

# A Game Theoretic Approach to Robust Optimization

Sayak Ray Chowdhury

ME-SSA  
Electrical Engineering  
Indian Institute Of Science

*sayakrc@ee.iisc.ernet.in*

23.06.2015

# Overview

1 Robust SVM

2 Approximate Robust Optimization

3 Approximate Robust SVM

4 Numerical Results

5 Summary

# Introduction : SVM

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|w\|_2^2 \\ & \text{subject to} \quad y_i(w^T x_i + b) \geq 1, \quad \forall i \in [m] \end{aligned}$$

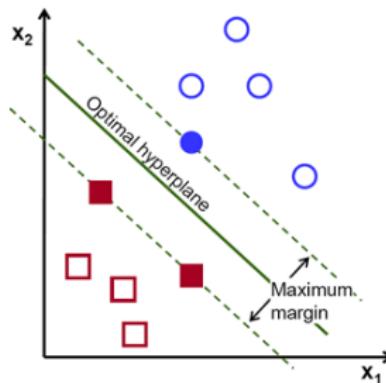


Figure: Hard Margin SVM

# Introduction : SVM

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\ & \text{subject to} \quad y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \forall i \in [m] \end{aligned}$$

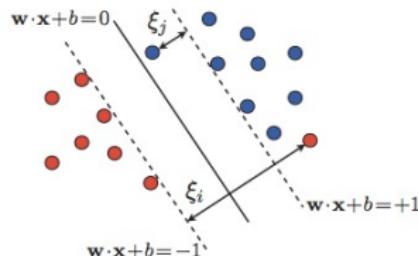
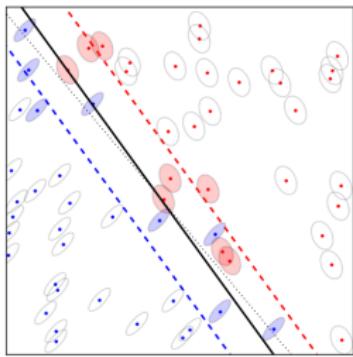


Figure: Soft Margin SVM

# Introduction : SVM

**Problem:** When data is noisy/uncertain ?

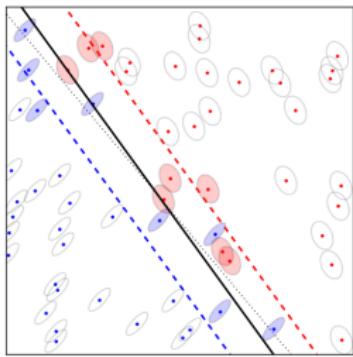
# Uncertainty Model



- $x \in \mathcal{B}(\bar{x}, \Sigma, \gamma)$
- $x : (x - \bar{x})^T \Sigma^{-1} (x - \bar{x}) \leq \gamma^2$

Figure: Ellipsoidal Uncertainty

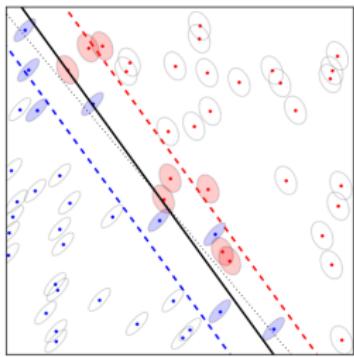
# Uncertainty Model



- $x \in \mathcal{B}(\bar{x}, \Sigma, \gamma)$
- $x : (x - \bar{x})^T \Sigma^{-1} (x - \bar{x}) \leq \gamma^2$
- **constraints:**  
 $y(w^T x + b) \geq 1, \forall x \in \mathcal{B}$

Figure: Ellipsoidal Uncertainty

# Uncertainty Model



- $x \in \mathcal{B}(\bar{x}, \Sigma, \gamma)$
- $x : (x - \bar{x})^T \Sigma^{-1} (x - \bar{x}) \leq \gamma^2$
- **constraints:**  
 $y(w^T x + b) \geq 1, \forall x \in \mathcal{B}$
- **Now solve SVM !**

