Advanced Machine Learning & Perception

Instructor: Tony Jebara

Topic 3

- Maximum Margin
- Empirical Risk Minimization
- VC Dimension & Structural Risk Minimization
- Support Vector Machines
- Reduced Working Sets
- Sequential Minimal Optimization (SMO)
- Ellipsoidal Kernel Machines
- Maximum Relative Margin
- Other SVM Extensions...

Generative vs. Discriminative

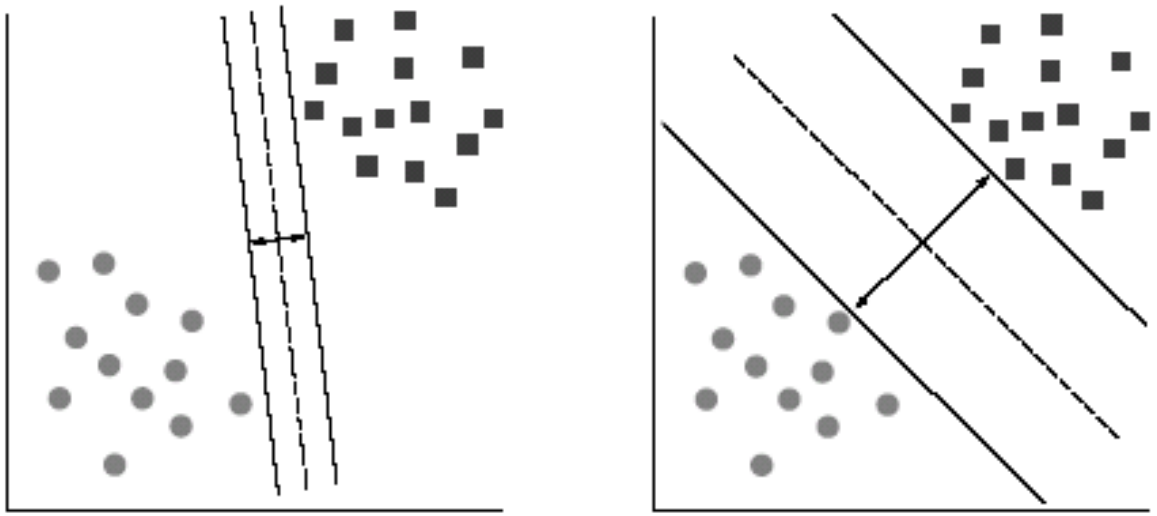
- Generative approach to classification:
  get p(x,y) then p(y|x) then via y = f(x) = argmax_y p(y|x)
- For example, learn Gaussians by fitting each with maximum likelihood and use as classifiers by posterior:

$$\left\{x_1^+,\ldots,x_N^+\right\} \rightarrow p\left(x \mid y=+1\right)$$
$$\left\{x_1^-,\ldots,x_M^-\right\} \rightarrow p\left(x \mid y=-1\right)$$
$$p\left(y=1 \mid x\right) = \frac{p\left(x \mid y=+1\right)p\left(y=+1\right)}{\sum_y p\left(x \mid y\right)p\left(y\right)}$$

- Vapnik: it is inefficient to learn a probability of everything if we only need a binary classifier y=f(x)
- Also, maximum likelihood tries to capture everything about the data, not the classification decision.
- Try just learning best possible classifier discriminatively.

Large Margins

- 1995: SVMs & VC Theory popularize large margin learning



- Other margin methods: Boosting, Max Margin Markov Nets, Max Margin Matrix Factorization, …
- Other margin theories: Boosting, PAC-Bayes, Rademacher
- Are large margins right? Can SDPs help?
- Let's re-visit SRM, VC, & SVMs…

Empirical vs. Structural Risk

- Example: want a linear classifier to separate two classes

$$f\left(x;\theta\right) = sign\left(w^T x + b\right) \quad where\ \theta = \left\{w, b\right\}$$

- Choose a loss function:

$$L\left(y, x, \theta\right) = step\left(-yf\left(x;\theta\right)\right)$$

- Empirical Risk Minimization fits only to training data:

$$R_{emp}\left(\theta\right) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, f\left(x_i;\theta\right)\right) \quad \in \left[0,1\right]$$

- Empirical $R_{emp}\left(\theta\right)$ *approximates* the true risk (expected error)

$$R\left(\theta\right) = E_P\left\{L\left(x, y, \theta\right)\right\} = \int_{X\times Y} P\left(x, y\right) L\left(x, y, \theta\right) dx\, dy \in \left[0,1\right]$$

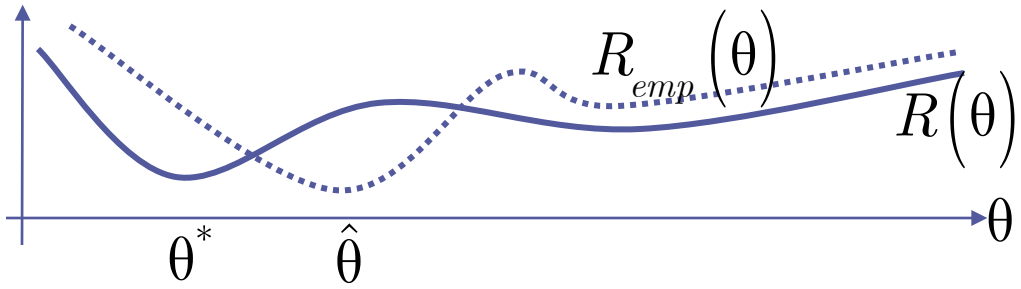
- We don't know the true P(x,y)

$$\arg\min_\theta R_{emp}\left(\theta\right) \neq \arg\min_\theta R\left(\theta\right)$$

- Instead SVMs perform Structural Risk Minimization

Empirical vs. Structural Risk

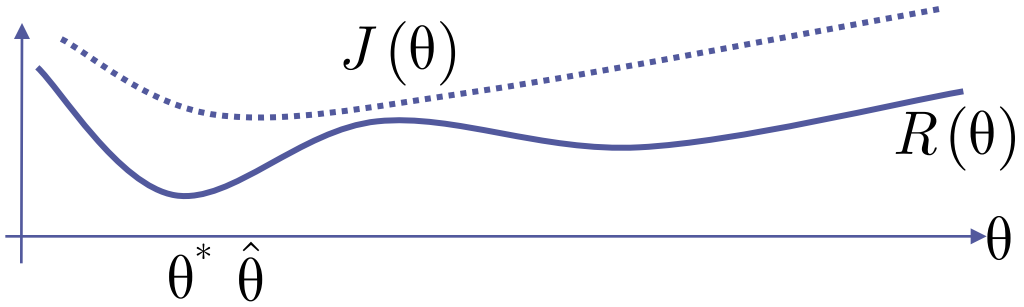
- ERM: inconsistent
  Not guaranteed.
  May do better on training than on test!



$$R(\hat{\theta}) \geq R_{emp}(\hat{\theta})$$

- SRM: add a prior or regularizer to $R_{emp}(\theta)$
- Define capacity or confidence = $C(\theta)$ which favors simpler θ

$$J(\theta) = R_{emp}(\theta) + C(\theta)$$



- If, $R(\theta) \leq J(\theta)$ we have bound $J(\theta)$ is a guaranteed risk
- SRM: minimize J, guarantee future error rate is $\leq \min_\theta J(\theta)$

Structural Risk Minimization & VC

- How to bound risk? Learning theory…
- Theorem (Vapnik): with probability 1-η the following holds:

