
  it depends on the solution of another optimization problem
- Isn't it more difficult to optimize $g(\mathbf{Z})$?

Optimization Scheme

Reformulation of the optimization problem

$$\min_{\mathbf{Z}} g(\mathbf{Z})$$

$$g(\mathbf{Z}) = \min_{\beta, \boldsymbol{b}} Q(\beta, b, \mathbf{Z})$$

- What's special about $g(\mathbf{Z})$ -
  it depends on the solution of another optimization problem
- Isn't it more difficult to optimize $g(\mathbf{Z})$? -
  Answer: Not necessarily!

The Optimal Value Function

For our problem, $g(\mathbf{Z})$ not only exists but is also differentiable

Conditions for differentiability (Bonnans 1998)

- Uniqueness of optimal solution (strict convexity) for β and b given $\mathbf{Z}$
  - unique value for $g(\mathbf{Z})$ at each $\mathbf{Z}$
- Continuous differentiablity of function Q over β and b
  - avoid drastic change of $g(\mathbf{Z})$ over local neighbourhood, i.e. differentiability

Computation of derivatives

We can compute the derivative of $g(\mathbf{Z})$ for each $\mathbf{Z}$ as if it does not depend on β and b

$$\frac{\partial g}{\partial \mathbf{z}_j} = \sum_{i=1}^{N} \overline{\beta}_i \overline{\beta}_j \frac{\partial \kappa(\mathbf{z}_i, \mathbf{z}_j)}{\partial \mathbf{z}_j} + C \sum_{i=1}^{m} \frac{1}{n_i} \ell'(y_i, f_i) \overline{\beta}_j \sum_{p=1}^{n_i} \frac{\partial \kappa(\mathbf{x}_{i,p}, \mathbf{z}_j)}{\partial \mathbf{z}_j}$$

$\overline{\beta}_j$'s are the optimal values of β_j's

Algorithm

Algorithm 1 Sparse SVM for MI Classification

Input: data (B_i, y_i), N_{XV}, λ, $s_{\max}$ and $t_{\max}$
Output: classifier weights β, b and XVs $\mathbf{Z}$
Set $t = 0$ and initialize $\mathbf{Z}$
Solve $\min_{\beta,b} Q(\beta, b, \mathbf{Z}^{(t)})$ and denote the optimizer as $(\beta^{(t)}, b^{(t)})$ and optimal value as $g(\mathbf{Z}^{(t)})$
repeat
  for $s = 1$ **to** $s_{\max}$ **do**
    Set $\mathbf{Z}' = \mathbf{Z}^{(t)} - \lambda \frac{\partial g(\mathbf{Z})}{\partial \mathbf{Z}}$
    Solve $\min_{\beta,b} Q(\beta, b, \mathbf{Z}')$ and denote the optimizer as (β', b') and optimal value as $g(\mathbf{Z}')$
    if $g(\mathbf{Z}') < g(\mathbf{Z}^{(t)})$ **then**
      Set $(\beta^{(t+1)}, b^{(t+1)}) = (\beta', b')$, $\mathbf{Z}^{(t+1)} = \mathbf{Z}'$
      Set $\lambda = 2\lambda$ if s equals 1
      break
    end if
    Set $\lambda = \lambda/2$
  end for
  Set $t = t + 1$
until Convergence or $t \geq t_{\max}$ or $s \geq s_{\max}$

Remarks

- Same optimization strategy has been adopted in solving other problems (SimpleMKL, SKLA)
- Sparse-MI is most related to SKLA with two main differences
  - SKLA can not be used to handle MI classification problems (need a MI formulation with unique optimal solution)
  - SKLA performs optimization and update of SVs in the dual formulation (optimization in the primal is much more efficient)