Figure: Ellipsoidal Uncertainty

# Second Order Cone Programming Formulation

[Bhattacharyya,2005]

$$\text{minimize} \frac{1}{2} \|w\|_2^2$$

**subject to**  $y_i(w^T \bar{x}_i + b) \geq 1 + \gamma_i \|\Sigma_i^{1/2} w\|, \forall i \in [m]$

# Second Order Cone Programming Formulation

[Bhattacharyya,2005]

$$\text{minimize} \frac{1}{2} \|w\|_2^2$$

subject to  $y_i(w^T \bar{x}_i + b) \geq 1 + \gamma_i \|\Sigma_i^{1/2} w\|, \forall i \in [m]$

**robust to data uncertainty**

# Problems !!

- More complex mathematical program
- Do not scale well with the **dimensionality** of the problem
- Current solvers are slow

**Goal:** Solve robust **problem** without solving robust **program**

**Solution:** Borrow techniques from **Online Convex Optimization**  
**[Shalev-shwartz, 2011]**

## Idea: Oracle Based Robust Optimization [Ben-Tal, 2014]

- Solve the robust problem **approximately**
- Invoke the solver of the **original** non-robust problem repeatedly
- Number of iteration depends on the **target accuracy** and **size** of the uncertainty set
- Not on dimensionality of the problem
- **Online Gradient Descent** based primal-dual algorithm

# Online Gradient Descent [Zinkevich, 2003]

At each time:

1. Algo picks decision  $x_t \in \mathcal{K}$ , convex decision set
2. Algo suffers loss  $f_t(x_t)$ , convex function
3. Update  $x_{t+1} \leftarrow x_t - \eta \nabla_x f_t(x_t)$

# Online Gradient Descent [Zinkevich, 2003]

At each time:

1. Algo picks decision  $x_t \in \mathcal{K}$ , convex decision set
2. Algo suffers loss  $f_t(x_t)$ , convex function
3. Update  $x_{t+1} \leftarrow x_t - \eta \nabla_x f_t(x_t)$

$$\text{minimize } \text{Regret} = \sum_{t=1}^T f_t(x_t) - \min_u \sum_{t=1}^T f_t(u)$$

Optimal Regret Bound:  $GD\sqrt{T}$

# Online Gradient Descent [Zinkevich, 2003]

At each time:

1. Algo picks decision  $x_t \in \mathcal{K}$ , convex decision set
2. Algo suffers loss  $f_t(x_t)$ , convex function
3. Update  $x_{t+1} \leftarrow x_t - \eta \nabla_x f_t(x_t)$

$$\text{minimize } \text{Regret} = \sum_{t=1}^T f_t(x_t) - \min_u \sum_{t=1}^T f_t(u)$$

Optimal Regret Bound:  $GD\sqrt{T}$

- $\max_{u,v \in \mathcal{K}} \|u - v\|_2 \leq D$
- $\|\nabla_x f_t(x_t)\|_2 \leq G, \forall t$

# Online Gradient Descent [Zinkevich, 2003]

At each time:

1. Algo picks decision  $x_t \in \mathcal{K}$ , convex decision set
2. Algo suffers loss  $f_t(x_t)$ , convex function
3. Update  $x_{t+1} \leftarrow x_t - \eta \nabla_x f_t(x_t)$

$$\text{minimize } \text{Regret} = \sum_{t=1}^T f_t(x_t) - \min_u \sum_{t=1}^T f_t(u)$$

Optimal Regret Bound:  $GD\sqrt{T}$

- $\max_{u,v \in \mathcal{K}} \|u - v\|_2 \leq D$
- $\|\nabla_x f_t(x_t)\|_2 \leq G, \forall t$

Opposite Online Gradient Ascent

# General Framework

## Feasibility problem

$\exists w \in \mathcal{D} : f_i(w, u_i) \leq 0, \forall i \in [m]$ .  $f_i$  are convex in  $w$  and concave in  $u$ ,  $u_i \in \mathcal{U}$  are fixed noise vectors, both  $\mathcal{D}$  and  $\mathcal{U}$  are convex set.

# General Framework

## Feasibility problem

$\exists w \in \mathcal{D} : f_i(w, u_i) \leq 0, \forall i \in [m]$ .  $f_i$  are convex in  $w$  and concave in  $u$ ,  $u_i \in \mathcal{U}$  are fixed noise vectors, both  $\mathcal{D}$  and  $\mathcal{U}$  are convex set.

