

The cross entropy method for classification

Shie Mannor

McGill University

Dori Peleg
Technion

Reuven Rubinstein
Technion

Setup

- Standard classification problem
 - Given $(x_1, y_1), (x_2, y_2) \dots, (x_n, y_n)$
 - $x_i \in X$ (general input space)
 - $y_i \in \{-1, 1\}$

- We consider classifiers of the form:

$$f(x) = \sum_{i=1}^n y_i \alpha_i k(x_i, x) + b, \alpha_i \geq 0$$

- **Support vectors** = number of samples with $\alpha_i \neq 0$
- k is not necessarily a Mercer kernel – any mapping from $X \times X \rightarrow \mathbb{R}$ is ok

Objective

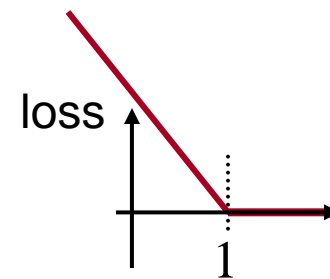
- Solve the following:

minimize Complexity penalty + C Loss term
subject to Constraints on classifier

- We will look at a special case:
 - Complexity = number of support vectors
 - Loss term is hinge loss (ϕ)

of Support vectors

minimize $\|\alpha\|_0 + C \sum_{i=1}^n \phi(y_i f(x_i))$
subject to $0 \leq \alpha \leq C,$



Sparsity

- = number of support vectors
- Translates to generalization error
- Makes life easy on real data
- Hard to achieve
 - Remove redundant points artificially after the run [Jones 01']
 - Iterative approaches
 - Active set methods
 - Change optimization formulation
 - L_1 SVM [Mangasarian 01']
 - One-norm SVMs [Smola et. al 00']
 - L_0 approach ← This work

Reformulation

$$\begin{aligned} &\text{minimize} && \|\alpha\|_0 + C \sum_{i=1}^n \phi(y_i f(x_i)) \\ &\text{subject to} && 0 \leq \alpha \leq C, \end{aligned}$$

is a hard problem (discontinuous!)

Let $\sigma \in \{0, 1\}^n$ denote the “active” support vectors;

$$\Sigma := \text{diag}(\sigma_1, \dots, \sigma_n)$$

$$\begin{aligned} &\text{minimize} && \|\Sigma \hat{\alpha}\|_0 + C \mathbf{1}^\top \xi \\ &\text{subject to} && Y (KY \Sigma \hat{\alpha} + b \mathbf{1}) \geq \mathbf{1} - \xi \end{aligned}$$

$$\hat{\alpha}, \sigma \text{ and } \xi \in \mathbb{R}^n, b \in \mathbb{R}$$

$$\begin{aligned} &\xi \geq 0 \\ &0 \leq \Sigma \hat{\alpha} \leq C \\ &\sigma \in \{0, 1\}^n \end{aligned}$$

$$K_{ij} = K(x_i, x_j)$$

$$Y = \text{diag}(y_1, \dots, y_n)$$

If SVs are known

- Number of SVs = $\|\Sigma\hat{\alpha}\|_0 \approx \sum_{i=1}^n \sigma_i$

~~σ~~ → minimize $\mathbf{1}^\top \xi$
 subject to $Y \left(\tilde{K} \tilde{Y} \tilde{\alpha} + b \mathbf{1} \right) \geq \mathbf{1} - \xi$
 $\xi \geq 0$
 $0 \leq \tilde{\alpha} \leq C,$

$\hat{\alpha} \neq 0$ usually

$\tilde{\alpha}$ = non-zero elements of vector $\Sigma\hat{\alpha}$

$\tilde{K} = K$ without columns indicated by the zeros of σ

$\tilde{Y} = Y$ without columns indicated by the zeros of σ

- This is a “simple” LP, variables = (# of SVs) + n + 1
 constraints = n + n + 2 (# of SVs)

