6998: Semidefinite Programming for Classification

Tony Jebara

joint work with Pannaga Shivaswamy

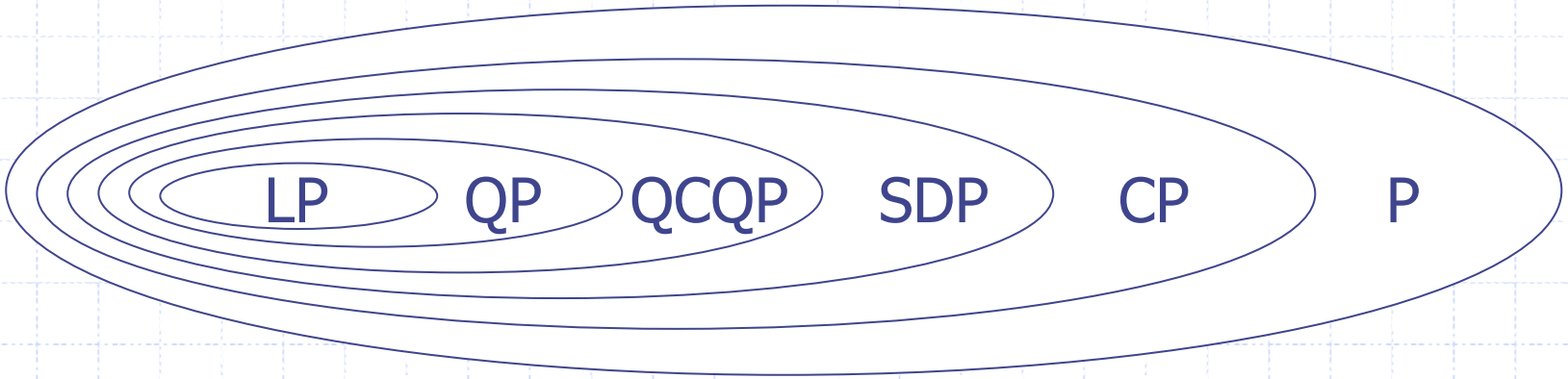
Outline

- LP < QP < QCQP < SDP < Convex Programming
- Structural Risk Minimization & VC Dimension
- Support Vector Machines
- Minimum Volume Ellipsoids
- Ellipsoidal Machines **(AISTAT 2007 Shivaswamy & Jebara)**
- Kernelized Ellipsoidal Machines
- Estimating Ellipsoidal VC
- SDP for General Spectral Functions

Optimization Tools

- A hierarchy of optimization problems for Machine Learning:

Linear Programming
<Quadratic Programming
<Quadratically Constrained Quadratic Programming
<Semidefinite Programming
<Convex Programming
<Polynomial Time Algorithms

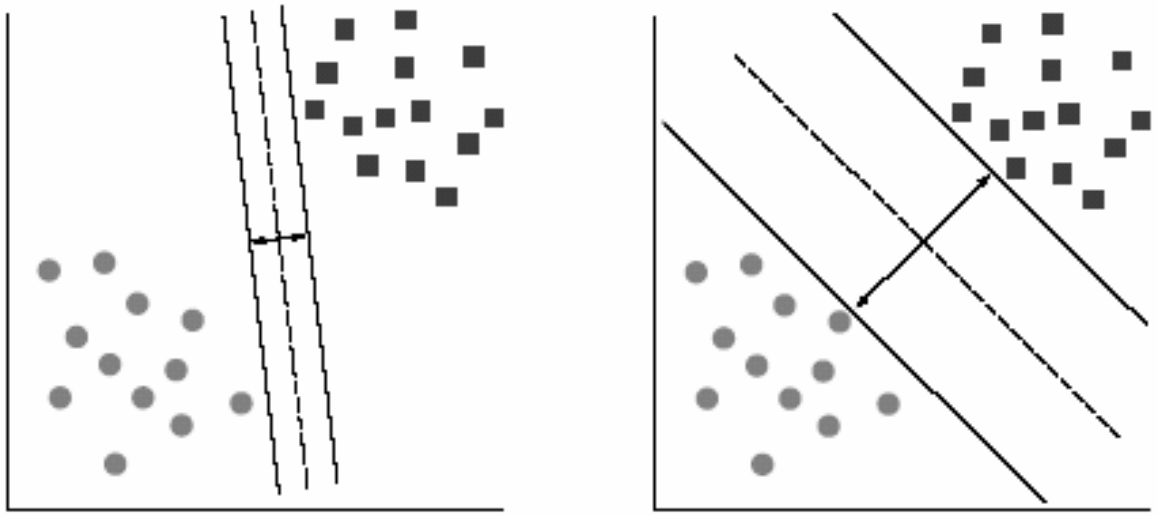


Optimization Tools

- LP < QP < QCQP < SDP < Convex Programming
- Matlab, Matlab, Mosek, Yalmip, Ellipsoid Method
- 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 $\min_K -\log|K| \;\; s.t. \; tr\left(C_i^T K\right) \geq \alpha_i \;\; \forall i, K \succeq 0$
- CP $\min_{\vec{x}} f\left(\vec{x}\right) \;\; s.t. \; g\left(\vec{x}\right) \geq \alpha$