## Robust Formulation

$\exists w \in \mathcal{D} : f_i(w, u_i) \leq 0, \forall u_i \in \mathcal{U}, \forall i \in [m]$

# General Framework

## Feasibility problem

$\exists w \in \mathcal{D} : f_i(w, u_i) \leq 0, \forall i \in [m]$ .  $f_i$  are convex in  $w$  and concave in  $u$ ,  $u_i \in \mathcal{U}$  are fixed noise vectors, both  $\mathcal{D}$  and  $\mathcal{U}$  are convex set.

## Robust Formulation

$\exists w \in \mathcal{D} : f_i(w, u_i) \leq 0, \forall u_i \in \mathcal{U}, \forall i \in [m]$

## $\epsilon$ -approximate robust solution

$\exists w \in \mathcal{D} : f_i(w, u_i) \leq \epsilon, \forall u_i \in \mathcal{U}, \forall i \in [m]$ .

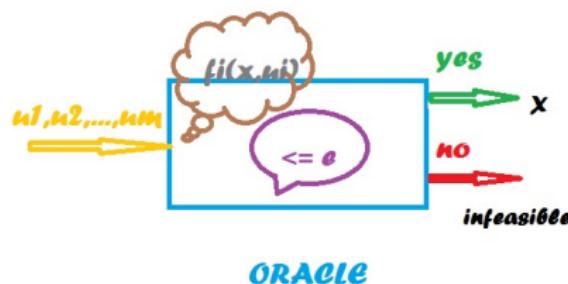
## Two Player Min-Max Game

- minimize w.r.t  $w$
- maximize w.r.t  $u$

$$\min \max f(w, u) \leq \epsilon$$

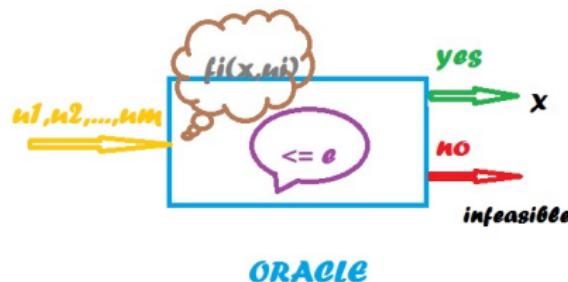
# Primal Dual Algorithm

**Primal Step:** Call the **oracle** using **current** noise vectors



# Primal Dual Algorithm

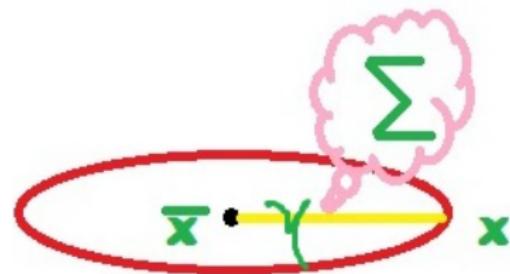
**Primal Step:** Call the **oracle** using **current** noise vectors



**Dual Step:** Generate noise samples via **OGD**

$$u^{t+1} \leftarrow \prod_{\mathcal{U}} [u^t + \eta \bigtriangledown_u f(x^t, u^t)]$$