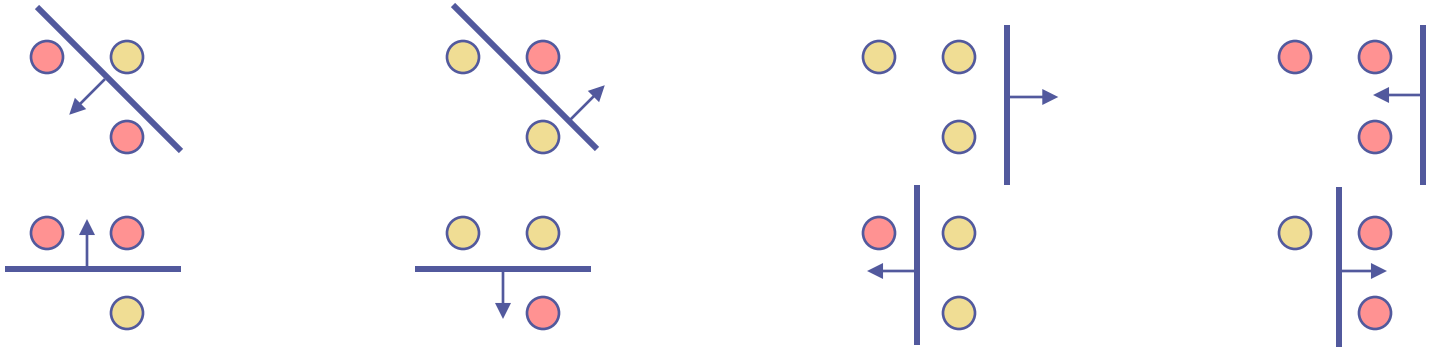
$$R(\theta) \le J(\theta) = R_{emp}(\theta) + \Phi\left(\frac{h}{N}, \frac{\ln \eta}{N}\right)$$

$$R(\theta) \le J(\theta) = R_{emp}(\theta) + \frac{2h\log\left(\frac{2eN}{h}\right) + 2\log\left(\frac{4}{\eta}\right)}{N}\left(1 + \sqrt{1 + \frac{NR_{emp}(\theta)}{h\log\left(\frac{2eN}{h}\right) + \log\left(\frac{4}{\eta}\right)}}\right)$$

N = number of data points

h = Vapnik-Chervonenkis (VC) dimension

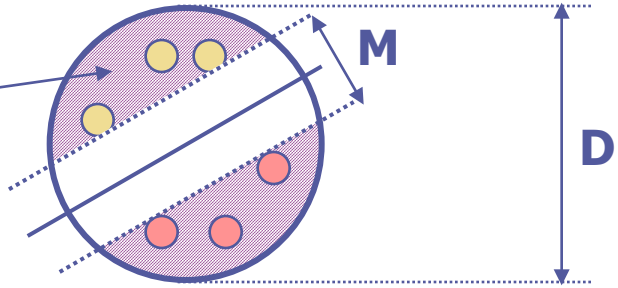
- VC dimension of linear classifiers in d-dimensions is h=d+1



VC of Gap-Tolerant Classifiers

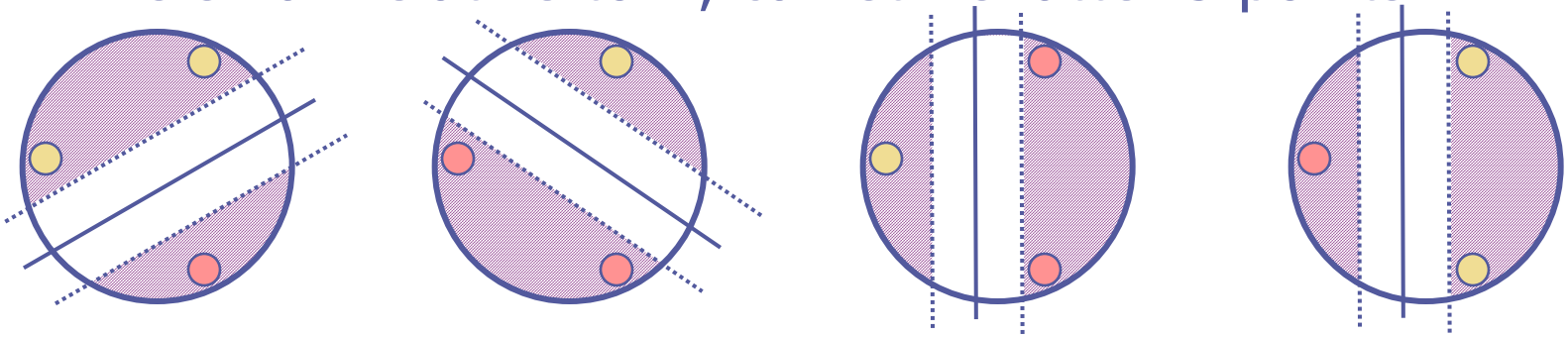
- Arbitrary linear classifiers are too flexible as a function class
- Can improve estimate of VC dimension if we restrict them
- Constrain linear classifiers to data living inside a sphere
- Gap-Tolerant classifiers: a linear classifier whose activity is constrained to a sphere & outside a margin

Only count errors in shaded region Elsewhere have $L(x,y,\theta)=0$



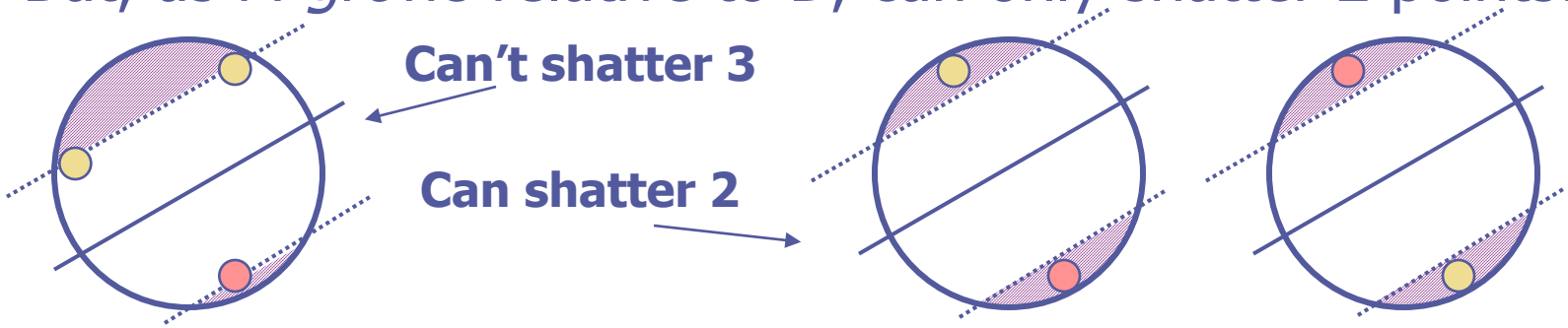
M=margin
D=diameter
d=dimensionality

- If M is small relative to D, can still shatter 3 points:



VC of Gap-Tolerant Classifiers

- But, as M *grows* relative to D, can only shatter 2 points!