Large Margins

- SVMs & VC Theory popularized large margin estimation



- 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

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

- Choose a loss function:

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

- Empirical Risk Minimization fits only to training data:

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

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

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

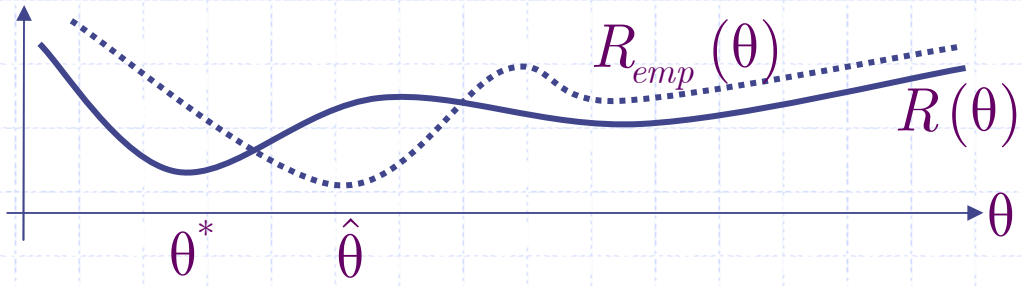
- We don't know the true P(x,y)

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

- Instead SVMs perform Structural Risk Minimization

Empirical vs. Structural Risk

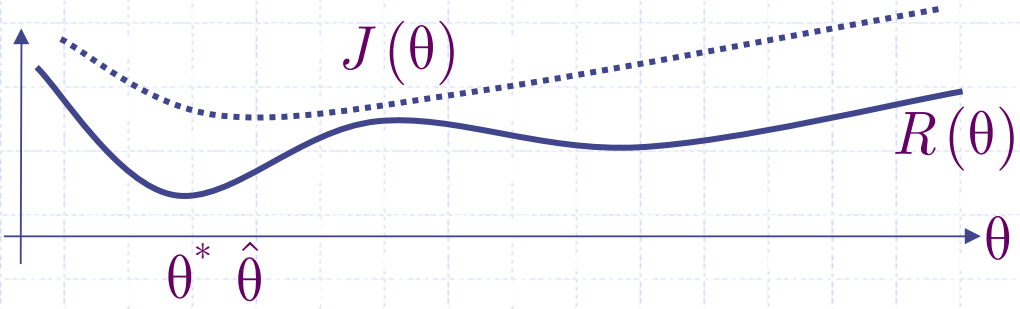
- ERM: inconsistent
  Not guaranteed.
  May do better on training than on test!



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

- 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 methods…
- Theorem (Vapnik): with probability 1-η the following holds:

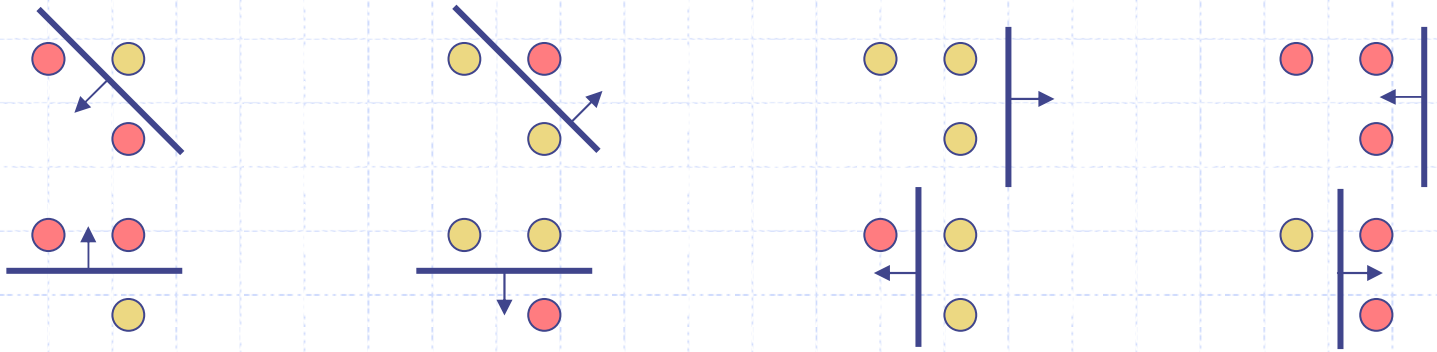
$$R(\theta) \leq J(\theta) = R_{emp}(\theta) + \sqrt{\frac{h\left(\log\left(\frac{2N}{h}\right)+1\right) - \log\left(\frac{\eta}{4}\right)}{N}}$$