# Back to SVM



$$(x - \bar{x})^T \Sigma^{-1} (x - \bar{x}) \leq \gamma^2$$

**robust constraint:**

$$y(w^T \bar{x} + b) \geq 1 + \gamma \|\Sigma^{1/2} w\|$$

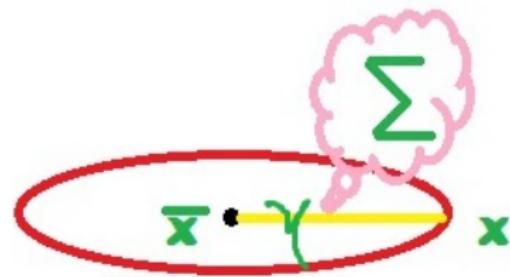


$$x = \bar{x} + \gamma \Sigma^{1/2} u$$

**robust constraint:**

$$y\{w^T(\bar{x} + \gamma \Sigma^{1/2} u) + b\} \geq 1, \forall u$$

## Back to SVM



$$(x - \bar{x})^T \Sigma^{-1} (x - \bar{x}) \leq \gamma^2$$

**robust constraint:**

$$y(w^T \bar{x} + b) \geq 1 + \gamma \|\Sigma^{1/2} w\|$$

$$x = \bar{x} + \gamma \Sigma^{1/2} u$$

**robust constraint:**

$$y\{w^T(\bar{x} + \gamma \Sigma^{1/2} u) + b\} \geq 1, \forall u$$

## Algorithm $\mathcal{A}$ : OGD-based SVM

**Input:** Tuples  $(\bar{x}_i, \Sigma_i, \gamma_i)$ ,  $\forall i \in [m]$ , target accuracy  $\epsilon > 0$

## Algorithm $\mathcal{A}$ : OGD-based SVM

**Input:** Tuples  $(\bar{x}_i, \Sigma_i, \gamma_i)$ ,  $\forall i \in [m]$ , target accuracy  $\epsilon > 0$

1. **Set**  $D = 2$ ,  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$ ,  $T = \lceil \frac{G^2 D^2}{\epsilon^2} \rceil$  and  $\eta = \frac{D}{G\sqrt{T}}$
2. **Initialize**  $(u_1^0, \dots, u_m^0)$  and  $w_0$  arbitrarily

## Algorithm $\mathcal{A}$ : OGD-based SVM

**Input:** Tuples  $(\bar{x}_i, \Sigma_i, \gamma_i)$ ,  $\forall i \in [m]$ , target accuracy  $\epsilon > 0$

1. **Set**  $D = 2$ ,  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$ ,  $T = \lceil \frac{G^2 D^2}{\epsilon^2} \rceil$  and  $\eta = \frac{D}{G\sqrt{T}}$
2. **Initialize**  $(u_1^0, \dots, u_m^0)$  and  $w_0$  arbitrarily
3. **Repeat:** a. **Dual:**

Update  $u_i^t \leftarrow \frac{u_i^{t-1} - \eta y_i \gamma_i \Sigma_i^{1/2} w_{t-1}}{\max\{\|u_i^{t-1} - \eta y_i \gamma_i \Sigma_i^{1/2} w_{t-1}\|_2, 1\}}$

compute  $x_i^t = \bar{x}_i + \gamma_i \Sigma_i^{1/2} u_i^t$

b. **Primal:** Call the *oracle*

if Oracle declared infeasibility then return *infeasible*

else set  $w_t \leftarrow \frac{w_t}{\max\{\|w_t\|_2, 1\}}$

# Algorithm $\mathcal{A}$ : OGD-based SVM

**Input:** Tuples  $(\bar{x}_i, \Sigma_i, \gamma_i)$ ,  $\forall i \in [m]$ , target accuracy  $\epsilon > 0$

1. **Set**  $D = 2$ ,  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$ ,  $T = \lceil \frac{G^2 D^2}{\epsilon^2} \rceil$  and  $\eta = \frac{D}{G\sqrt{T}}$
2. **Initialize**  $(u_1^0, \dots, u_m^0)$  and  $w_0$  arbitrarily
3. **Repeat:** a. **Dual:**

Update  $u_i^t \leftarrow \frac{u_i^{t-1} - \eta y_i \gamma_i \Sigma_i^{1/2} w_{t-1}}{\max\{\|u_i^{t-1} - \eta y_i \gamma_i \Sigma_i^{1/2} w_{t-1}\|_2, 1\}}$