SVs are unknown

- Need to search for SVs
 - Given SVs LP does the trick
- Search for SVs is hard
 - Enumerate: p SVs $\rightarrow \binom{n}{p}$
 - Simulated annealing \rightarrow problem not continuous
 - Adaptive Local search methods \rightarrow a lot of memory
 - Genetic algorithms \rightarrow needs tweaking
- Solution: Use the Cross Entropy Method to search for the SVs



The cross entropy method

- www.cemethod.org
- A **heuristic** optimization method for solving hard problems (TSP, MAX-CLIQUE)
- Generic mechanism based on random guessing
→ update parameters → better guessing
- Convergence guarantees
- Other usage in ML: clustering, policy search, basis function approximation in RL
- Parallelizable
- Settles for sub-optimal solutions

The CE method

- Setup:
 - Solution space \mathcal{Z}
 - Parameterized PDF on $\mathcal{Z} : f(\mathbf{Z};\theta), \theta \in \Theta$
 - Penalty function $S(\mathbf{Z})$
- Objective:
 - Minimize $S(\mathbf{Z})$ over $\mathbf{z} \in \mathcal{Z}$

Method:

1. Generate a sample of random data (trajectories, vectors, etc.) according to current θ
2. Update θ on the basis of the data, in order to produce a “better” sample in the next iteration.

The CE method (cont')

Iteration t :

1. Generation of random data Z_1, Z_2, \dots, Z_m using θ_{t-1}
2. Calculating θ_t :

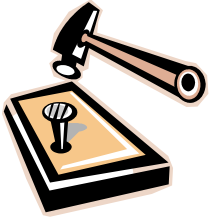
$$\gamma_t = 1 - \rho \text{ percentile of } S(Z_1), S(Z_2), \dots, S(Z_m)$$

Find the next parameter by solving:

$$\theta_t = \arg \max_v \frac{1}{m} \sum_{\ell=1}^m I_{\{S(Z_\ell) \leq \gamma_t\}} \log f(Z_\ell; v)$$

Parameters: ρ , m , and shape of f

- f is NEF \rightarrow closed form solution



CE for classification

- \mathcal{Z} is the space $\{0, 1\}^n$
- Θ : n Bernoulli variables ($\Theta = [0, 1]^n$)
- At each iteration of CE we select a subset of support vectors (σ)
- We solve for that σ

$$\begin{aligned} & \text{minimize} && \mathbf{1}^\top \xi \\ & \text{subject to} && Y \left(\tilde{K} \tilde{Y} \tilde{\alpha} + b \mathbf{1} \right) \geq \mathbf{1} - \xi \\ & && \xi \geq 0 \\ & && 0 \leq \tilde{\alpha} \leq C, \end{aligned}$$

- Score = $S(\sigma) = \mathbf{1}^\top \sigma + C \mathbf{1}^\top \xi$

CE for classification (cont')

- Update of θ between iterations:

$$\hat{\theta}_t(k) = \frac{\sum_{\ell=1}^m I_{\{S(\sigma_\ell^k) \leq \gamma_t\}} I_{\{\sigma_\ell^k = 1\}}}{\sum_{\ell=1}^m I_{\{S(\sigma_\ell^k) \leq \gamma_t\}}}$$