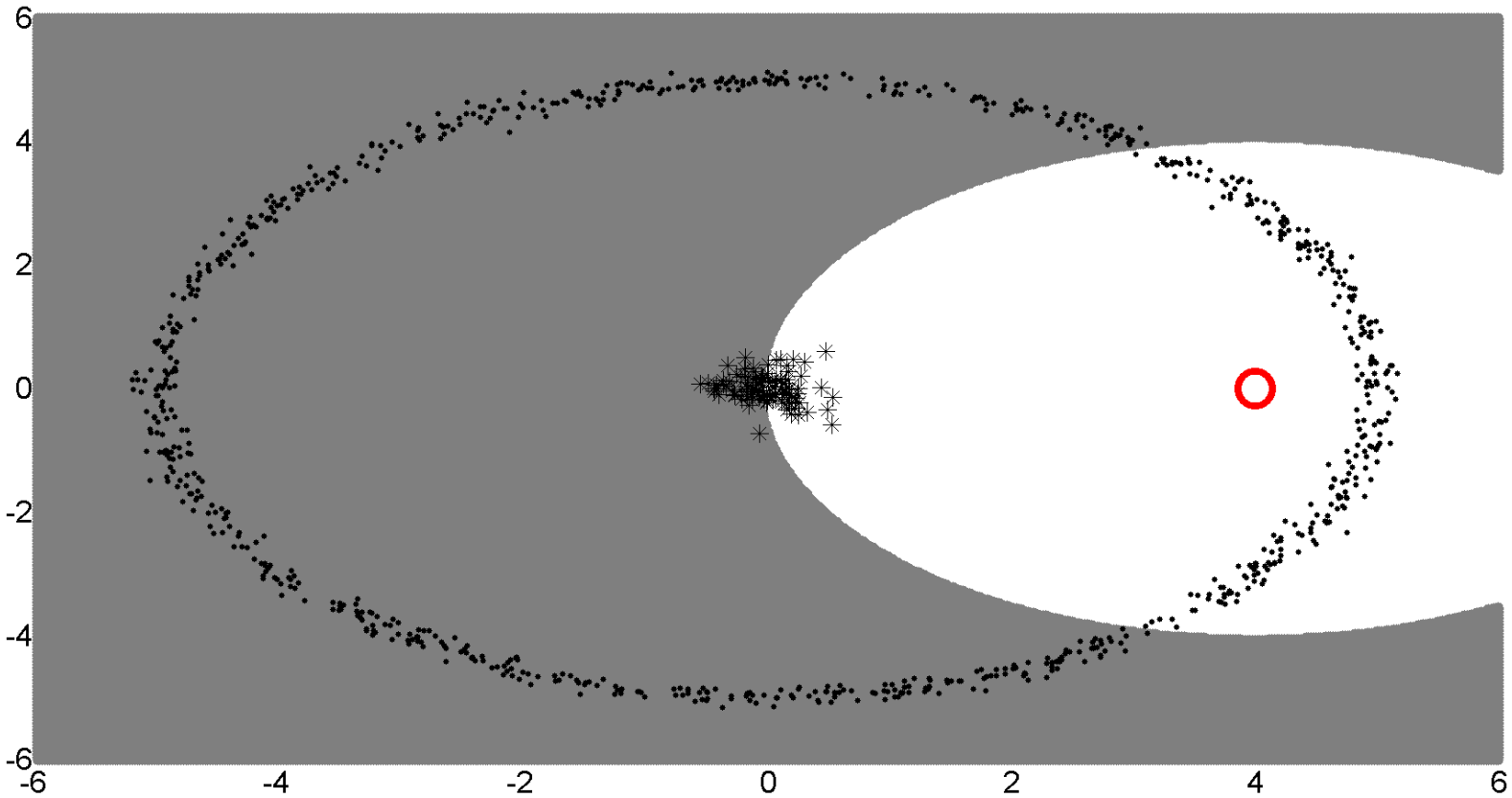
Multi-class Sparse MI Classifier

- One-vs-all scheme - converts to multiple binary MI classification problems
- $\mathbf{z}_j$'s are learned jointly - same XVs for different pairs of classifiers

$$\frac{\partial g}{\partial \mathbf{z}_j} = \sum_{c=1}^{M} \sum_{i=1}^{N_{sv}} \overline{\beta}_i^c \overline{\beta}_j^c \frac{\partial \kappa(\mathbf{z}_i, \mathbf{z}_j)}{\partial \mathbf{z}_j}$$
$$+ C \sum_{c=1}^{M} \sum_{i=1}^{m} \frac{1}{n_i} \ell'(y_i^c, f_i) \overline{\beta}_j^c \sum_{p=1}^{n_i} \frac{\partial \kappa(\mathbf{x}_{i,p}, \mathbf{z}_j)}{\partial \mathbf{z}_j}$$