compute  $x_i^t = \bar{x}_i + \gamma_i \Sigma_i^{1/2} u_i^t$

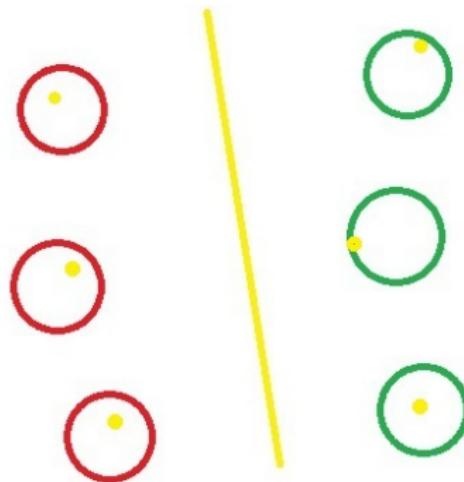
b. **Primal:** Call the *oracle*

if Oracle declared infeasibility then return *infeasible*

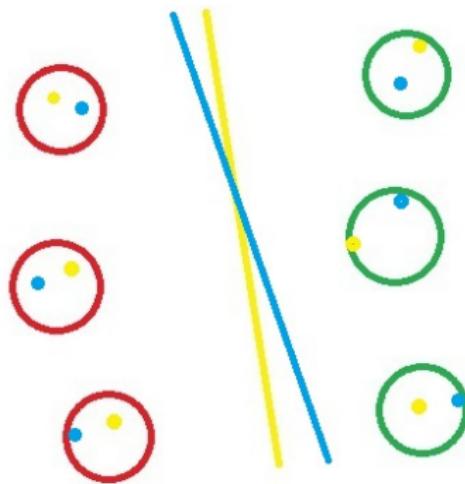
else set  $w_t \leftarrow \frac{w_t}{\max\{\|w_t\|_2, 1\}}$

4. **Return**  $\bar{w} = \frac{1}{T} \sum_{t=1}^T w_t$ ,  $\bar{b} = \frac{1}{T} \sum_{t=1}^T b_t$

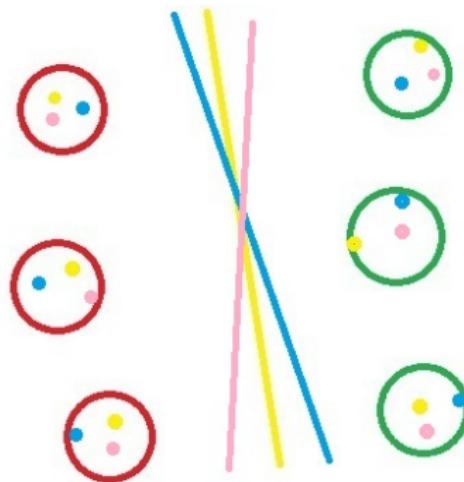
# How The Algorithm Works?



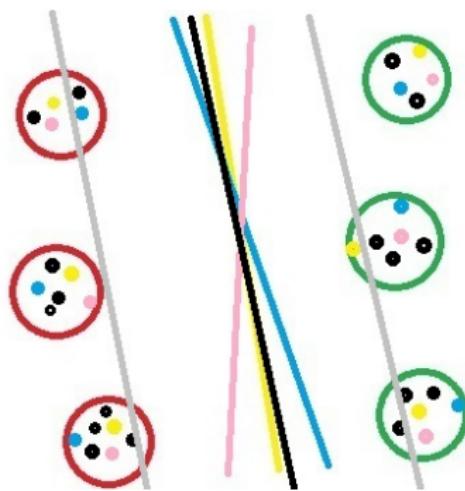
# How The Algorithm Works?



# How The Algorithm Works?



# How The Algorithm Works?



# Result

## Theorem

Algorithm  $\mathcal{A}$  returns an  $2\epsilon$ -approximate robust solution, that is  $y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall i \in [m]$ , after at most  $T = O(G^2/\epsilon^2)$  calls to the SVM-oracle, where  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$

# Result

## Theorem

Algorithm  $\mathcal{A}$  returns an  $2\epsilon$ -approximate robust solution, that is  
 $y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall i \in [m]$ , after at most  $T = O(G^2/\epsilon^2)$  calls to the SVM-oracle, where  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$

## Proof Sketch

1  $\| \nabla_w f_i(w, u) \|_2^2 = \gamma_i^2 w^T \Sigma_i w \leq \gamma_i^2 \lambda_{\max}(\Sigma_i) = G^2$

# Result

## Theorem

Algorithm  $\mathcal{A}$  returns an  $2\epsilon$ -approximate robust solution, that is  
 $y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall i \in [m]$ , after at most  $T = O(G^2/\epsilon^2)$  calls to the SVM-oracle, where  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$

## Proof Sketch

- ①  $\|\nabla_w f_i(w, u)\|_2^2 = \gamma_i^2 w^T \Sigma_i w \leq \gamma_i^2 \lambda_{\max}(\Sigma_i) = G^2$
- ②  $y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$  for all  $t \in [T]$  and  $i \in [m]$

# Result

## Theorem

Algorithm  $\mathcal{A}$  returns an  $2\epsilon$ -approximate robust solution, that is  
 $y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall i \in [m]$ , after at most  $T = O(G^2/\epsilon^2)$  calls to the SVM-oracle, where  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$