N = number of data points

h = Vapnik-Chervonenkis (VC) dimension

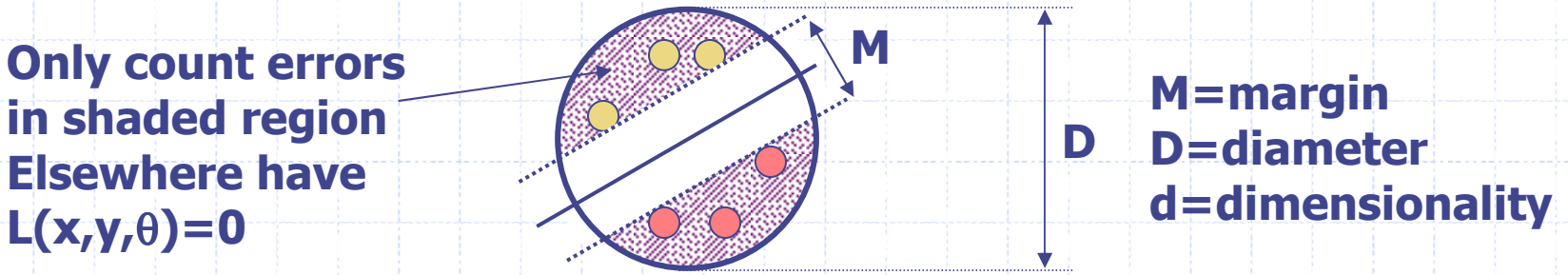
= capacity of the classifier class

- VC dimension of linear classifiers in d-dimensions is h=d+1
- Lines can shatter 3 points in 2d