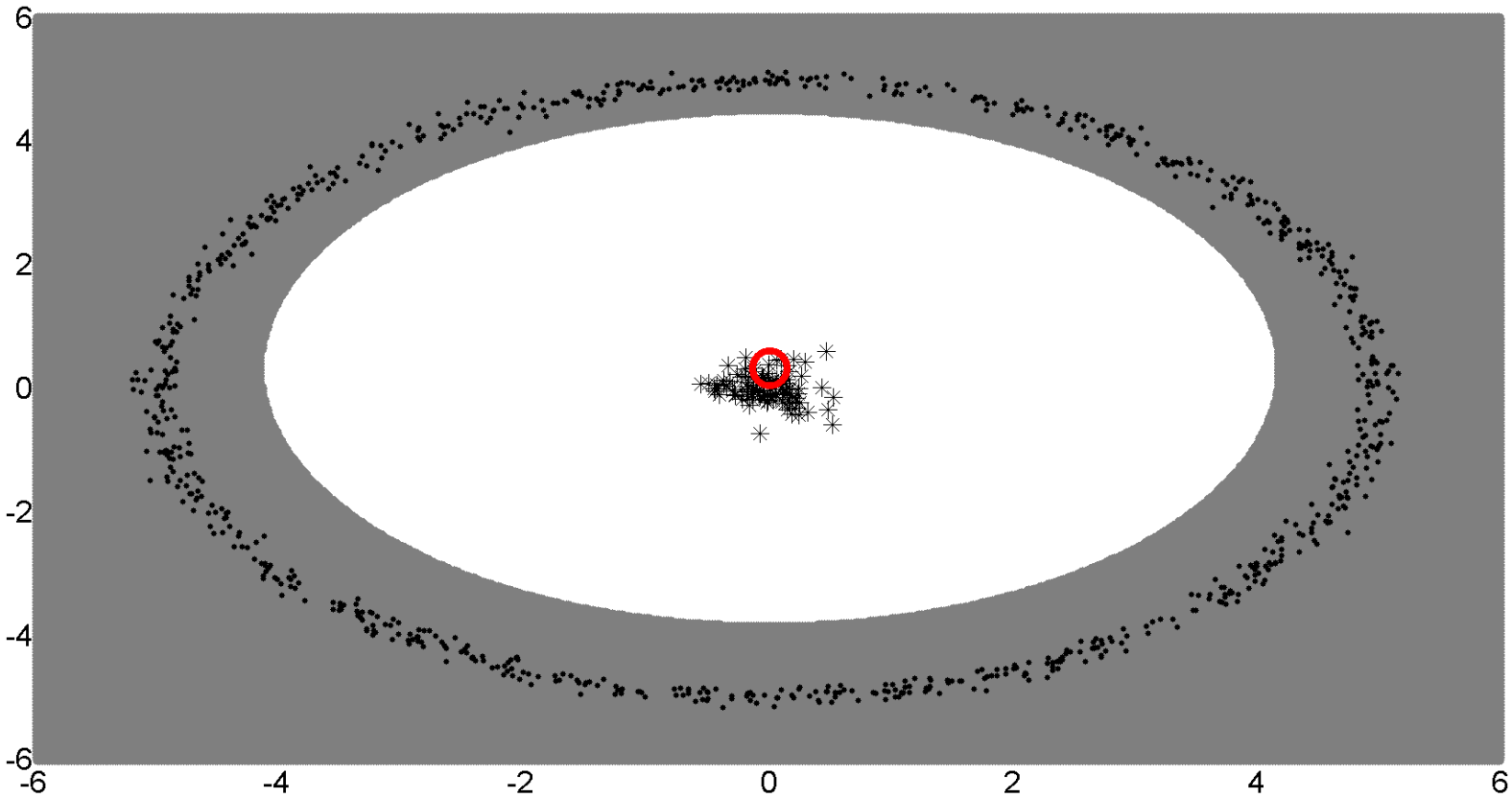
Synthetic Data 1

Data Set



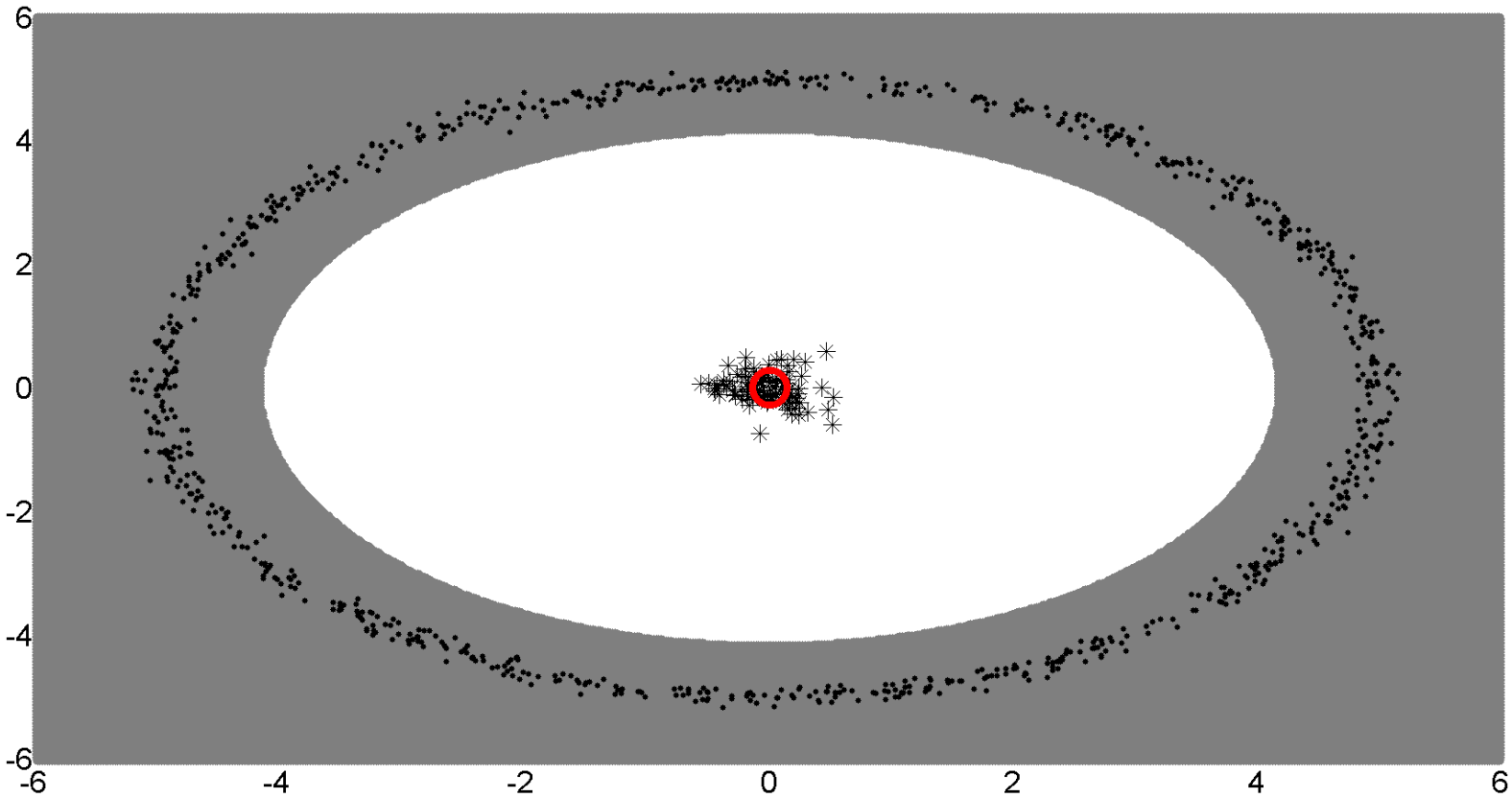
Synthetic Data 1

Iteration 1



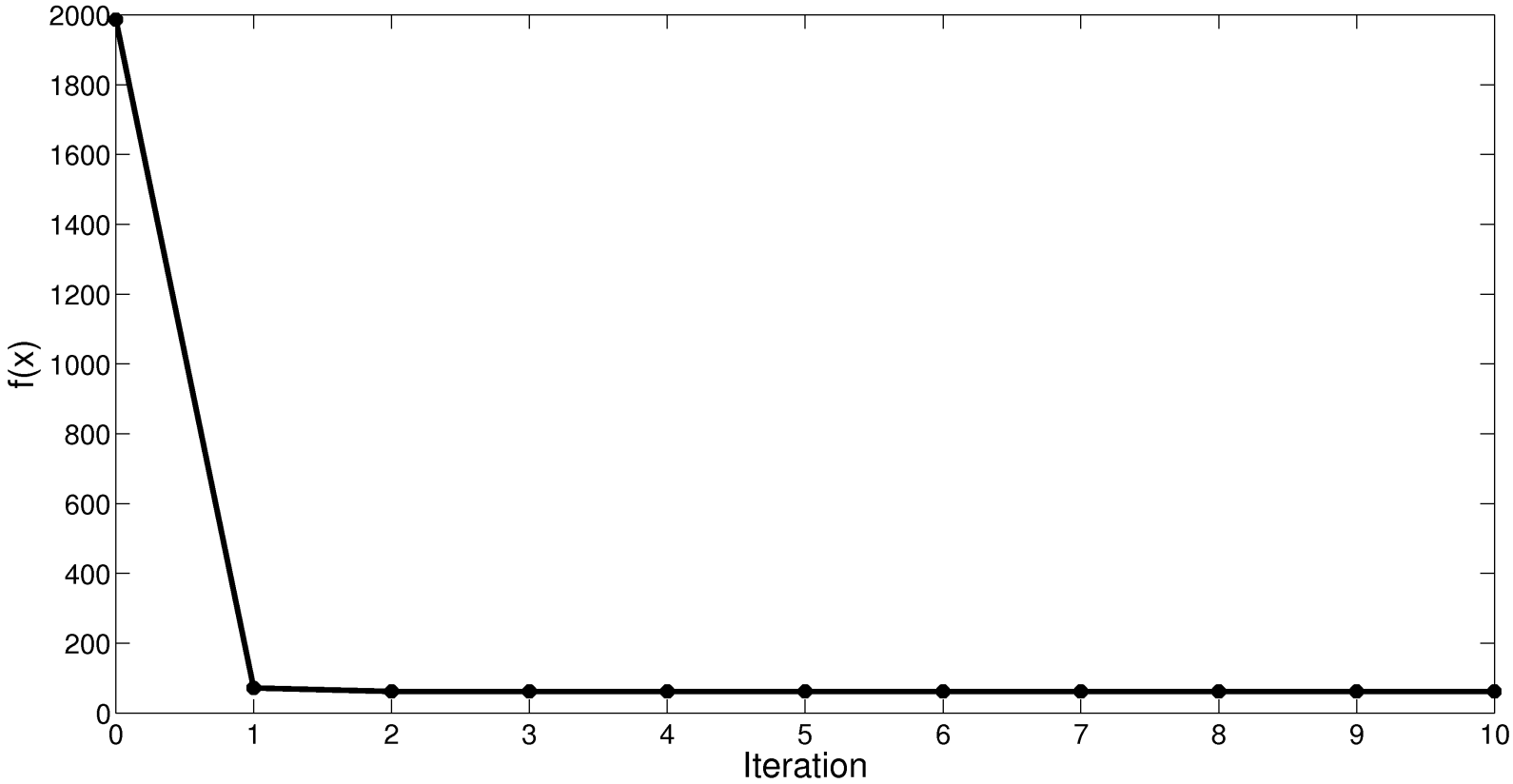
Synthetic Data 1