- A smoothed version is used between iterations

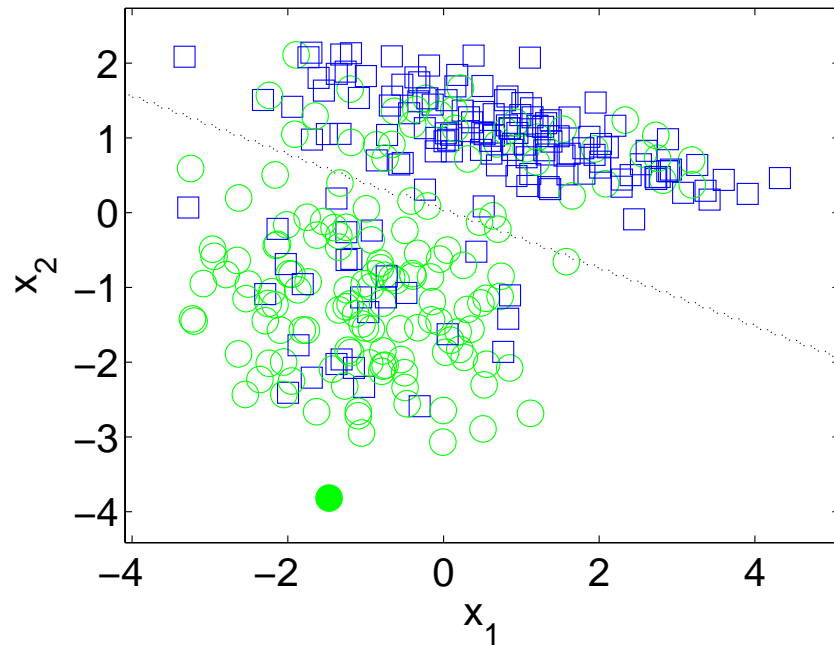
$$\theta_t = \beta \theta_{t-1} + (1 - \beta) \hat{\theta}_t$$

- Tweaks:

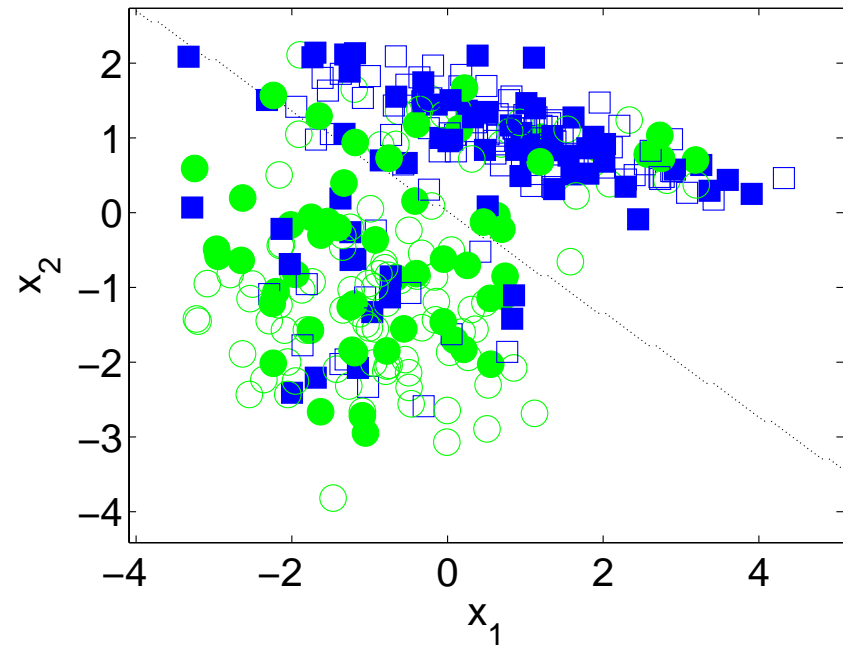
- Small $\tilde{\alpha}$ are truncated to 0
- SVM is solved first and non SVs are excluded
- Solution caching (same σ may be generated twice)
- Stopping condition for CE – no improvement in an iteration

Synthetic data

Two class problem, filled squares/circles = SVs



CE
(C set to minimize test error)