- For hyperplanes, as M grows vs. D, shatter fewer points!
- VC dimension h goes down if gap-tolerant classifier has larger margin, general formula is: $h \leq \min\left\{ceil\left[\frac{D^2}{M^2}\right], d\right\} + 1$
- Before, just had h=d+1. Now we have a smaller h
- If data is anywhere, D is infinite and back to h=d+1
- Typically real data is bounded (by sphere), D is fixed
- Maximizing M reduces h, improves guaranteed risk J(θ)
- There is no way to modify R with θ

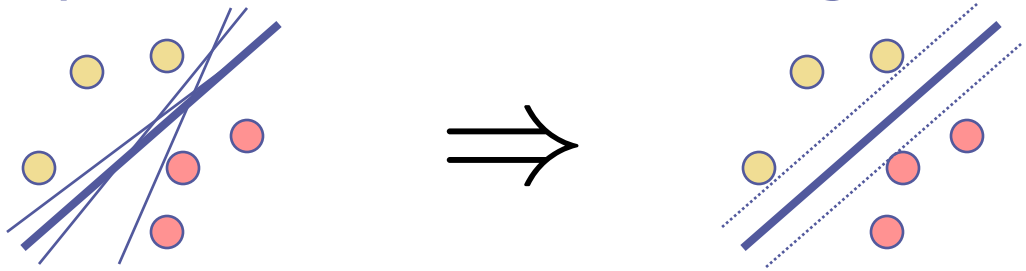
Support Vector Machines

- Support vector machines are (in the simplest case) linear classifiers that do structural risk minimization (SRM)
- Directly maximize margin to reduce guaranteed risk J(θ)
- Assume first the 2-class data is linearly separable:

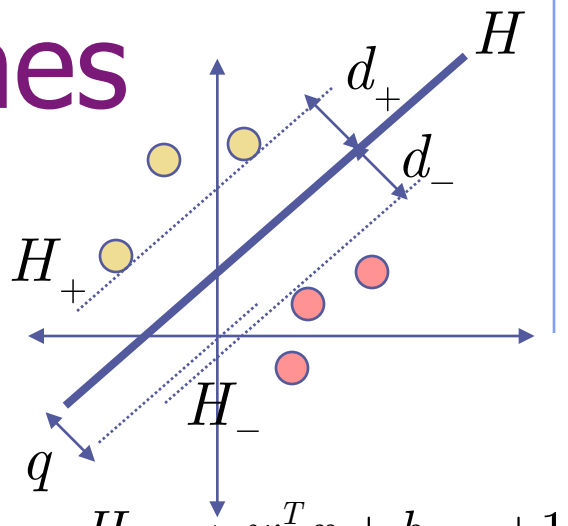
$$have \ \left\{\left(x_1, y_1\right), \ldots, \left(x_N, y_N\right)\right\} \ \ where \ x_i \in \mathbb{R}^D \ and \ y_i \in \left\{-1, 1\right\}$$

$$f\left(x;\theta\right) = sign\left(w^T x + b\right)$$

- Decision boundary or hyperplane given by $w^T x + b = 0$
- Note: can scale w & b while keeping same boundary
- Many solutions exist which have empirical error $R_{emp}(\theta)=0$
- Want unique widest one → max margin.



Support Vector Machines



$H \rightarrow w^T x + b = +1$
$H_- \rightarrow w^T x + b = -1$

- The constraints on the SVM for $R_{emp}(\theta)=0$ are:

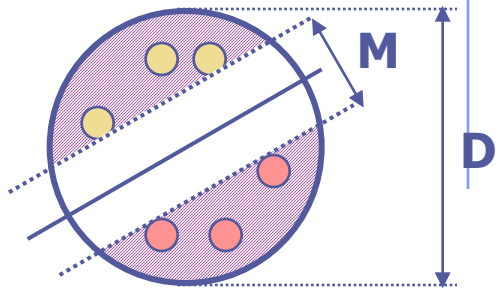
$$w^T x_i + b \geq +1 \quad \forall y_i = +1$$
$$w^T x_i + b \leq -1 \quad \forall y_i = -1$$

- Or more simply: $y_i\left(w^T x_i + b\right) - 1 \geq 0$
- The margin of the SVM is:

$$m = d_+ + d_-$$

- Distance to origin: $H \rightarrow q = \frac{|b|}{\|w\|} \quad H_+ \rightarrow q_+ = \frac{|b-1|}{\|w\|} \quad H_- \rightarrow q_- = \frac{|-1-b|}{\|w\|}$
- Therefore: $d_+ = d_- = \frac{1}{\|w\|}$ and margin $m = \frac{2}{\|w\|}$
- Want to max margin, or equivalently minimize: $\|w\| \ or \ \|w\|^2$
- SVM Problem: $\min \frac{1}{2}\|w\|^2 \quad subject\ to \quad y_i\left(w^T x_i + b\right) - 1 \geq 0$
- This is the primal quadratic program
- Dual, nonseparable & nonlinear case are straightforward.

Support Vector Machines



- Dual, kernelized, slackened SVM QP:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j k\left(x_i, x_j\right)$$
$$subject\ to\ \sum_i \alpha_i y_i = 0 \ \ \& \ \ \alpha_i \in \left[0, C\right]$$

- The margin is $M = \frac{2}{\sqrt{w^T w}}$ $where$ $w = \sum_t \alpha_t y_t k\left(x_t, .\right)$

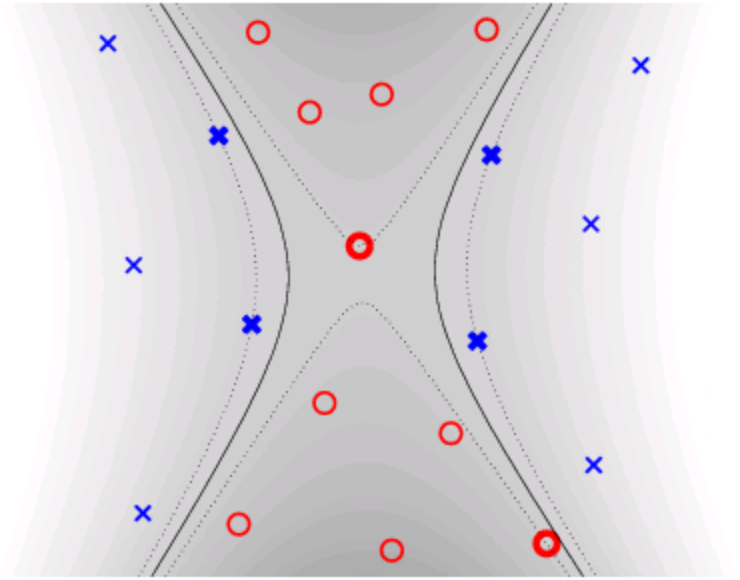
- Find bounding sphere on data to get radius using QP:

$$R^2 = \max_{\beta} \sum_i \beta_i k\left(x_i, x_i\right) - \sum_{i,j} \beta_i \beta_j k\left(x_i, x_j\right)$$
$$subject\ to\ \sum_i \beta_i = 1 \ \& \ \ \beta_i \geq 0$$

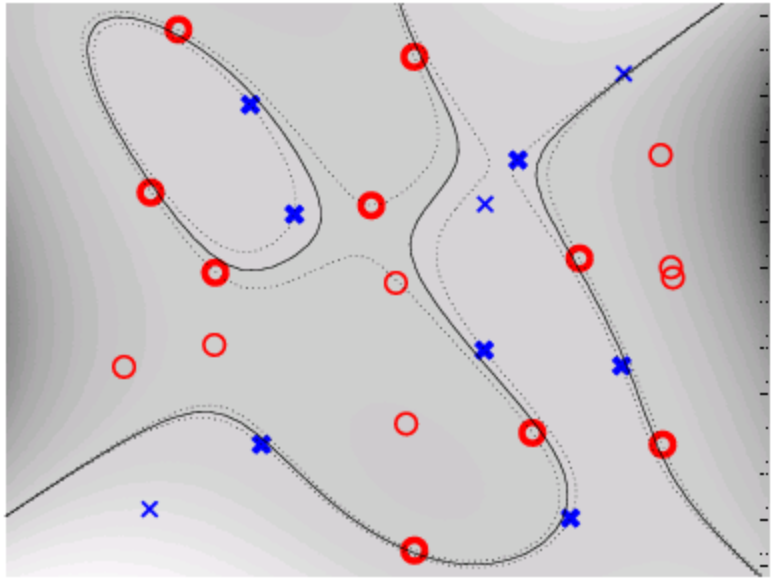
- The VC dimension is then: $h \leq \min\left\{ceil\left[\frac{4R^2}{M^2}\right], d\right\} + 1$

Support Vector Machines

- Can thus design all sorts of strange kernels on non vector objects as long as all computations involve kernel function and not explicit mapping and solve the SVM in dual space, for example using quadratic programming (how big is this quadratic program?)