Iteration 10



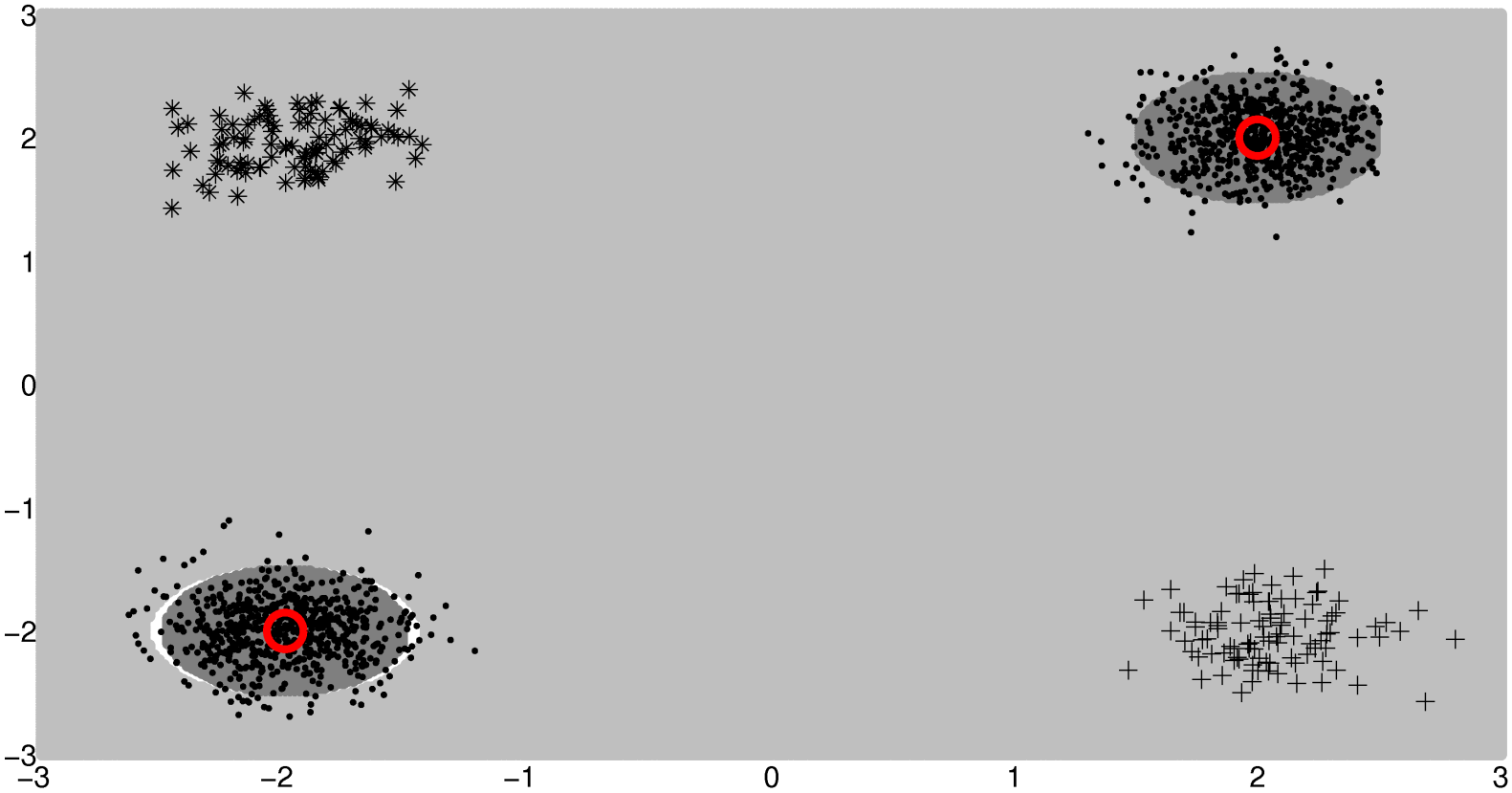
Synthetic Data 1

Function values over iterations



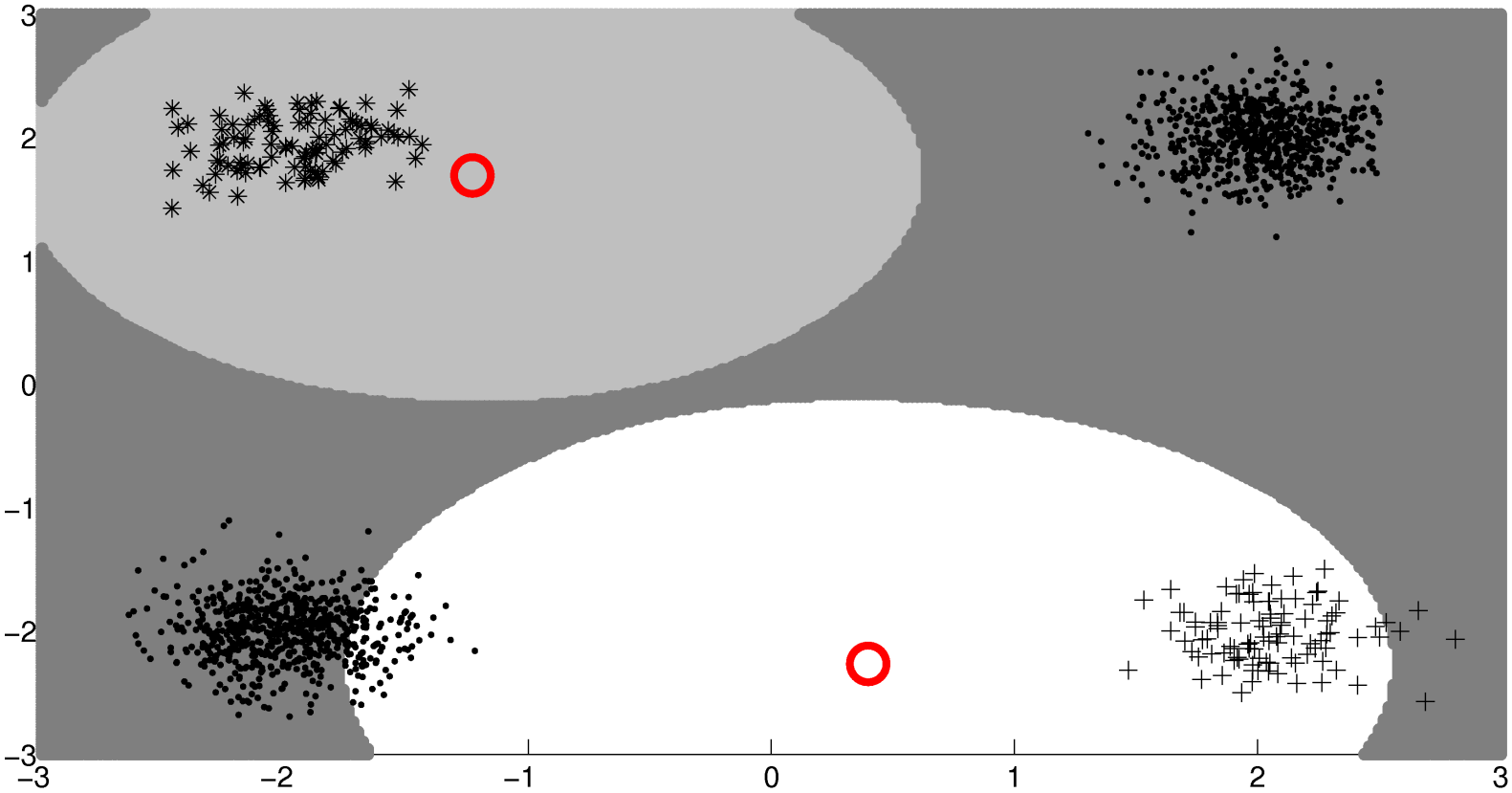
Synthetic Data 2

Data set



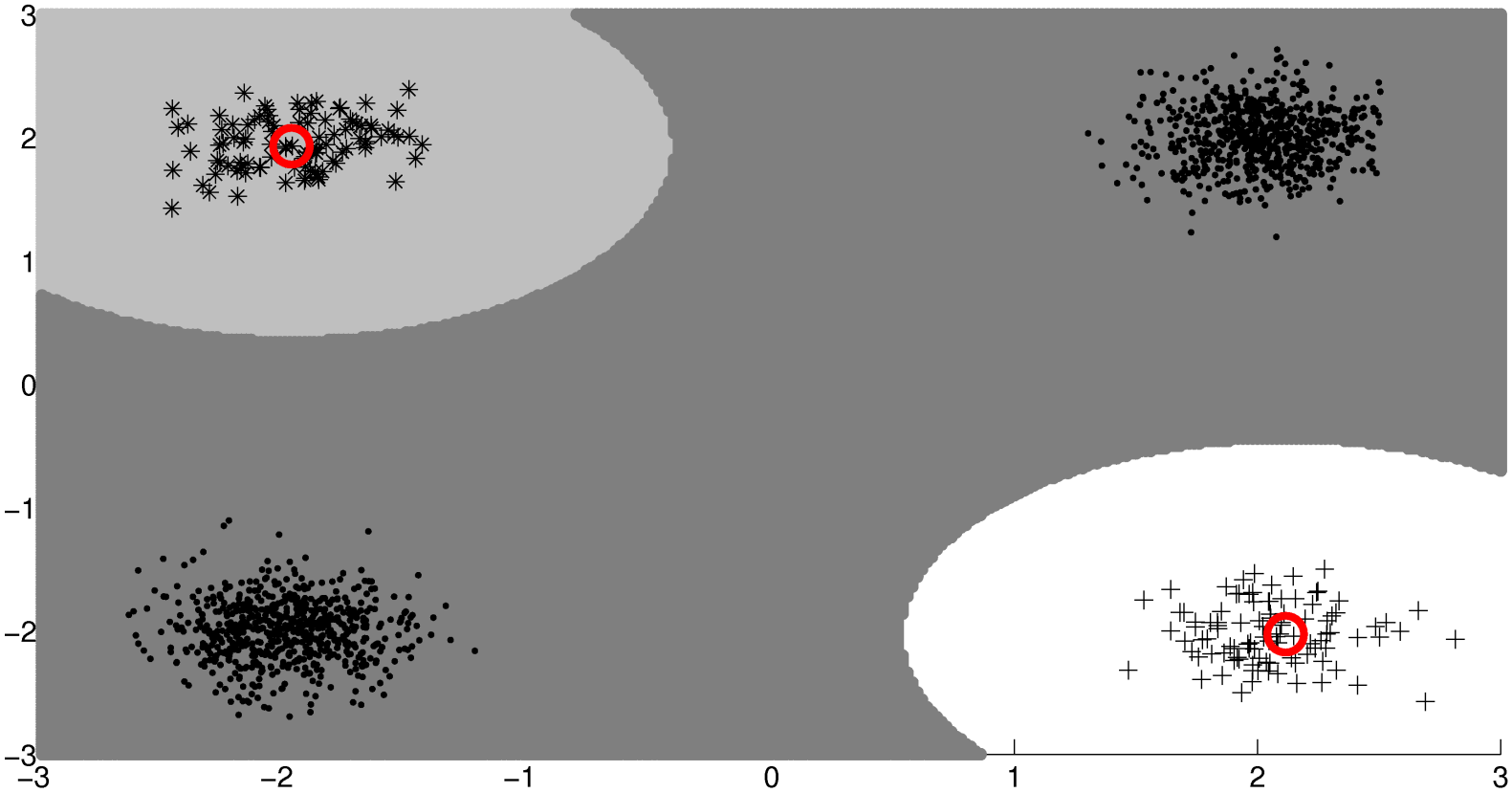