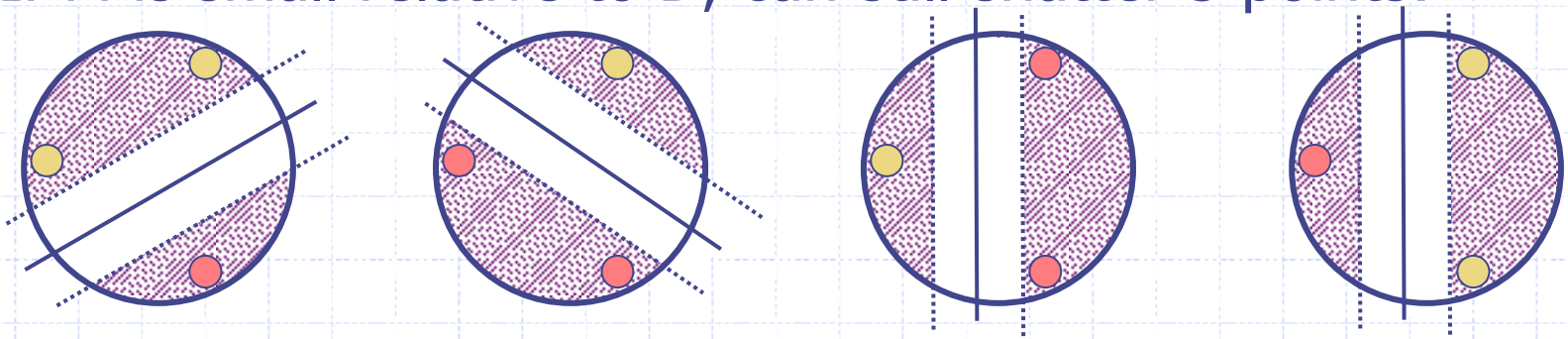


VC of Gap-Tolerant Classifiers

- 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

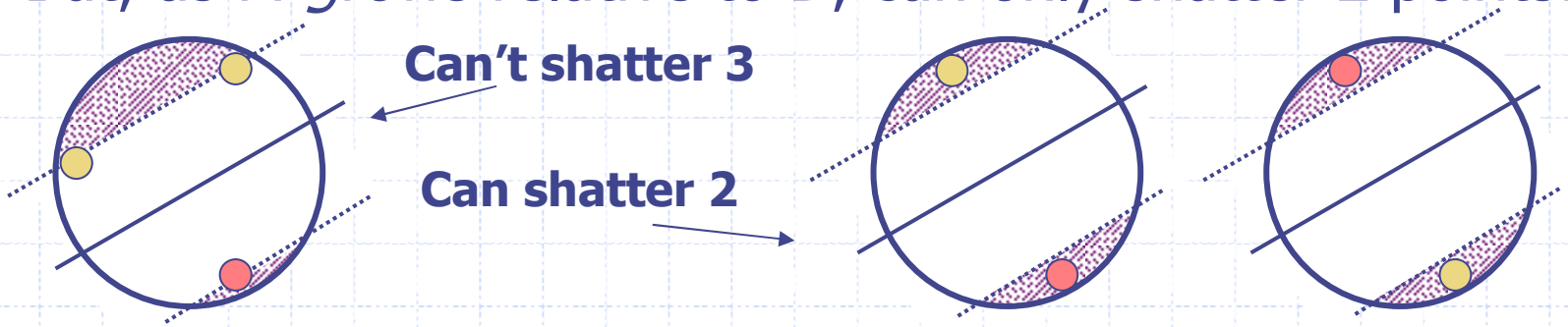


- If M is small relative to D, can still shatter 3 points:



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

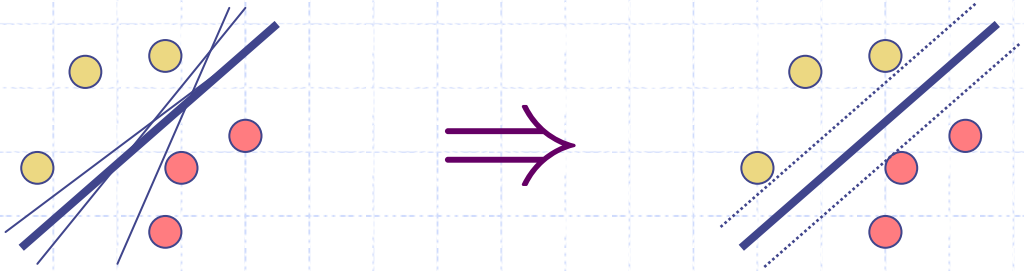
Support Vector Machines

- 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:

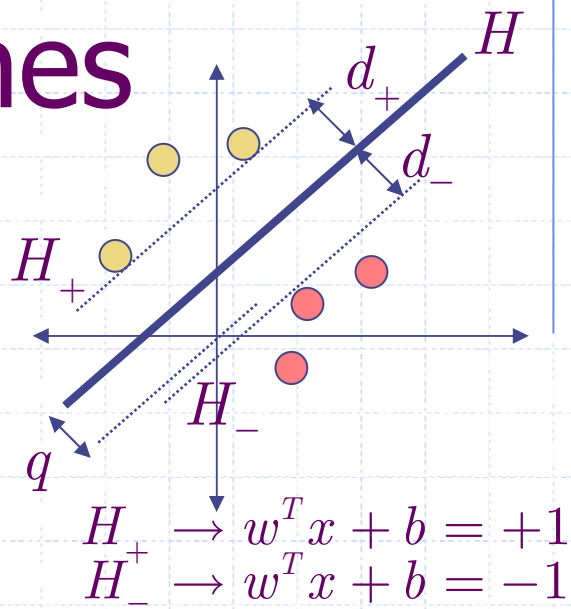
$$have\ \left\{\left(x_1,y_1\right),\ldots,\left(x_N,y_N\right)\right\}\ \ where\ x_i\in\mathbb{R}^D\ and\ y_i\in\{-1,1\}$$

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

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



Support Vector Machines



- 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:

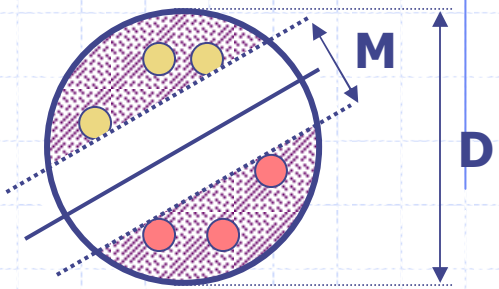
$$m = d_+ + d_-$$

- 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:

