VC Dimension

Note Title — 4/5/2010

Additional notes on VC dimension (from Stat 232)

Some changes in notation

VC-dim $\quad h$

Risk - Error

These notes contain:

- The VC bound theorem.
- Discussion of the capacity term $\phi(h, n, \delta)$
- VC's for margins and kernels (too advanced, kernels haven't been mentioned yet.)
- The "trends" of VC bounds.
- Structural risk minimization

Note Title 11/19/2006

VC-Dimension

Risk: $R(\alpha) = \sum_{x,y} L(\alpha(x), y)\, P(x,y)$

Empirical Risk $R_{emp}(\alpha) = \frac{1}{N} \sum_{i=1}^{N} L(\alpha(x_i), y_i)$.

Minimize the empirical risk by constraining the set Γ of classifiers.

Vapnik-Chervonenkis (VC) dimension of a set Γ of classifiers.

VC = largest number of datapoints which can be shattered by the classifier set.

Shattered means that all possible dichotomies of the dataset can be expressed by a classifier in the set.

E.G. Hyperplanes can shatter any set provided $n \leq d+1$, d = dim space

Hence, VC-dimension of hyperplanes is $d+1$.

(2) Probabably Approximately Correct (PAC).

Let h be the VC dimension of a set Γ of classifiers.

If number of data samples $n \leq h$, then learning (i.e. generalization) is impossible – you can only memorize.

VC Bound Theorem: Suppose $n > h$

then, with probability at least $1-\delta$:

$$R[\alpha] \leq R_{emp}[\alpha] + \phi(h,n,\delta) \quad , \forall \alpha \in \Gamma$$

where $\phi(h,n,\delta) = \sqrt{\left(\frac{8}{n}\right)\left\{h \log\left(\frac{n}{h}\right) + h + \log \frac{4}{\delta}\right\}}$

(Several different ϕ's appear in different proofs)

Comment: require $\phi(h,n,\delta)$ to be small to ensure generalization.

requires: $n \gg h$ // small h/n

$n \gg |\log \delta|$ // small $|\log \delta|/n$

note: to require certainty, $\delta \to 0 \Rightarrow n \to \infty$.

(3) Comment: the derivation of VC bounds is complicated. The bounds require many approximations — so they are not tight.

Also, the proofs require ruling out all other possible interpretations of the data
→ this is too conservative.

VC Margins for Hyperplanes

Consider hyperplanes, $\underline{a} \cdot \underline{x} = 0$,
normalized w.r.t. data $\{\underline{x}_\mu : \mu = 1 \text{ to } n\}$
s.t. $\min_{\mu \in \{1,..n\}} |\underline{a} \cdot \underline{x}_\mu| = 1$.

Then, the set of hyperplane classifiers of form $\text{sign}(\underline{x} \cdot \underline{a})$, with $|\underline{a}| < \Lambda$
has VC dimension $h < R^2 \Lambda^2$ — inverse margin
where R^2 is radius of smallest sphere containing datapoints.
Intuition: low VC, if we require large margin & data lies in small region

(4) The same result also applies to kernels, because the kernel trick can be used.

The margin Λ, depends only on the kernel.
The radius of the minimum sphere R depends only on the kernel.

<u>Proof</u>, the minimum sphere can be found by a quadratic optimization problem.

$$L_p(R, \underline{x}^*) = R^2 - \sum_{\mu=1}^{n} \lambda_\mu \left(R^2 - |\underline{x}_\mu - \underline{x}^*| \right)$$

$\lambda_\mu \geq 0.$ Lagrange term enforce $R^2 \geq |x_\mu - \underline{x}^*|$ $\forall \mu$

$$\frac{\partial L_p}{\partial x^*} = 0 \Rightarrow \sum_\mu \lambda_\mu (\underline{x}_\mu - \underline{x}^*) = 0 \Rightarrow \sum_\mu \lambda_\mu \underline{x}_\mu \cdot \underline{x}^* = \sum_\mu \lambda_\mu \underline{x}^* \cdot \underline{x}^*.$$

$$\frac{\partial L_p}{\partial R} = 0 \Rightarrow 1 = \sum_{\mu=1}^{n} \lambda_\mu .$$

<u>Hence</u>: $L_d = \sum_{\mu=1}^{n} \lambda_\mu (\underline{x}_\mu - \underline{x}^*)^2$

$$L_d = \sum_{\mu=1}^{n} \lambda_\mu \underline{x}_\mu \cdot \underline{x}_\mu - 2 \sum_{\mu=1}^{n} \lambda_\mu \underline{x}_\mu \cdot \underline{x}^* + \sum_{\mu=1}^{n} \lambda_\mu |\underline{x}^*|^2$$

$$= \sum_{\mu=1}^{n} \lambda_\mu \underline{x}_\mu \cdot \underline{x}_\mu - \sum_{\mu,\nu=1}^{n} \lambda_\mu \lambda_\nu \underline{x}_\mu \cdot \underline{x}_\nu$$