Synthetic Data 2

Iteration 1



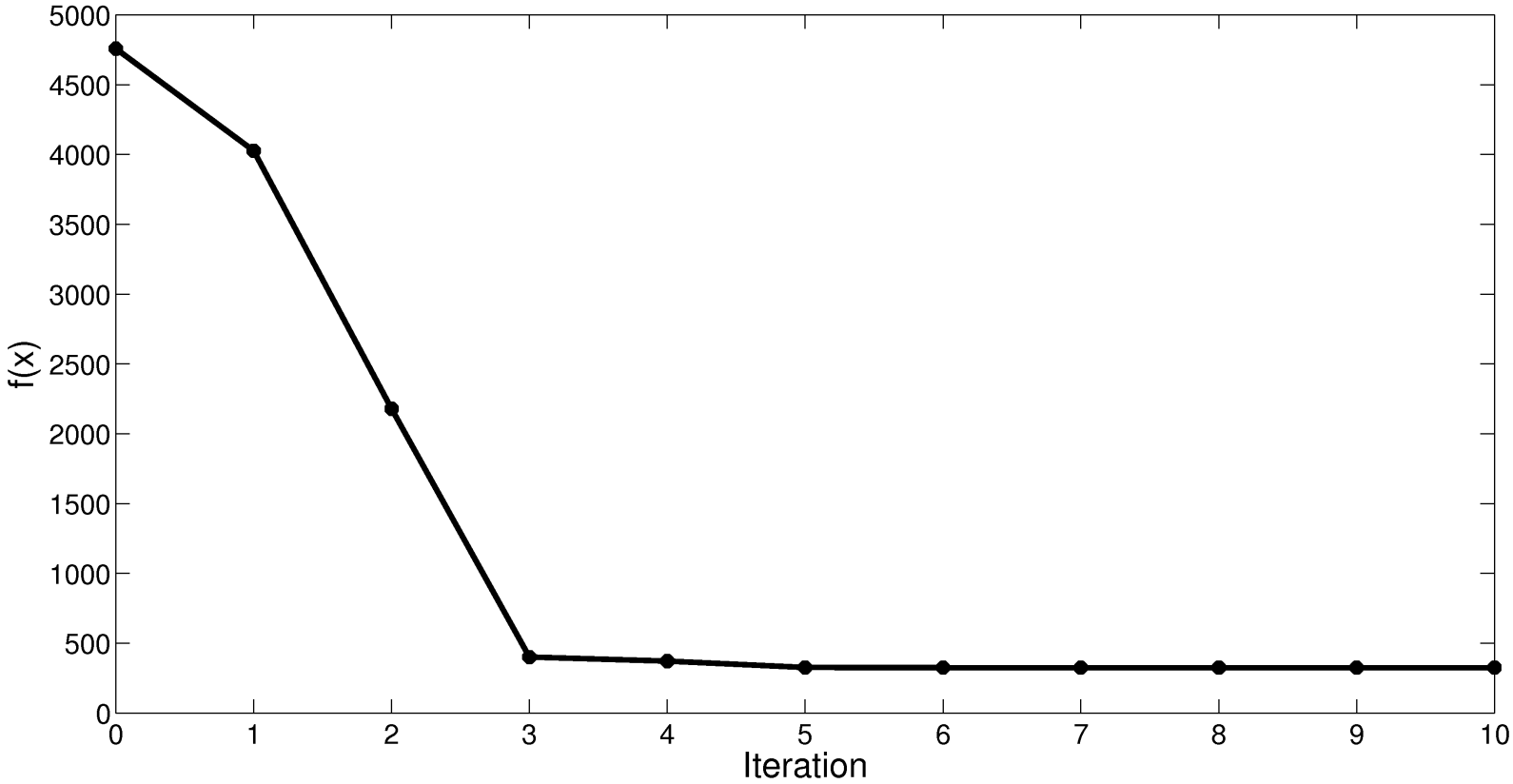
Synthetic Data 2

Iteration 10



Synthetic Data 2

Function values over iterations



Data Set Descriptions