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



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

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

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

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

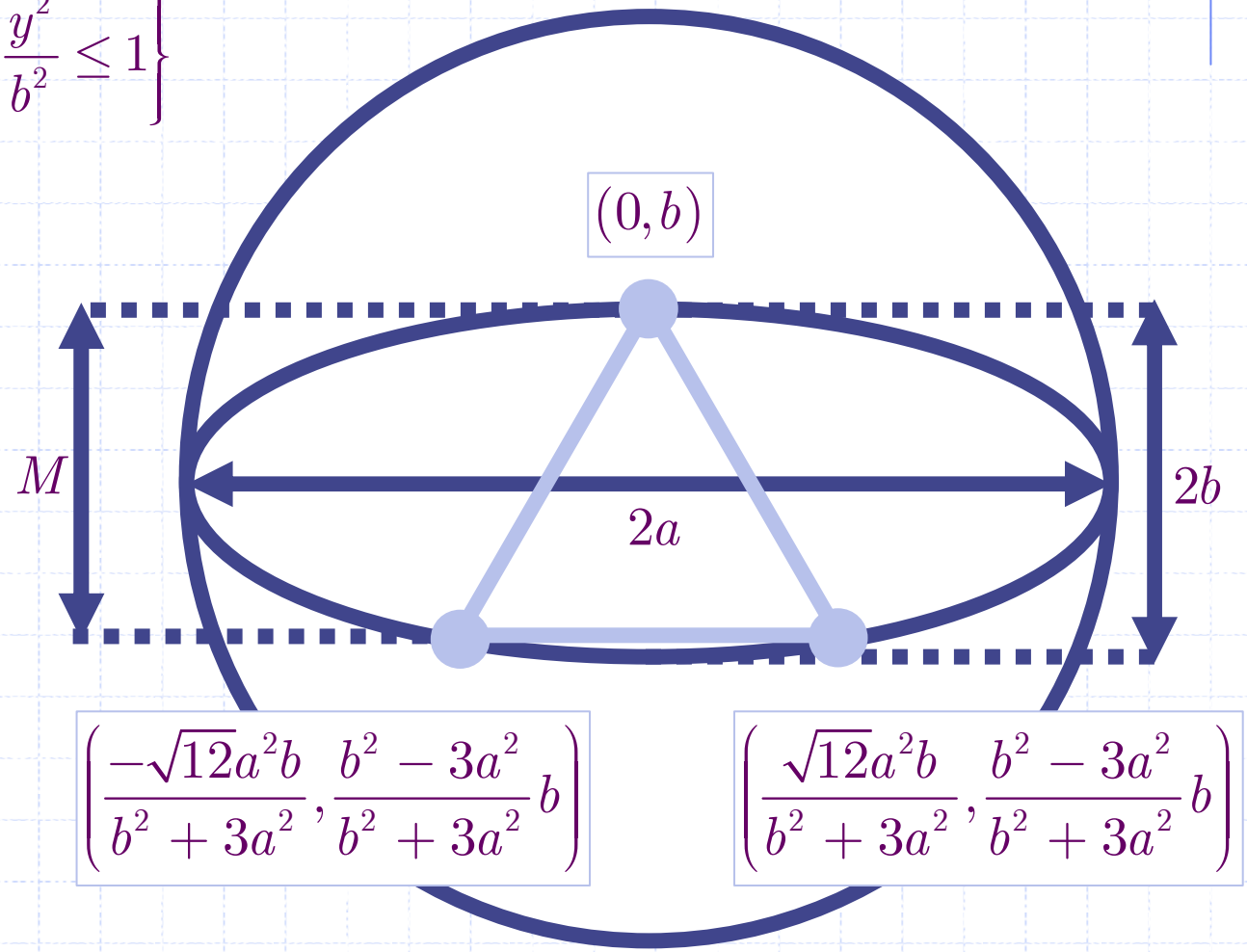
- How about a bounding ***ellipsoid***? Does anything change?