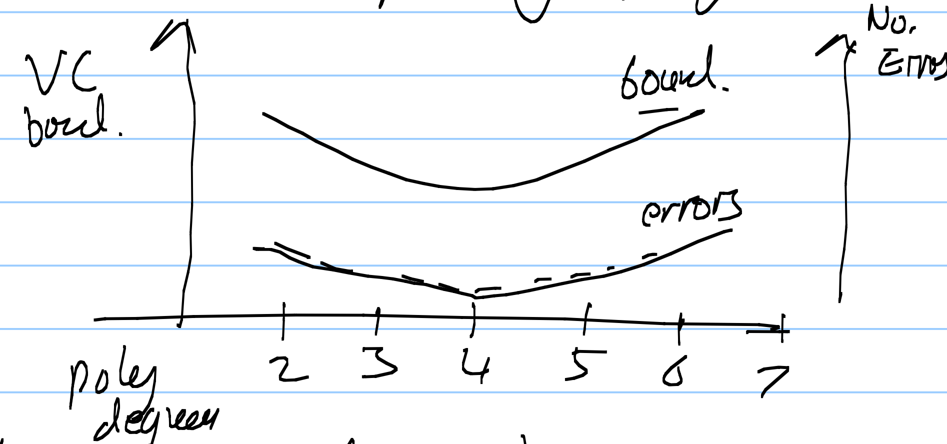
<u>depends</u> on dot product. Hence kernel trick applies.

(5) Application of VC-bound.

Use the Margin VC to decide which kernels will generalize best for learning to classify handwritten digits (US Post Office dataset).

→ For each kernel, solve the dual problem to estimate R.

Schölkopf did this comparing polynomial kernels.



The VC bound predicted the form of the errors (as a function of poly. degree).

But, the bounds very significantly worse than the errors.

(6) Structured Risk Minimization

Take the generalization into account when estimating the best classifier.

Statistics says, learn a classifier on a training dataset and use cross-validation to evaluate it.

VC theory says, evaluate the bound $\phi(h,n,S)$. and ensure there are sufficient number of samples to make $\phi(h,n,S)$ very small.

Alternative – Structured Risk Minimization.

Divide the set of classifiers into a hierarchy of sets $S_p \subset S_{p+1} \subset \ldots$
with VC dimensions $h_p, h_{p+1}, \ldots$

Select classifiers to minimize

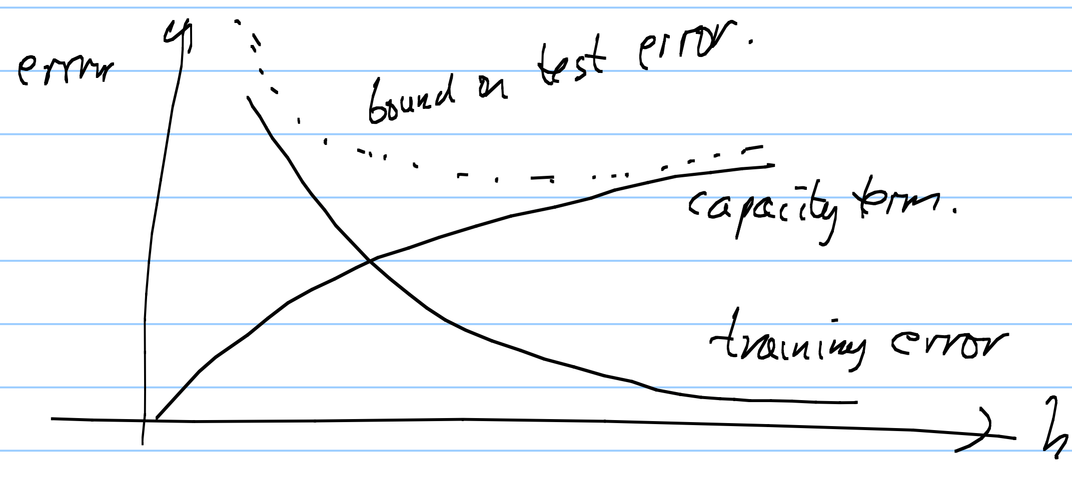
$$R_{emp}[\alpha] + \sqrt{(1/m)\{h_p(\log(2m/h)+1) + \log 4/\delta\}}$$

empirical risk + capacity term.

(7) A small capacity term ensures good generalization (with high probability)

Increasing the amount of data allows you to increase p

<u>But</u>, is the bound good enough? Does it give a fair balance between performance on the training set (R_{emp}) and generalizability (capacity term)? Does it overweight the importance of generalization?



$$\subset S_{p-1} \subset S_p \subset \ldots$$