## Proof Sketch

①  $\|\nabla_w f_i(w, u)\|_2^2 = \gamma_i^2 w^T \Sigma_i w \leq \gamma_i^2 \lambda_{\max}(\Sigma_i) = G^2$

②  $y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$  for all  $t \in [T]$  and  $i \in [m]$

③  $\forall i \in [m], \frac{1}{T} \sum_{t=1}^T y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$

# Result

## Theorem

Algorithm  $\mathcal{A}$  returns an  $2\epsilon$ -approximate robust solution, that is  
 $y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall i \in [m]$ , after at most  $T = O(G^2/\epsilon^2)$  calls to the SVM-oracle, where  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$

## Proof Sketch

- ①  $\|\nabla_u f_i(w, u)\|_2^2 = \gamma_i^2 w^T \Sigma_i w \leq \gamma_i^2 \lambda_{\max}(\Sigma_i) = G^2$
- ②  $y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$  for all  $t \in [T]$  and  $i \in [m]$
- ③  $\forall i \in [m], \frac{1}{T} \sum_{t=1}^T y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$
- ④  $\max_{x_i} \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i + b_t)] - \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i^t + b_t)] \leq \frac{GD}{\sqrt{T}} \leq \epsilon$

# Result

## Theorem

Algorithm  $\mathcal{A}$  returns an  $2\epsilon$ -approximate robust solution, that is  
 $y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall i \in [m]$ , after at most  $T = O(G^2/\epsilon^2)$  calls to the SVM-oracle, where  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$

## Proof Sketch

- ①  $\|\nabla_u f_i(w, u)\|_2^2 = \gamma_i^2 w^T \Sigma_i w \leq \gamma_i^2 \lambda_{\max}(\Sigma_i) = G^2$
- ②  $y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$  for all  $t \in [T]$  and  $i \in [m]$
- ③  $\forall i \in [m], \frac{1}{T} \sum_{t=1}^T y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$
- ④  $\max_{x_i} \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i + b_t)] - \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i^t + b_t)] \leq \frac{GD}{\sqrt{T}} \leq \epsilon$
- ⑤  $\epsilon \geq \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i^t + b_t)] \geq \max_{x_i} \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i + b_t)] - \epsilon$

# Result

## Theorem

Algorithm  $\mathcal{A}$  returns an  $2\epsilon$ -approximate robust solution, that is

$y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall i \in [m]$ , after at most  $T = O(G^2/\epsilon^2)$  calls to the SVM-oracle, where  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$

## Proof Sketch

- ①  $\|\nabla_u f_i(w, u)\|_2^2 = \gamma_i^2 w^T \Sigma_i w \leq \gamma_i^2 \lambda_{\max}(\Sigma_i) = G^2$
- ②  $y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$  for all  $t \in [T]$  and  $i \in [m]$
- ③  $\forall i \in [m]$ ,  $\frac{1}{T} \sum_{t=1}^T y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$
- ④  $\max_{x_i} \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i + b_t)] - \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i^t + b_t)] \leq \frac{GD}{\sqrt{T}} \leq \epsilon$
- ⑤  $\epsilon \geq \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i^t + b_t)] \geq \max_{x_i} \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i + b_t)] - \epsilon$
- ⑥  $\epsilon \geq \max_{x_i} [1 - y_i(\bar{w}^T x_i + \bar{b})] - \epsilon$

# Result

## Theorem

Algorithm  $\mathcal{A}$  returns an  $2\epsilon$ -approximate robust solution, that is

$y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall i \in [m]$ , after at most  $T = O(G^2/\epsilon^2)$  calls to the SVM-oracle, where  $G = \sqrt{\max_{i=1}^m (\gamma_i^2 \lambda_{\max}(\Sigma_i))}$