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

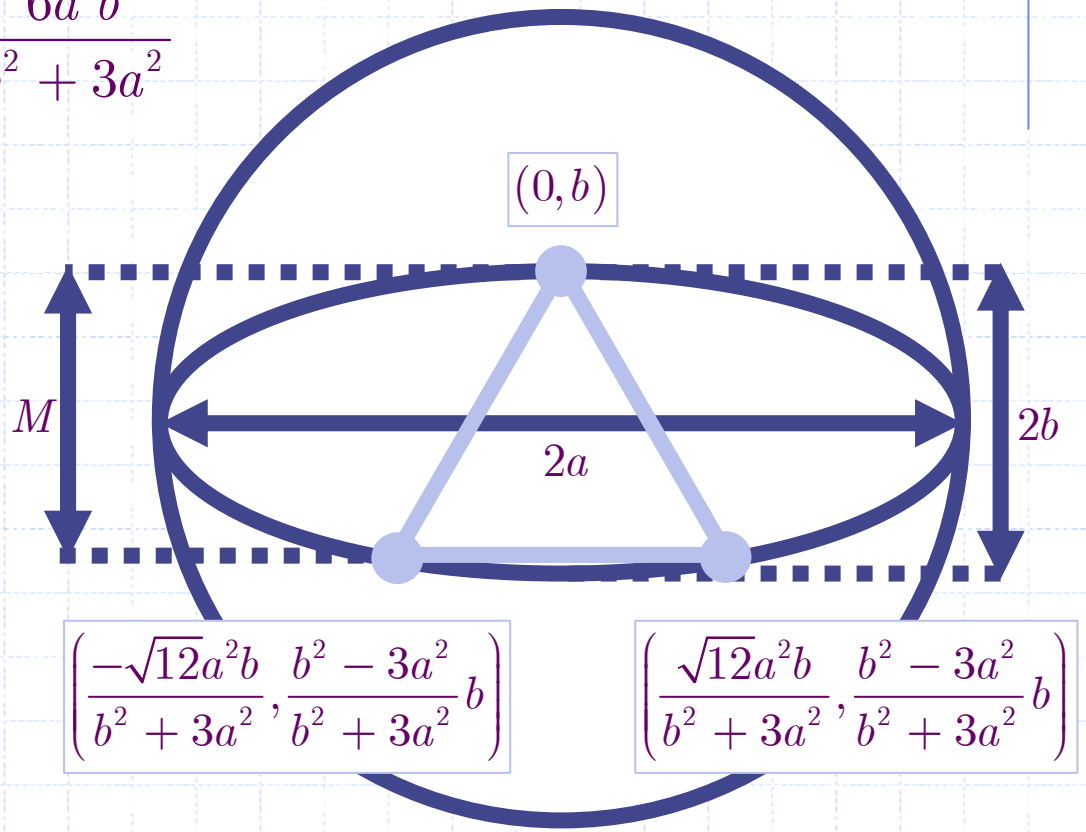


VC of Ellipsoidal Gap-Tolerance

- 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

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

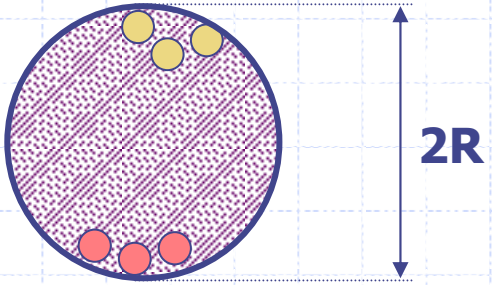
Sphere says VC=3
But, Ellipse says VC=2

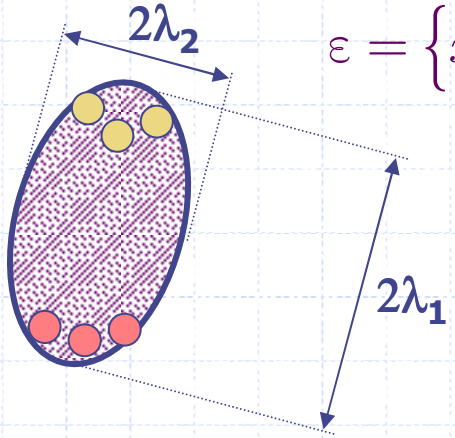


- For d>2, ellipsoids always give lower VC than spheres

Minimum Volume Ellipsoids

- Extend QP from bounding sphere to ellipsoid.