(b) Polynomial Kernel SVM



(c) RBF Kernel SVM

Reduced Working Set Algorithms

- Since QP needs NxN Gram matrix, this is impossible to store or use for N=50,000 training points
- Idea (Vapnik): only support vectors with non-zero Lagrange multipliers need to be used and stored
- So, split data into chunks that fit into memory and solve QP only over their Lagrange multipliers, freezing others
- Definitions of the points and note KKT conditions:

Non-critical points are safely outside margin:

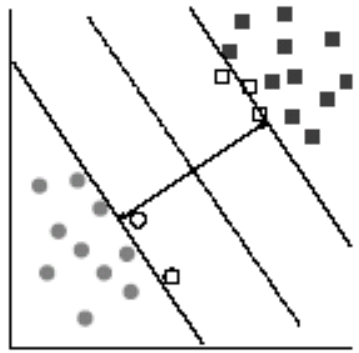
$$\lambda_i = 0 \quad \Leftrightarrow \quad f\left(x_i\right) y_i > 1$$

Support vectors on the margin:

$$\lambda_i \in \left(0, C\right) \quad \Leftrightarrow \quad f\left(x_i\right) y_i = 1$$

Violaters inside margin or on wrong side:

$$\lambda_i = C \quad \Leftrightarrow \quad f\left(x_i\right) y_i < 1$$



Reduced Working Set Algorithms

- For example Osuna splits training set into two sets:

  B: working set N: inactive set

  1) Init with random choice of points in B
  2) Solve SVM on B only
  3) If have violater j in N where $f\left(x_j\right)y_j < 1$ replace with non-critical i in B where $\lambda_i = 0$
  4) Go to 2

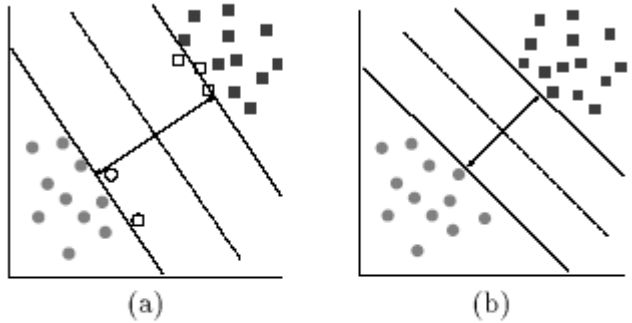


Figure 2: (a) A sub-optimal solution where the non-filled points have $\lambda = 0$ but are violating optimality conditions by being inside the ± 1 area. (b) The decision surface is redefined. Since no points with $\lambda = 0$ are inside the ± 1 area, the solution is optimal. Notice that the size of the margin has decreased, and the *shape* of the decision surface has changed.

- Should to converge in finite number of steps
- Since we only add in new points which do not satisfy KKT and dual maximization problem has to increase.

Sequential Minimal Optimization

- What is the smallest working set we could consider?

$$\max \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi\left(x_i\right)^T \phi\left(x_j\right) \quad s.t. \ \sum_i \alpha_i y_i = 0 \ \& \ \alpha_i \in \left[0, C\right]$$

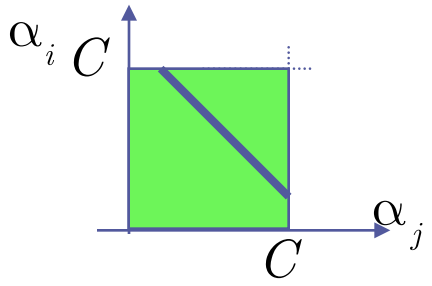
Sequential Minimal Optimization

- What is the smallest working set we could consider?

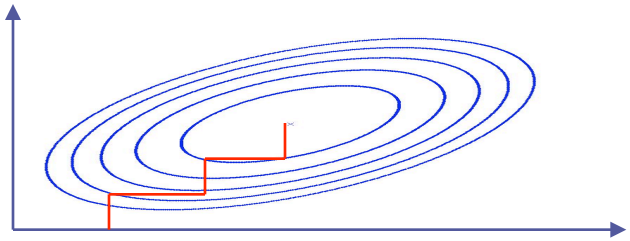
$$\max \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \quad s.t. \ \sum_i \alpha_i y_i = 0 \ \& \ \alpha_i \in [0, C]$$

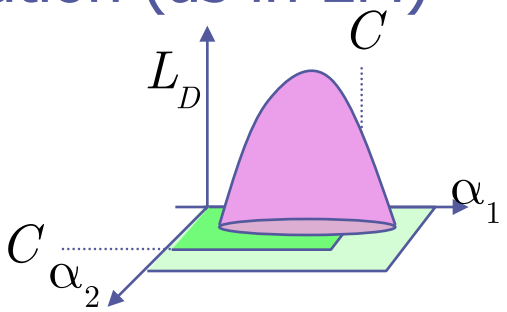
- Can update just 2 Lagrange multipliers at a time (Platt) since we have the sum to 0 constraint

$$\sum_i \alpha_i y_i = 0$$



- Like a constrained axis-parallel optimization (as in EM)





- But, since SVM is convex program, updating subset of variables while others fixed will also converge globally

Sequential Minimal Optimization

- SMO is fast since update for 2 alphas is just quadratic eqn
- Rewrite SVM dual problem as a function of just 2 alphas

$$L_D \propto \alpha_i + \alpha_j - \tfrac{1}{2}\left(K_{ii}\alpha_i^2 + 2K_{ij}\alpha_i\alpha_j + K_{jj}\alpha_j^2\right) - h_i\alpha_i - h_j\alpha_j$$

$$subject\ to:\ y_i\alpha_i + y_j\alpha_j + \sum_{t\neq i,j} y_t\alpha_t = 0 \quad and\ \alpha_i, \alpha_j \in \left[0, C\right]$$

- Can now write a simple update rule, noting constraints (also, avoid whole Gram matrix, only compute needed K's)

$$S = y_i y_j$$

$$L = \max\left(0, \alpha_j + S\alpha_i - \tfrac{1}{2}\left(S+1\right)C\right) \qquad E1 = \sum_t \alpha_t y_t k\left(x_i, x_t\right) + b - y_i$$