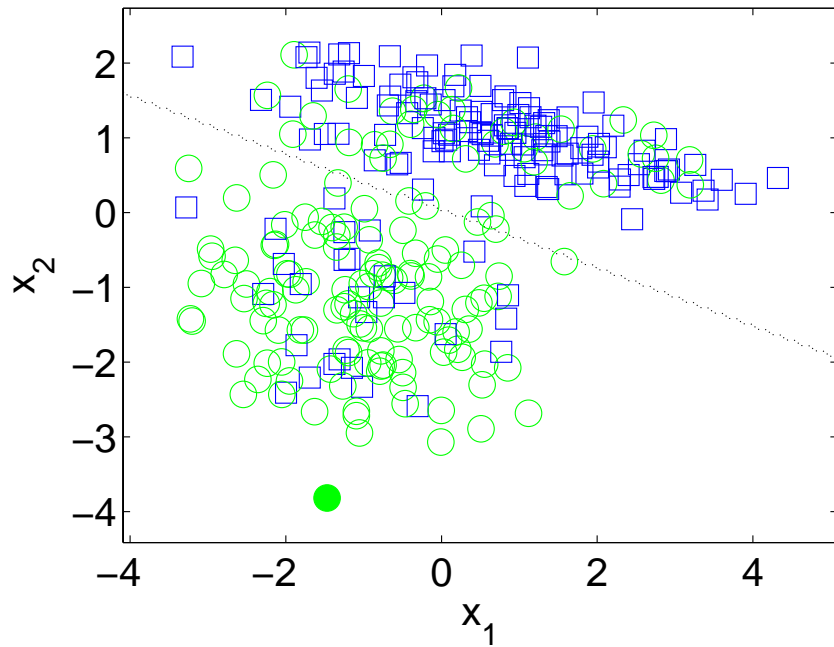
SVM (L_2)
(C set to minimize test error)

One-norm SVM [Smola 00]

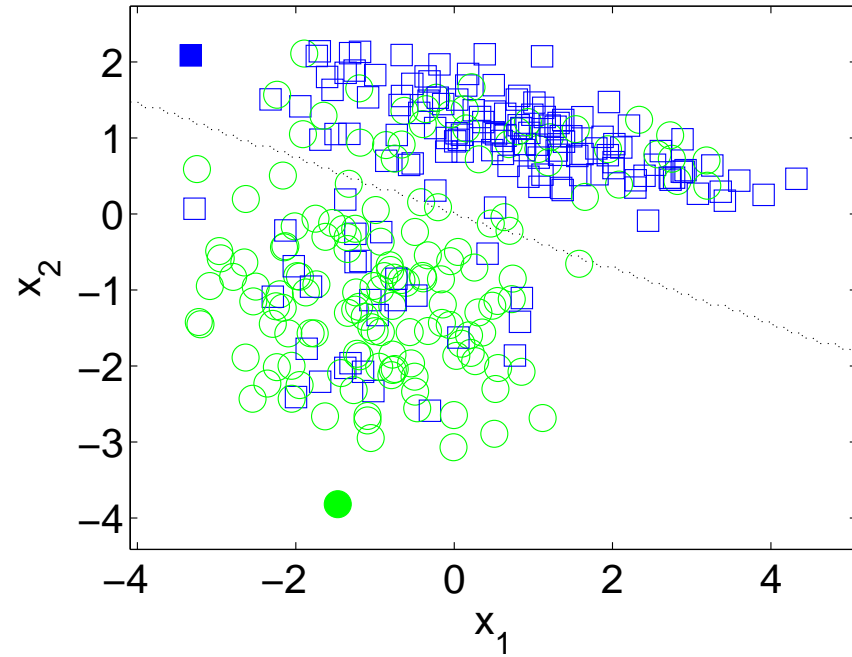
$$\begin{aligned} & \text{minimize} && \mathbf{1}^\top \alpha + C \mathbf{1}^\top \xi \\ & \text{subject to} && Y^\top (KY\alpha + b\mathbf{1}) \geq \mathbf{1} - \xi \\ & && \xi \geq 0 \\ & && 0 \leq \alpha \leq C, \end{aligned}$$

- Manages to get pretty sparse solutions, though less than CE
- An upper bound on the 0-norm
- Easily solved by an LP
- A good starting point for CE

Synthetic data



CE
(1 SV)



SVM (L_1)
(2 SVs)

Data sets:

Name	# Features	# of patterns
Ionosphere	34	351
Pima	8	768
Wdbc	30	569
Bupa	6	345
Sonar	60	208

Real data (linear kernel)