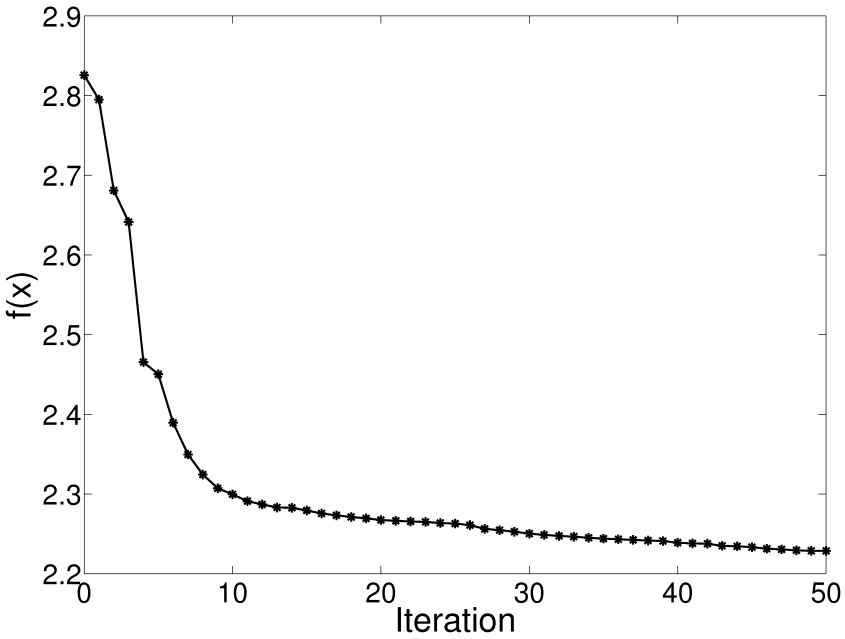
- Drug Actitity Classification
  - MUSK1 - 47 positive and 45 negative, 5.2 instances per bag
  - MUSK2 - 39 positive and 63 negative, 64.7 instances per bag
- Image Classification
  - COREL10 - 10 classes, 100 images per class
  - COREL20 - 20 classes, 100 images per class
  - $2 - 13$ regions per image
- Music Classification
  - GENRE - 10 classes, 100 songs per class, 30 segments per song