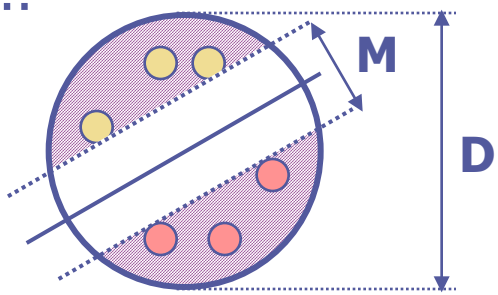
$$H = \min\left(C, \alpha_j + S\alpha_i - \tfrac{1}{2}\left(S-1\right)C\right) \qquad E2 = \sum_t \alpha_t y_t k\left(x_j, x_t\right) + b - y_j$$

$$\alpha_j^{NEW} = \alpha_j + \frac{y_j\left(E1 - E2\right)}{k\left(x_i, x_i\right) + k\left(x_j, x_j\right) - 2k\left(x_i, x_j\right)} \quad clipped\ inside\left[L, H\right]$$

$$\alpha_i^{NEW} = \alpha_i + S\left(\alpha_j - \alpha_j^{NEW}\right)$$

VC of Spherical Gap-Tolerance

- Led to large margin SVMs…



- The margin is $M = \frac{2}{\sqrt{w^T w}}$ $where$ $w = \sum_t \alpha_t y_t k\left(x_t, .\right)$
- Find bounding sphere on data to get radius using QP:

$$R^2 = \max_\beta \sum_i \beta_i k\left(x_i, x_i\right) - \sum_{i,j} \beta_i \beta_j k\left(x_i, x_j\right)$$

$$subject\ to\ \sum_i \beta_i = 1\ \&\ \beta_i \geq 0$$

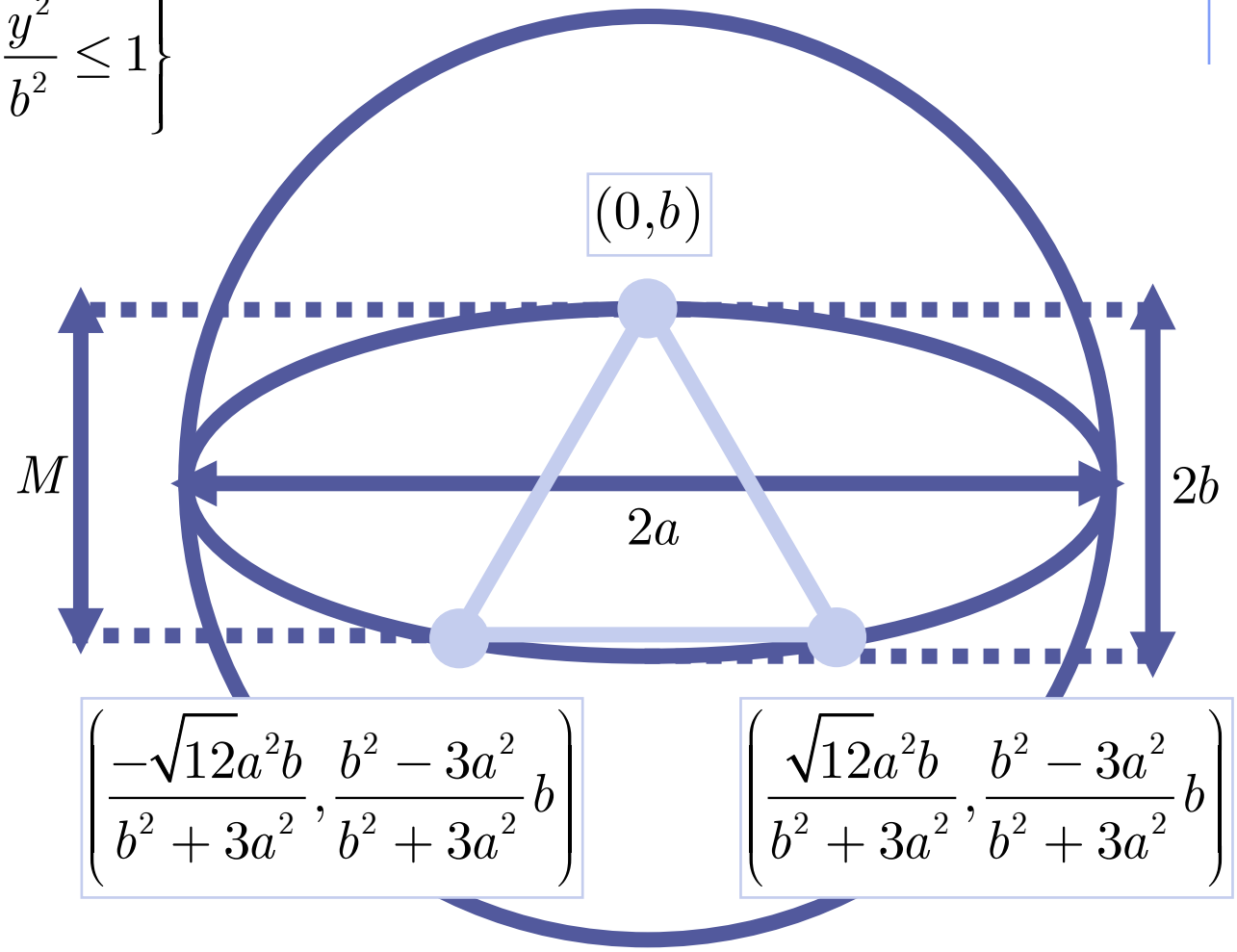
- The VC dimension is then: $h \leq \min\left\{ceil\left[\frac{4R^2}{M^2}\right], d\right\} + 1$

VC of Ellipsoidal Gap-Tolerance

- Consider 2d ellipse:

$$\varepsilon = \left\{ (x, y) : \frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1 \right\}$$

- Largest margin configuration for 3 points is a centered equilateral triangle

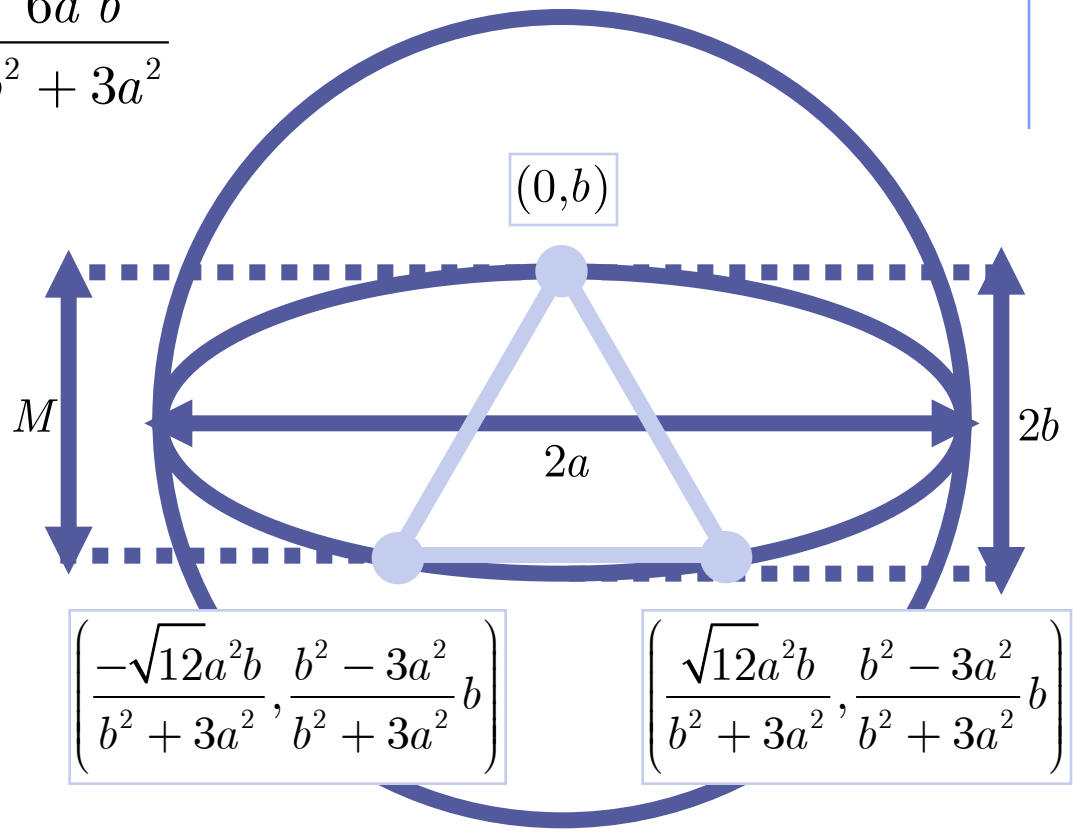


VC of Ellipsoidal Gap-Tolerance

- Ellipse max margin is $\frac{6a^2b}{b^2+3a^2}$
- Sphere rounds up b=a
- Its max margin is $\frac{3}{2}a$
- So, if M is