## Proof Sketch

- ①  $\|\nabla_u f_i(w, u)\|_2^2 = \gamma_i^2 w^T \Sigma_i w \leq \gamma_i^2 \lambda_{\max}(\Sigma_i) = G^2$
- ②  $y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$  for all  $t \in [T]$  and  $i \in [m]$
- ③  $\forall i \in [m]$ ,  $\frac{1}{T} \sum_{t=1}^T y_i(w_t^T x_i^t + b_t) \geq 1 - \epsilon$
- ④  $\max_{x_i} \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i + b_t)] - \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i^t + b_t)] \leq \frac{GD}{\sqrt{T}} \leq \epsilon$
- ⑤  $\epsilon \geq \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i^t + b_t)] \geq \max_{x_i} \frac{1}{T} \sum_{t=1}^T [1 - y_i(w_t^T x_i + b_t)] - \epsilon$
- ⑥  $\epsilon \geq \max_{x_i} [1 - y_i(\bar{w}^T x_i + \bar{b})] - \epsilon$
- ⑦  $y_i(\bar{w}^T x_i + \bar{b}) \geq 1 - 2\epsilon$ ,  $\forall x_i \in \mathcal{B}, \forall i$

# Error Measures [Shivaswamy, 2006]

## Worst Case Error

$x \in \mathcal{B}(\bar{x}, \Sigma, \gamma)$  has true label  $y$

$$e_{wc}(\mathcal{B}) = 1, \text{ if } yz \leq \gamma$$

$$z = \frac{w^T \bar{x} + b}{\sqrt{w^T \Sigma w}}$$

## Expected error

$$e_{exp} = \frac{Vol(Misclassified)}{Vol(\mathcal{B})}$$

# Error Measures [Shivaswamy, 2006]

## Worst Case Error

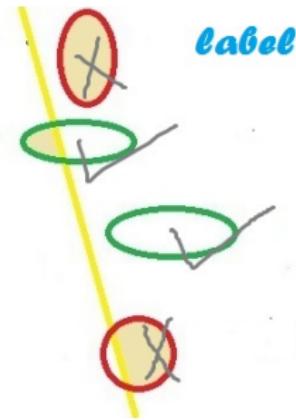
$x \in \mathcal{B}(\bar{x}, \Sigma, \gamma)$  has true label  $y$

$$e_{wc}(\mathcal{B}) = 1, \text{ if } yz \leq \gamma$$

$$z = \frac{w^T \bar{x} + b}{\sqrt{w^T \Sigma w}}$$

## Expected error

$$e_{exp} = \frac{Vol(Misclassified)}{Vol(\mathcal{B})}$$



# Error Measures [Shivaswamy, 2006]

## Worst Case Error

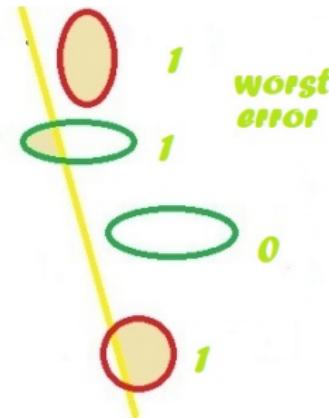
$x \in \mathcal{B}(\bar{x}, \Sigma, \gamma)$  has true label  $y$

$$e_{wc}(\mathcal{B}) = 1, \text{ if } yz \leq \gamma$$

$$z = \frac{w^T \bar{x} + b}{\sqrt{w^T \Sigma w}}$$

## Expected error

$$e_{exp} = \frac{Vol(Misclassified)}{Vol(\mathcal{B})}$$



# Error Measures [Shivaswamy, 2006]

## Worst Case Error

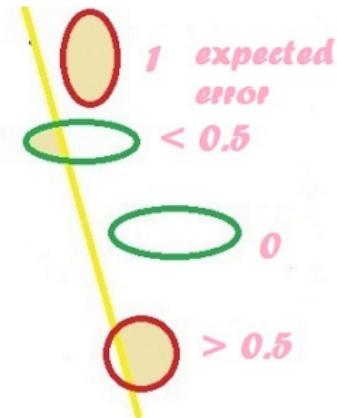
$x \in \mathcal{B}(\bar{x}, \Sigma, \gamma)$  has true label  $y$

$$e_{wc}(\mathcal{B}) = 1, \text{ if } yz \leq \gamma$$

$$z = \frac{w^T \bar{x} + b}{\sqrt{w^T \Sigma w}}$$

## Expected error

$$e_{exp} = \frac{Vol(Misclassified)}{Vol(\mathcal{B})}$$



# Experiments

- Pima dataset [UCI repository]
- $m = 768, d = 8$
- Soft margin SVM
- C is chosen by cross validation
- $\Sigma_i = \Sigma$  and  $\gamma_i = \gamma$  for all  $i$
- Compare with SOCP formulation