Validation Error

% of support vectors

Data set	SVM	CE	One norm SVM
Ionosphere	14.7 ± 2.0 (36.4 ± 8.9)	14.6 ± 1.8 (7.7 ± 2.6)	13.0 ± 1.8 (15.1 ± 2.5)
Pima	24.3 ± 1.4 (51.2 ± 6.2)	24.8 ± 1.5 (3.9 ± 1.2)	24.6 ± 1.1 (4.9 ± 0.5)
Wdbc	5.7 ± 1.2 (10.1 ± 2.5)	5.9 ± 0.8 (3.7 ± 1.4)	5.9 ± 1.4 (4.8 ± 1.1)
Bupa	32.6 ± 2.1 (71.9 ± 3.8)	33.4 ± 2.8 (3.1 ± 0.6)	32.5 ± 1.7 (4.0 ± 0.0)
Sonar	25.9 ± 3.7 (53.7 ± 7.9)	25.5 ± 4.7 (10.3 ± 1.9)	25.5 ± 4.7 (14.7 ± 2.4)

Real data (polynomial kernel)

Validation Error

% of support vectors

Data set	SVM	CE	One norm SVM
Ionosphere	15.3 ± 2.7 (36.1 ± 3.7)	12.5 ± 1.3 (7.1 ± 1.1)	13.7 ± 2.6 (20.5 ± 8.4)
Pima	33.2 ± 1.5 (48.8 ± 5.2)	30.2 ± 2.4 (11.2 ± 6.6)	30.6 ± 1.8 (29.5 ± 4.6)
Wdbc	6.0 ± 2.1 (21.9 ± 2.7)	5.6 ± 1.3 (2.5 ± 0.7)	8.5 ± 2.8 (15.1 ± 3.2)
Bupa	33.7 ± 5.2 (58.0 ± 6.0)	37.9 ± 4.4 (14.4 ± 9.9)	36.3 ± 2.2 (33.9 ± 3.5)
Sonar	15.9 ± 4.7 (70.3 ± 1.7)	23.3 ± 5.3 (6.9 ± 1.6)	20.3 ± 7.0 (51.1 ± 6.8)

Real data (RBF kernel)

Validation Error

% of support vectors

Data set	SVM	CE	One norm SVM
Ionosphere	9.8 ± 2.3 (76.3 ± 2.2)	6.6 ± 2.3 (14.1 ± 2.6)	6.2 ± 1.5 (19.3 ± 3.1)
Pima	27.5 ± 1.7 (67.9 ± 5.1)	25.4 ± 3.3 (8.5 ± 1.7)	25.2 ± 3.0 (12.9 ± 4.9)
Wdbc	7.5 ± 0.8 (42.4 ± 3.4)	4.7 ± 1.4 (9.7 ± 1.2)	4.6 ± 1.5 (14.4 ± 1.5)
Bupa	34.4 ± 3.0 (93.4 ± 1.6)	36.9 ± 4.6 (10.4 ± 4.3)	36.9 ± 3.9 (28.3 ± 25.5)
Sonar	46.7 ± 6.3 (100.0 ± 0.0)	24.5 ± 3.7 (22.5 ± 2.5)	24.3 ± 3.5 (41.7 ± 6.2)

Experiments - conclusions

- Standard SVMs are **not** (usually) sparse
 - Especially for non-linear kernels
- CE provides comparable accuracy to SVM
 - Statistically no method is better
- CE has much fewer support vectors than “vanilla” SVM
- CE has fewer SVs than one-norm SVMs
 - Linear kernels – 29% less
 - Non-linear kernels (70% for polynomial and 40% for RBF)
- Conclusions holds for different data sets and different kernels

Wrap-up

- SV selection problem: NP-hard is not that bad
- Kernel does not have to be a Mercer kernel
- Computation time is potentially lengthy
 - Can be parallelized
 - Excellent available LP solvers
 - Starting point is important
- Zero-norm is a good complexity measure
 - Feature selection
 - Regression
- Another hammer in the toolbox
- Code available from <http://www.ee.technion.ac.il/people/dorip/Code.html>

The cross entropy method for classification

Thank you !

Shie Mannor
McGill University

Dori Peleg
Technion

Reuven Rubinstein
Technion