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

with volume related to $|\Sigma|$

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

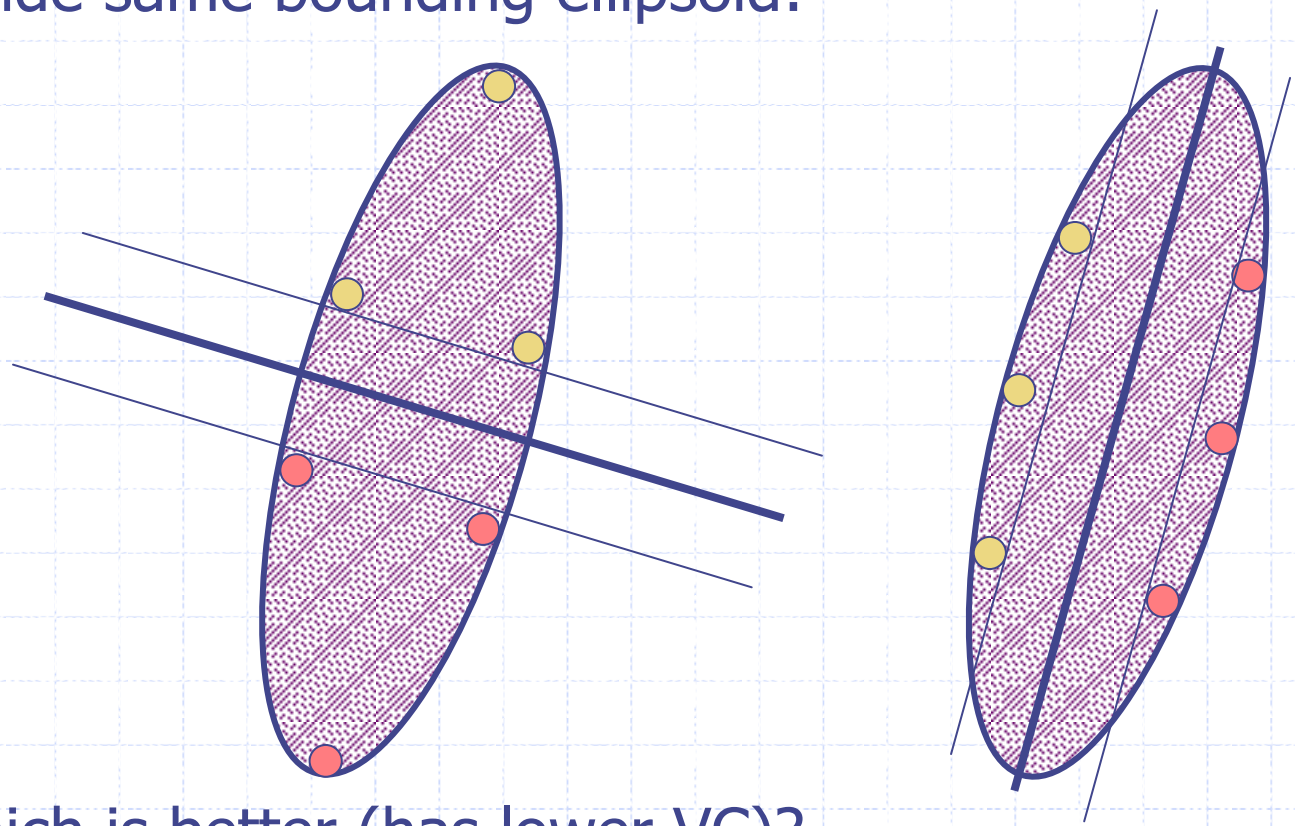
$$\min_{A,b,\tau} -\ln|A| + E\sum_i \tau_i \;\; s.t. \; (Ax_i - b)^T (Ax_i - b) \le 1 + \tau_i \;\; \& \;\; A \succeq 0$$

- Enforce quadratic SDP constraints via:

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

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:

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

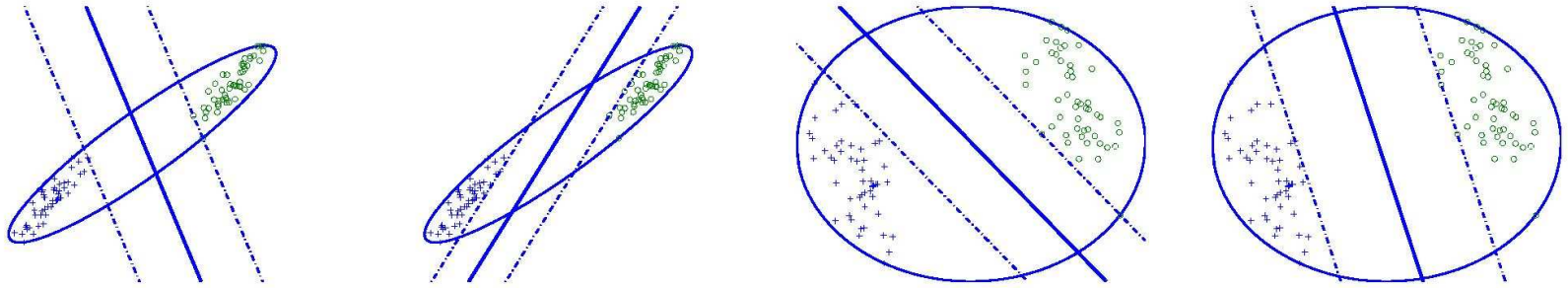


Figure 2: (a) Classical SVM solution on the data, (b) Ellipsoidal Machine solution on the data, (c) Classical SVM solution from the first plot after making the data spherical and (d) Ellipsoidal Machine solution from the second plot after making the data spherical.

- 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	$\mathbf{0.766 \pm 0.001}$

Kernelizing Ellipsoidal Machines

- Computations for minimum volume ellipsoid (see Kumar & Yildirim 2005) are kernelizable.
- Get nonlinear extensions to ellipsoidal margin.
- Follow a kernel PCA approach, consider inverse covariance:

$$\Sigma = \sum_i \lambda_i \sum_{j,k} \nu_j^i \nu_k^i \phi(x_j)\phi(x_k)^T \quad and \quad \Sigma^{-1} = \sum_i \frac{1}{\lambda_i} \sum_{j,k} \nu_j^i \nu_k^i \phi(x_j)\phi(x_k)^T$$

- Resulting SDP to solve becomes:

$$\min_{P,\tau,\Gamma} -\ln|P| + E\sum_i \tau_i$$

$$s.t. \sum_j \Gamma_{i,i} k(x_i, x_j)^2 \le 1 + \tau_i, \tau_i \ge 0, \forall i \quad and \quad P = \Gamma K \succeq 0$$

- Can easily kernelize transformed metric margin SVM:

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

- Experiments forthcoming, more dramatic gains over SVMs.

