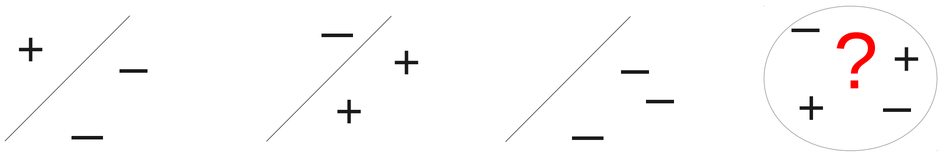
Classifiers and the VC-dimension

- VC-dimension (Vapnik & Chervonenkis) measures the discriminating capacity of a classification algorithm (family of classifiers)
- Somehow, the cardinality of the largest set of points that the algorithm can shatter (separate positive/negative points w/o error)
- Ex:



Basic Notions

- Def: Classifier f with parameters t shatters a set of n points $\{x_1, x_2, \dots , x_n\}$ iff
  for all +/- assignment, exists a t for which $f\,(x,t)$ makes NO error
- In the example, an affine classifier (with $t=(a,b)$) is able to shatter only 3 points in general position
- $f\,(x,t) = y^{est} = +/-1$ is the estimation for x given the parameter value t (cf. SVM notations)

Typical classifiers

- Linear
  $f(x,w) = \text{sg} <x|w> = \text{sg}\,(x.w)$
- Affine
  $f(x,w,b) = \text{sg}\,(<x|w>+\text{b}) = \text{sg}\,(x.w + b)$
- Circular
  $f(x,b) = \text{sg}\,(x.x - b)$
  could be any type of ellipsoid
- Others?

Statistical model (Vapnik's theory)

- Assumptions:
  1. Training points (x,y) are drawn i.i.d.
  2. Future test points also from same distribution
- Probability of misclassification:
  $R(t) = \text{E}\,(1/2\,|y - f(x,t)|)$
- Empirical training error (on set with L elements):

  $R^{emp}\,(t) = 1/L \sum_{i=1..L} 1/2\ \ |y_i - f(x_i,t)|$

  (Apply to SVMs)

Theorem (V+C)

- Let $0 \leq q \leq 1$. Assume losses take 0-1 values. Then, with probability $1-q$, the probability of misclassification is bounded by:

  $R(t) \leq R^{emp}(t)$
  $+ \text{sqrt}(1/L\ (H(\log(2L/H) + 1) - \log q/4)))$
- Note that $R(t)$ depends on the probability distribution $P(x,y)$ but not the upper bound
- H is the VC-dimension of machine f.
- The sqrt term is the VC-confidence

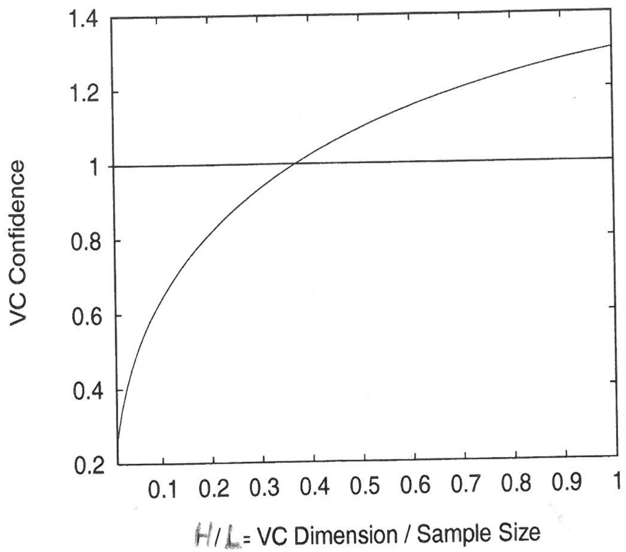
Comments

- The VC-dimension H is the maximum number of points in general position that can be shattered by the class (some H sets will not be)
- H can be infinite
- When H is finite, VC theorem allows comparison of machines for given q.
- VC-dimension in $\mathbf{R}^d$ is $d+1$, with oriented hyperplane classifiers. Proof by considering the convex hulls of +/- subsets

Infinite VC-dimension

- $f(x,t) = h(\sin tx)$, for x, t real; h step function
- Consider L points on the real axis:
  $x_i = 10^{-i}$, for $i = 1..L$
- Freely assign +/-1 values to y_i's
- Choose $t = \pi\ (1+1/2\ \Sigma_{i=1..L}\ (1-y_i)10^i)$
- Prove that the L points are shattered
- But would not work with regularly spaced ++-+
- VC-dim is infinite

VC-confidence



- VC-confidence is monotonic in H/L
- For $H/L > .37$, exceeds 1 with $q = .05$ and $L = 10000$
- Bound is NOT tight

Examples (1)

- k-NN classifier (nearest neighbour): k=1
  Check that VC-dimension is infinite and that empirical risk is 0
  Comments?

  Bound is useless
  but classifier can be quite good, unless???

Examples (2)

- Notebook classifier: L training size
  1. Store the exact results of $m \leq L$ training observations
  2. All other patterns in the same class, whatever their y value
  3. Prob of +/- is 1/2
  4. Empirical risk for first m is 0; 1/2 for others
  5. Actual error 1/2 ; VC-dim $H=m$
- $R(.)=m/4L \leq \log(2L/m)+1 - 1/m \log h/4$
  true for all h (prove it)

Non linear kernels

- Polynomial kernels
  $K(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}.\boldsymbol{y})^p$ for $\boldsymbol{x},\boldsymbol{y}$ in $\mathbf{R}^d$
  Note that for d=2, $K(\boldsymbol{x},\boldsymbol{y}) = \varphi(\boldsymbol{x})\ \varphi(\boldsymbol{y})$ with
  $\varphi(\boldsymbol{x}) = (x_1^2 ; \sqrt{2}\, x_1\, x_2 ; x_2^2)$
- Generalizes to higher d
  $[x_1^{r1} \ldots x_d^{rd}]\ \varphi(\boldsymbol{x})$ = sqrt ($p! / r1! \ldots rd!$), $p = \Sigma\, r$i
- Embedding vector space has dimension

  Bin(d+p+1, p), hence VC-dim = 1+...

Gaussian kernels

- $K(\boldsymbol{x},\boldsymbol{y}) = \exp(-\ ||\boldsymbol{x}^2 + \boldsymbol{y}^2|| / 2\sigma^2)$
- Decision functions on support vectors

  $f(s_j) = \Sigma_i\ \alpha_i\ y_i \exp(-\ ||s_i - s_j||^2 / 2\sigma^2) + b$

  independent if training points are distant enough
- Then ||Support set|| = ||Training set||
- VC-dimension is infinite