$$\frac{6a^2b}{b^2+3a^2} < M < \frac{3}{2}a$$

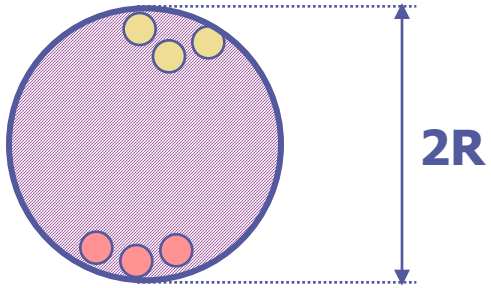
Sphere says VC=3
But, Ellipse says VC=2

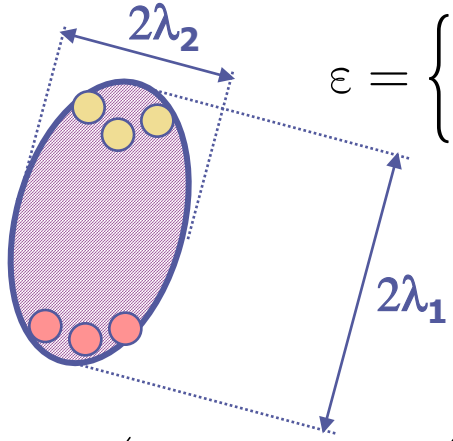


- Ellipsoids always give lower VC than spheres

Minimum Volume Ellipsoid

- Extend QP from bounding sphere to ellipsoid.





$$\varepsilon = \left\{ x : \left(x-\mu\right)^T \Sigma^{-1} \left(x-\mu\right) \le 1 \right\}$$

with volume related to $|\Sigma|$

- Change variables: $A = \Sigma^{-1/2} \;\; and \;\; b = \Sigma^{-1/2}\mu$
- Get SDP: $\min_{A,b} -\ln|A| \;\; s.t. \left(Ax_i - b\right)^T \left(Ax_i - b\right) \le 1 \;\; \forall i \;\; \& \;\; A \succeq 0$
- Slacken SDP for outliers:

$$\min_{A,b,\tau} -\ln|A| + E\sum_i \tau_i \;\; s.t. \left(Ax_i - b\right)^T \left(Ax_i - b\right) \le 1 + \tau_i \;\; \& \;\; A \succeq 0$$

- Enforce quadratic SDP constraints via:

$$\begin{bmatrix} I & \left(Ax_i + b\right) \\ \left(Ax_i - b\right)^T & 1 + \tau_i \end{bmatrix} \succeq 0$$

Minimum Volume Ellipsoid SDP

- LP < QP < QCQP < SDP < Convex Programming
- Matlab, Matlab, Mosek, Yalmip

- LP $\min_{\vec{x}} \vec{b}^T \vec{x} \;\; s.t. \; \vec{c}_i^T \vec{x} \geq \alpha_i \;\; \forall i$

- QP $\min_{\vec{x}} \frac{1}{2} \vec{x}^T H \vec{x} + \vec{b}^T \vec{x} \;\; s.t. \; \vec{c}_i^T \vec{x} \geq \alpha_i \;\; \forall i$

- QCQP $\min_{\vec{x}} \frac{1}{2} \vec{x}^T H \vec{x} + \vec{b}^T \vec{x} \;\; s.t. \; \vec{c}_i^T \vec{x} \geq \alpha_i \;\; \forall i, \vec{x}^T \vec{x} \leq \eta$

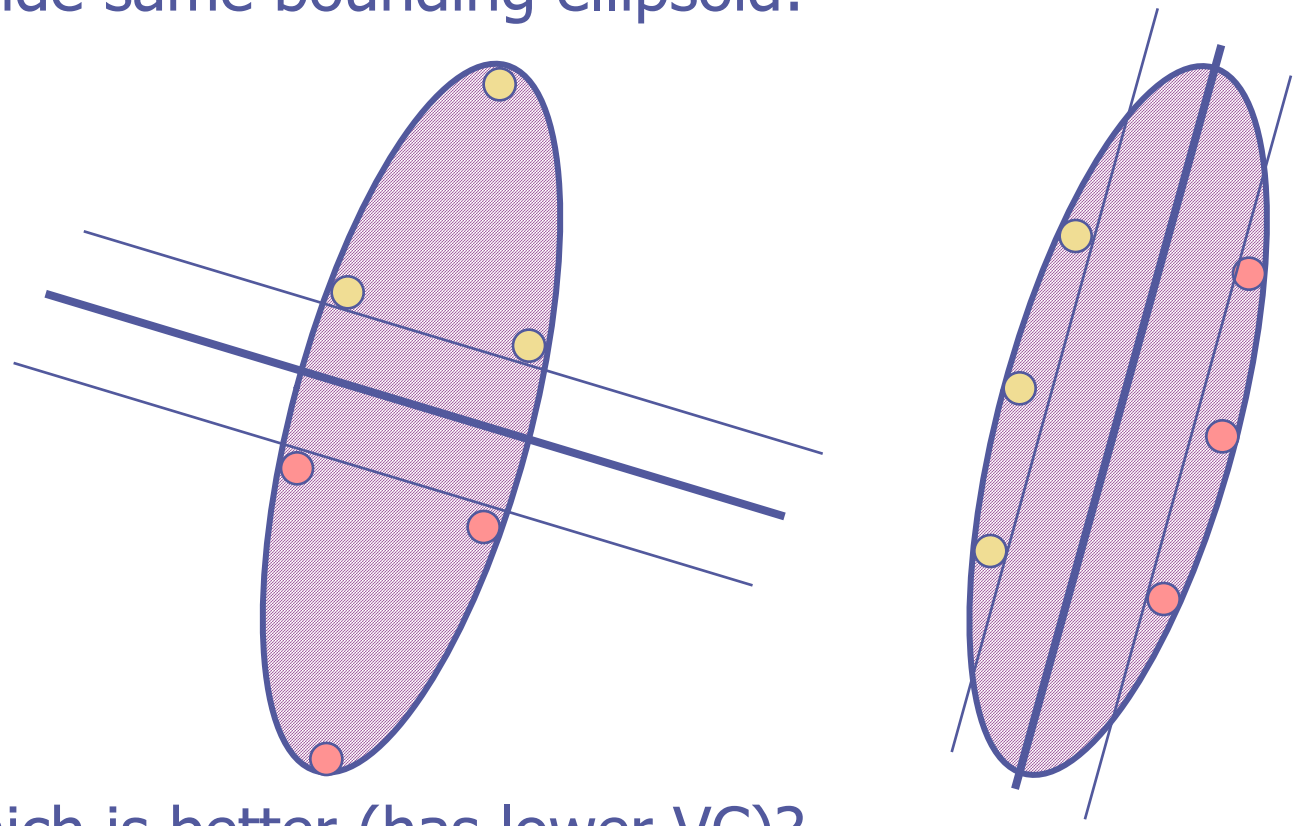
- SDP $\min_K tr\left(B^T K\right) \;\; s.t. \; tr\left(C_i^T K\right) \geq \alpha_i \;\; \forall i, K \succeq 0$

- SDP det $\min_K -\log\left|K\right| \;\; s.t. \; tr\left(C_i^T K\right) \geq \alpha_i \;\; \forall i, K \succeq 0$

- We use the SDP determinant maximization in YALMIP

Ellipsoidal Machines

- How does shape affect classification?
- Consider two linear classifiers of margin M
- Inside same bounding ellipsoid:



- Which is better (has lower VC)?