# Dependency of test error with radius of uncertainty

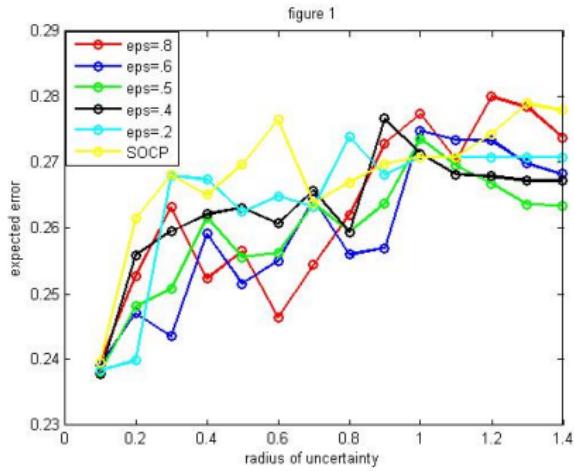


Figure: Expected Error

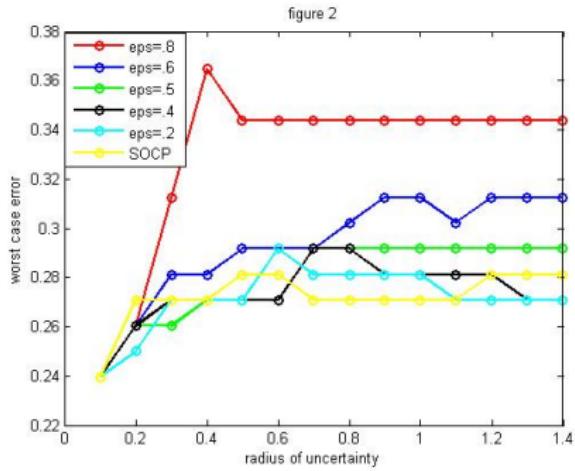


Figure: Worst Case Error

# Dependency of test error with target accuracy

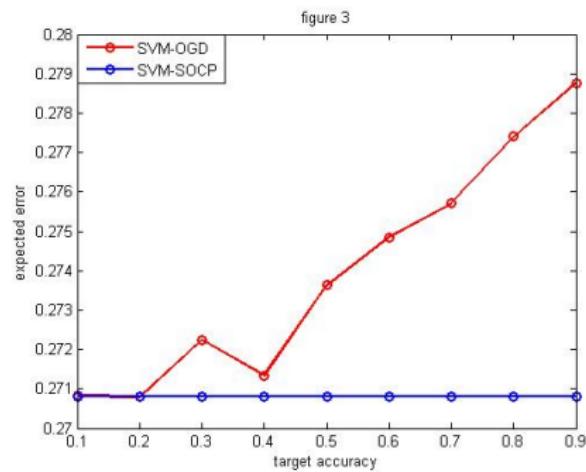


Figure: Expected Error

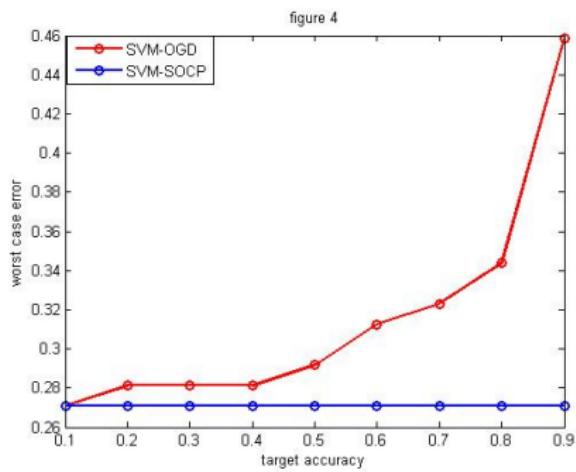


Figure: Worst Case Error

# Conclusion

- Solved robust SVM problem avoiding SOCP formulation
- OGD based approach
- Error related with the choice of target accuracy  $\epsilon$
- Dependency of uncertainty parameter  $\gamma$  with  $\epsilon$

## Future Work

- Assume more structure on the reward functions
- Strong concavity, lipschitz continuity
- Extend to other uncertainty sets ,e.g. multi-dimensional simplex
- Extend to kernel set up of robust SVM

# Thank You