VC of Ellipsoidal Machines

- Can we estimate an EVM's VC dimension? Tricky…
- Run EVM, get margin M, get ellipsoid Σ. What is the VC?
- Shatter h points in Σ to get $M^*=\max_X(\min_Y(\text{margin}))$
- **Proposition**: If M > M* then VC<h.
- But max margin layout of h points in ellipsoid is **Tricky**.
- Instead set up i=1…h *equidistant* points to shatter in R^d.
- Make them vertices on regular polyhedron distance M apart
- Their equidistance M overestimates their max margin.
- Enclose polyhedron with rotated and scaled version of Σ

$$\min_{\delta,\hat{V}} \ \delta \ \ s.t.\ (x_i-\mu)^T \hat{\Sigma}^{-1}(x_i-\mu) \le 1, \hat{\Sigma} = \delta \hat{V} \Lambda \hat{V}^T,\ \hat{V}\hat{V}^T = I$$

$$where\ \Sigma = V \Lambda V^T\ and\ VV^T\ and\ \Lambda = diag(\Lambda)$$

- **Proposition**: if δ<1 then VC<h.
- Above optimization is not an SDP or even convex

VC of Ellipsoidal Machines

- We have the following optimization:

$$\min_{\delta,\hat{V}} \ \delta \ \ s.t. \ (x_i-\mu)^T \hat{\Sigma}^{-1} (x_i-\mu) \leq 1, \hat{\Sigma} = \delta \hat{V} \Lambda \hat{V}^T, \hat{V}\hat{V}^T = I$$

- The above is not an SDP or even convex

- ***CONJECTURE:***
- Can reformulate into optimization over PSD cone
- Define $K=\Sigma^{-1}$ and try this *spectral cost function*:

$$\max_K \sum_{j=1}^{d} \frac{\lambda_j(K)}{\lambda_j(\Sigma)} \quad s.t. \ (x_i-\mu)^T K (x_i-\mu) \leq 1, \ K \succeq 0$$

- Eigenvalues of K get weighted by desired relative scales
- Arbitrary spectral costs not directly solvable via SDP
- But, can be efficiently handled via an iterated SDP...

General Spectral Functions

- SDPs don't optimize arbitrary spectral functions. (any functions of the eigenvalues of matrix K).
- Variance or Trace costs (tr(K) or –tr(K)) are fine

$$\min_K \sum_i \lambda_i \;\; s.t.\; tr\left(KB_j\right) \geq c_j \; \forall j \;\; and \;\; K \succeq 0$$

- Volume of log-det costs (–log|K|) are fine

$$\min_K \sum_i -\log \lambda_i \;\; s.t.\; tr\left(KB_j\right) \geq c_j \; \forall j \;\; and \;\; K \succeq 0$$

- What about linear functions of eigenvalues of K?

$$\min_K \sum_i \alpha_i \lambda_i \;\; s.t.\; tr\left(KB_j\right) \geq c_j \; \forall j \;\; and \;\; K \succeq 0$$

- How to solve? Global optima?

SDP for Spectral Functions

- Not a standard SDP.

$$\min_K \sum_{i=1}^{d} \alpha_i \lambda_i \quad s.t.\ K \in \kappa$$

$$= \min_K \sum_{i=1}^{d} \alpha_i tr(\lambda_i)$$

$$s.t.\ K \in \kappa, Kv_i = \lambda_i v_i, \lambda_i \geq \lambda_{i+1}, v_i^T v_j = \delta_{ij}$$

$$= \min_K \sum_{i=1}^{d} \alpha_i tr\left(K v_i v_i^T\right)$$

$$s.t.\ K \in \kappa, Kv_i = \lambda_i v_i, \lambda_i \geq \lambda_{i+1}, v_i^T v_j = \delta_{ij}$$

$$= \min_K \min_V \sum_{i=1}^{d} \alpha_i tr\left(K v_i v_i^T\right)$$

$$s.t.\ K \in \kappa, \lambda_i \geq \lambda_{i+1}, v_i^T v_j = \delta_{ij}$$

- Variational upper bound on cost à Iterated Monotonic SDP
- Lock V and solve SDP K. Lock K and solve SVD for V.
- If alphas are increasing with i then convex.