Ellipsoidal Machines

- Affine transform ellipsoid space to sphere $\hat{x}_i = \Sigma^{-\frac{1}{2}}\left(x_i - \mu\right)$
  Then solve standard SVM in transformed space.
- Or, implicitly solve SVM with new margin metric:

$$\min_{w,\xi} \frac{1}{2} w^T \Sigma w + C\sum_i \xi_i \quad subject\,to \quad y_i\left(w^T x_i + b\right) \geq 1 - \xi_i \quad and \; \xi_i \geq 0$$

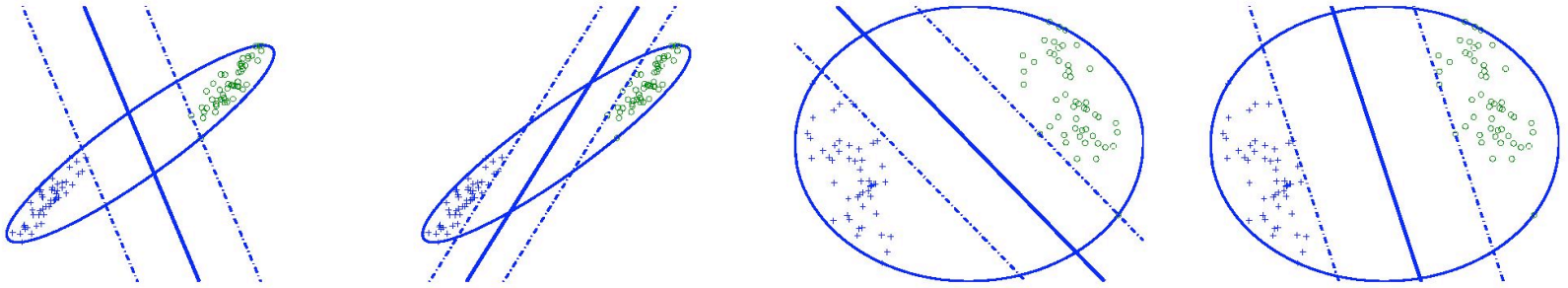


Figure 2: (a) Classical SVM solution on the data, (b) Ellipsoidal Machine solution on the data, (c) Classical SVM solution from the first plot after making the data spherical and (d) Ellipsoidal Machine solution from the second plot after making the data spherical.

- The linear boundary tilts, ***margins are not enough!***
- Linear SVMs not affine invariant (just rotation & translation)

Experiments: SVM vs EVM

Setup 1: UCI Data, Ten Folds per Dataset
Split into 80% Train for (w,b) and (Σ,μ)
10% Cross-validate over C & E
10% Test accuracy

Dataset	Classical	Ellipsoidal
Heart	0.819 ± 0.013	$\mathbf{0.831 \pm 0.015}$
Pima	0.763 ± 0.001	$\mathbf{0.764 \pm 0.001}$
Ion	0.803 ± 0.003	$\mathbf{0.835 \pm 0.002}$
Pen Digit	0.997 ± 0.000	$\mathbf{0.999 \pm 0.000}$
Iris	$\mathbf{0.965 \pm 0.002}$	$\mathbf{0.965 \pm 0.002}$
Bupa	0.655 ± 0.008	$\mathbf{0.658 \pm 0.006}$
Segmentation	0.798 ± 0.005	$\mathbf{0.825 \pm 0.005}$

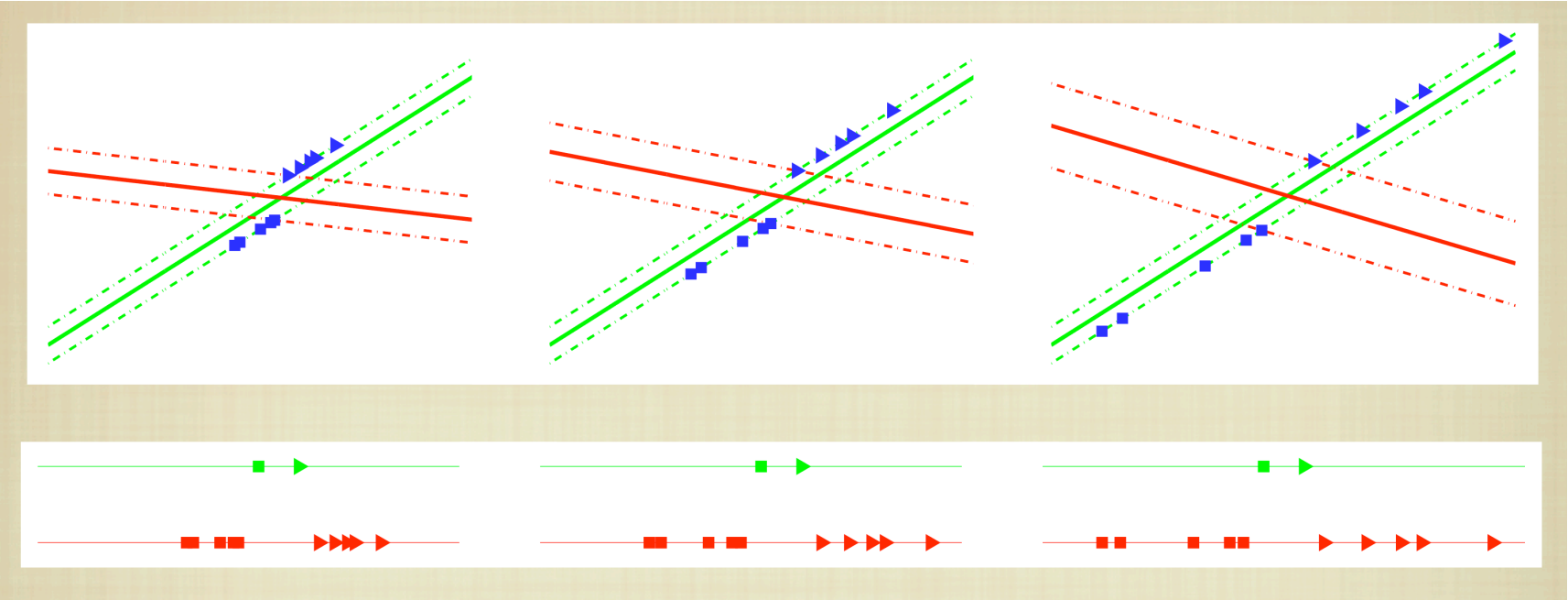
Experiments: SVM vs EVM

Setup 2: UCI Data, Ten Folds per Dataset
Train (Σ,μ) for various values of E on *all* x's.
Split into 80% Train for (w,b)
10% Cross-validate over C & E
10% Test accuracy

Dataset	Classical	Ellipsoidal
Sonar	0.752 ± 0.005	$\mathbf{0.757 \pm 0.009}$
Segmentation	0.804 ± 0.003	$\mathbf{0.838 \pm 0.005}$
Pen Digit	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
Bupa	0.676 ± 0.004	$\mathbf{0.685 \pm 0.005}$
Iris	0.946 ± 0.003	$\mathbf{0.966 \pm 0.002}$
Ionosphere	0.854 ± 0.002	$\mathbf{0.857 \pm 0.003}$
Heart	$\mathbf{0.859 \pm 0.005}$	0.855 ± 0.003
Pima	0.761 ± 0.001	$\mathbf{0.766 \pm 0.001}$