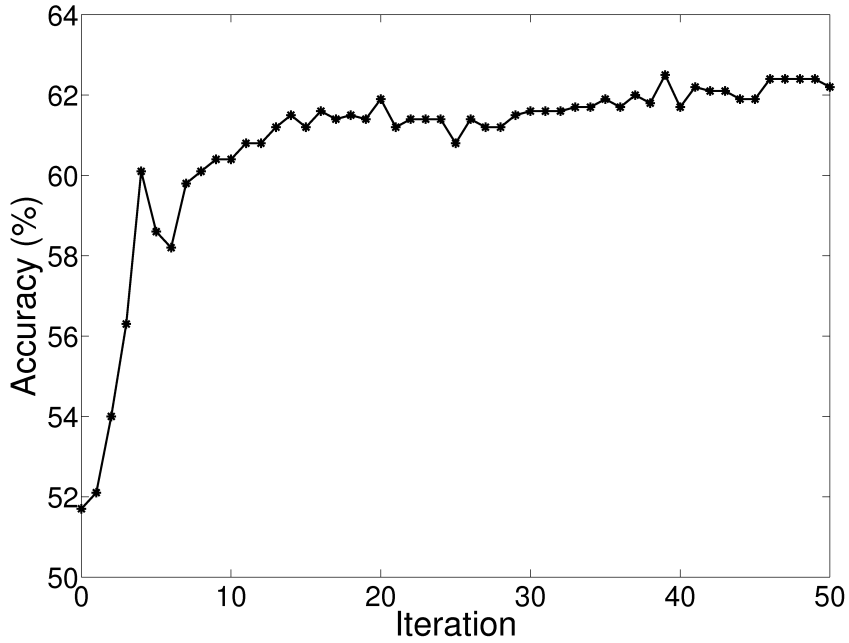
Performance Comparison

Data set		*MUSK1*	*MUSK2*	*COREL10*	*COREL20*	*GENRE*
mi-SVM		87.30% 400.2	80.52% 2029.2	75.62% 1458.2	52.15% 2783.3	80.28% 12439
MI-SVM		77.10% 277.1	83.20% 583.4	74.35% 977	55.37% 2300.1	72.48% 3639.8
MI-Kernel		89.93% 362.4	90.26% 3501	84.30% 1692.6	73.19% 3612.7	77.05% 13844
MILES		85.73% 40.8	87.64% 42.5	82.86% 379.1	69.16% 868	69.87% 1127.9
$N_{sv} = 10$	RS	**88.68%**	86.39%	75.13%	55.20%	54.68%
	RSVM	74.66%	77.70%	69.25%	48.53%	46.01%
	SparseMI	**88.44%**	**88.52%**	**80.10%**	**62.49%**	**71.28%**
$N_{sv} = 100$	RS	**90.18%**	**89.16%**	78.81%	65.35%	67.77%
	RSVM	89.02%	88.26%	77.63%	63.82%	67.41%
	SparseMI	**90.40%**	87.98%	**84.31%**	**72.22%**	**76.06%**

Convergence Results

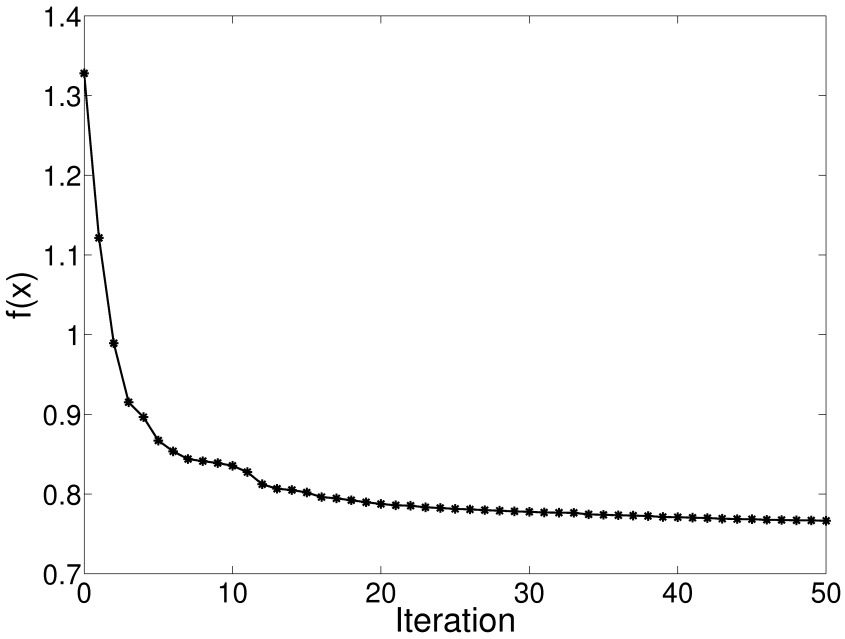
COREL20

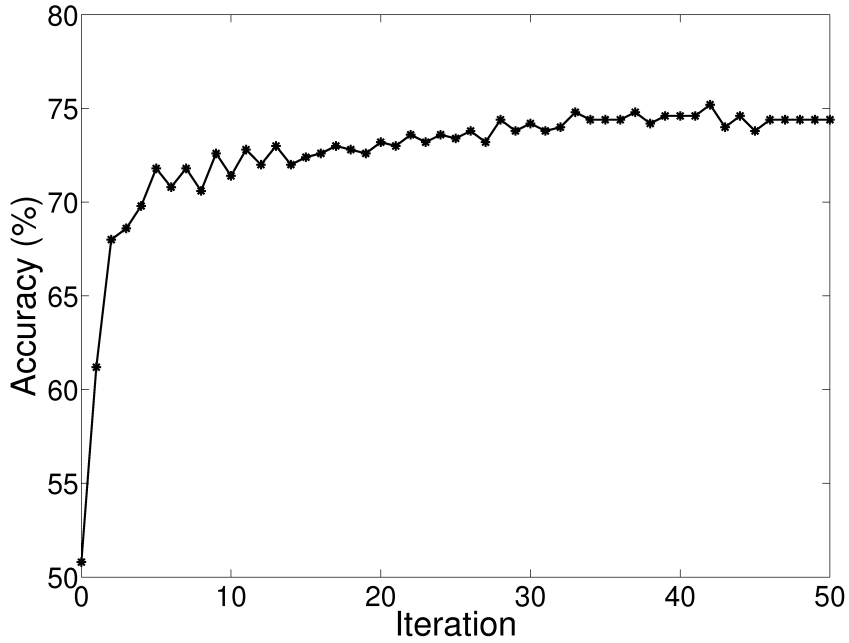




Convergence Results

GENRE





Conclusions

- A sparse SVM classifier proposed for MI classification
- Joint optimization of SVs and classifier weights
- Efficient learning in the primal formulation
- With controlled sparsity
- Comparable performance but much more efficient in testing

Future Work

- Sparse MI classification with non i.i.d. instance distributions
- Incorporation with alternative convex MI classifiers
- Constrained SV selection
- Standard sparse SVM learning from the primal formulation