Maximum Relative Margin

- Details in Shivaswamy and Jebara in NIPS 2008



- Red is maximum margin, Green is max relative margin
- Top is a two d classification problem
- Bottom is projection of data on solution $w^T x+b$
- SVM solution changes as axes get scaled, has large spread

Maximum Relative Margin

- Fast trick to solve the *same* problem as on previous slides:
  Bound the spread of the SVM!
- Recall original SVM primal problem (with slack):

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \quad subject\ to \quad y_i\left(w^T x_i + b\right) \geq 1 - \xi_i$$

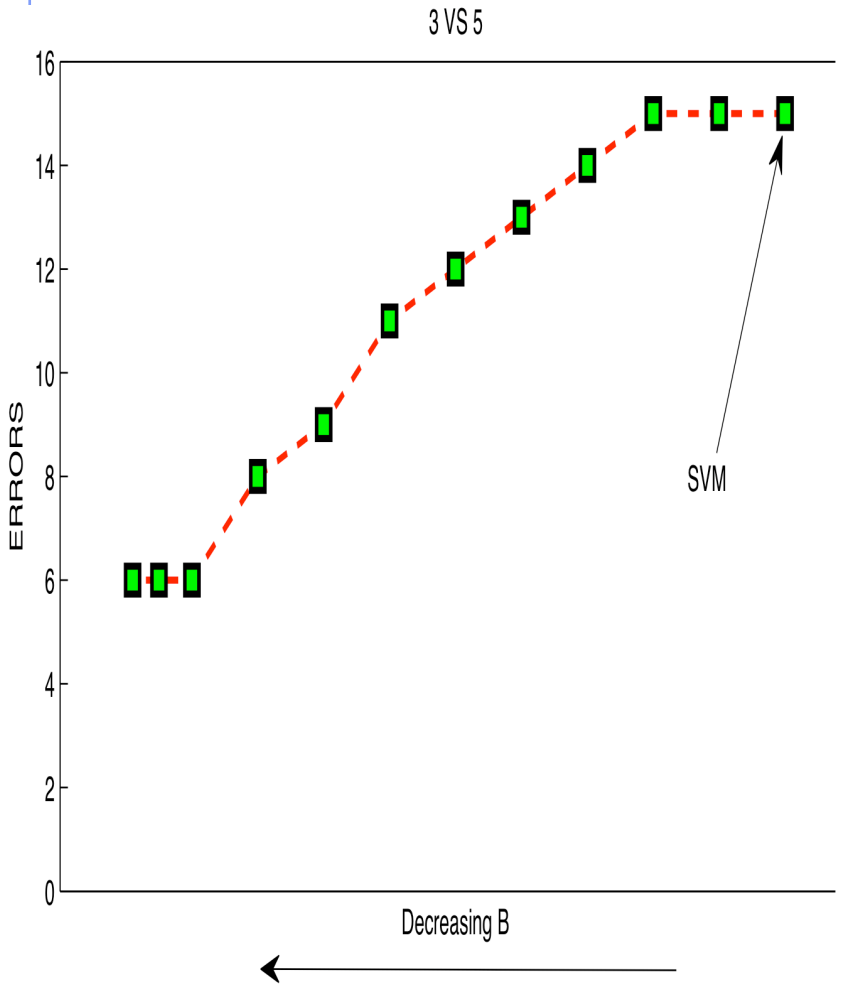
- Add the following constraints: $-B \leq w^T x_i + b \leq B$

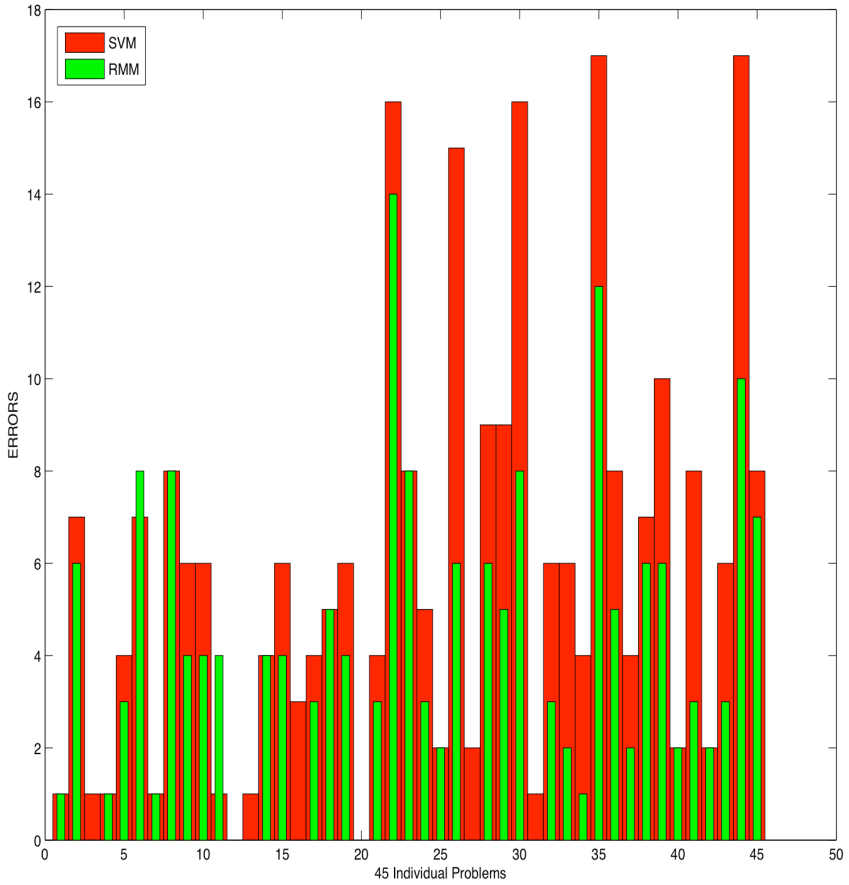
- This bounds the spread. Call it Relative Margin Machine.
- Above is still a QP, scales to 100k examples
- Can also be kernelized, solved in the dual, etc.
- Unlike previous SDP which only runs on ~1k examples

- RMM as fast as SVM but much higher accuracy...

Maximum Relative Margin

- RMM vs. SVM on digit classification (two-class 0,…,9)





Maximum Relative Margin

- RMM vs. SVM on digit classification (two-class 0,…,9)
- Cross-validate to obtain best B and C fro SVM and RMM
- Compare also to Kernel Linear Discriminant Analysis
- Try different polynomial kernels and RBF
- RMM has consistently lower error for kernel classification

		1	2	3	4	5	6	7	RBF
OPT	SVM	71	57	54	47	40	46	46	51
	Σ-SVM	**61**	48	41	36	35	31	**29**	47
	KLDA	71	57	54	47	40	46	46	**45**
	RMM	71	**36**	**32**	**31**	**33**	**30**	**29**	51
USPS	SVM	145	109	109	103	100	95	93	104
	Σ-SVM	**132**	**108**	99	94	**89**	**87**	**90**	**97**
	KLDA	132	119	121	117	114	118	117	101
	RMM	153	109	**94**	**91**	91	90	**90**	98
Full MNIST	SVM	536	198	170	156	157	141	136	146
	RMM	**521**	**146**	**140**	**130**	**119**	**116**	**115**	**129**