A CONTINOUS COMPLEXITY ANALYSIS OF SUPPORT VECTOR MACHINES

Mark A. Kon, Boston University

Paradigm

1. Paradigm: Support vector machine as a continous algorithm

Fundamental objects: Basic arithmetic operations and evaluation operations, e.g., function evaluation.

How complex is it from a continuous complexity theory viewpoint?

2. The complexity theoretic model:

Given: Unknown probability measure $\rho(\mathbf{x}, y)$ on $\mathbb{R}^{d+1}$in space F of probability distributions on $\mathbb{R}^{d+1}$.

Goal: find the best functional relationship $y = f(\mathbf{x})$ reflected in $\rho(\mathbf{x}, y) \in F$.

Solution mapping- $S : F \to G =$ allowed class of functions f; defined by

$$S(\rho) = f.$$

We are given (Monte Carlo) partial information about ρ:

$$N\rho = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^{n},$$

where $\mathbf{z}_i \in \mathbb{R}^{d+1}$ chosen according to ρ.

Continuous Complexity Model

Goal: estimate the best function $f = S\rho \in G$.

Error criterion for estimate $\widehat{f} = \phi(N(\rho))$: start with risk function $V(f(\mathbf{x}), y)$ measuring distance between guess $f(\mathbf{x})$ and value y. Then compute

$$I(f) = E_\rho(V(\widehat{f}(\mathbf{x}) - y)) = \int_{\mathbb{R}^{d+1}} V\Big(\widehat{f}(\mathbf{x}) - y\Big) d\rho(\mathbf{x}, y).$$

Examples:

$$V(\widehat{f}(\mathbf{x}) - y) = \begin{cases} |\widehat{f}(\mathbf{x}) - y| \\ |\widehat{f}(\mathbf{x}) - y|^2 \\ (1 - \widehat{f}(\mathbf{x})y)_+ \quad \text{(categorical data } y = \pm 1\text{)} \end{cases}$$

(note $x_+ = \max(x, 0)$).

Continuous Complexity Model

Error of approximation $=$ distance from lowest risk:

$$e(\widehat{f}) = |I(\widehat{f}) - I(f_0)| = I(\widehat{f}) - I(f_0) \qquad (1)$$

where

$$f_0 \equiv \arg\min_{f \in G} I(f),$$

Goal: Find optimal algorithms for given information $N\rho$.

Note: error measure (1) (at least for finite dimensional hypothesis space G) equivalent to standard norm error.

Indeed assume $I(f)$ is twice differentiable function of $f \in G$.

Since $f_0 =$ minimum, Hessian matrix $H(f_0) \geq 0$ (pos. indef.).

Continuous Complexity Model

If in addition H positive definite, then for any norm $\|\cdot\|$ on G, $\exists$ constants c_1, c_2 s.t.

$$c_1\|f - f_0\|_G \leq I(f) - I(f_0) \leq c_2\|f - f_0\|_G.$$

So error $\sim$ a norm.

Given information

$$N\rho = \widehat{\rho} = \sum_{i=1}^{n} \delta_{\mathbf{z}_i}$$

and an algorithm $\phi(N(\rho)) = \widehat{f}$ for approximating best f from about ρ, define error of algorithm by:

$$e(\phi, \rho) = E_\rho\Big(R\Big[\widehat{f}\Big] - R(f_0)\Big),$$

where $\widehat{f} = \phi(N(\rho)) =$ best guess for f.

SVM Algorithm

3. Standard support vector machine algorithm:

Given data

$$\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n) = (\mathbf{x}_1, y_1; \mathbf{x}_2, y_2; \ldots; \mathbf{x}_n, y_n)$$

define empirical prob. dist. $\widehat{\rho}$ estimating ρ as:

$$\widehat{\rho} = \sum_{i=1}^{n} \delta_{\mathbf{z}_i}(\mathbf{z}),$$

($\delta_{\mathbf{z}_i}$ = point mass at $\mathbf{z}_i$) i.e., "best" guess of ρ; here $\mathbf{z} = (\mathbf{x}, y)$.

SVM Algorithm

Given cardinality of information n, use best guess $\widehat{\rho}$ to estimate minimizer of $I(f)$:

$$f_n = \arg\min_{f \in G} I_{\widehat{\rho}}(f) = \arg\min_{f \in G} \frac{1}{n} \sum_{i=1}^{n} V(f(\mathbf{x}_i) - y_i).$$

Denote $I_{\widehat{\rho}}(f) =$ empirical risk.

Thus goal is to estimate a minimizer over G of empirical risk $I_\rho(f)$ based on information $\widehat{\rho} = N(\phi(\rho))$.

For support vector machine (SVM): $y = \pm 1$ (classification of $\mathbf{x}$), and restrict $f \in G$ to be affine (for linear partition of classes in space):

$$G = \{f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b : \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\}.$$

4. Algorithmic error

Complexity-theoretic bounds on SVM algorithms: what is error as information cardinality $n \to \infty$?

SLT bound: intuitively define

VC dimension of $V(f(\mathbf{x}), y)$

$=$ capacity of this set of functions (as $f \in \mathcal{G}$ varies)

Algorithmic Error

Def. 1: Indexed family of functions $G = \{g_\beta\}_{\beta \in B}$ on space R

separates a finite set $X = \{x_i\}_{i=1}^n \subset R$ if $\forall\ Y \subset X$, $\exists \beta \in B$, $\alpha \in \mathbb{R}$ s.t.

$$g_\beta(y) - \alpha > 0 \text{ iff } x \in Y;$$

that is, all finite subsets Y can be separated out by some $g \in G$.

Algorithmic Error

Def. 1: ***VC dimension h*** of family G of functions on R

$=$ cardinality of the largest set of points X separated by G.

Need error estimates independent of distribution $\rho(\mathbf{z}) \in F$.

Let $p > 2$ and

$$\tau = \sup_{f \in G} \frac{\|V(f(\mathbf{x}), y))\|_p}{\|V(f(\mathbf{x}), y)\|_1},$$

and

$$a(p) = \frac{1}{2^{1/p}} \left(\frac{p-1}{p-2} \right)^{\frac{p-1}{p}}.$$

Algorithmic Error

Finally, define

$$\mathcal{E} \equiv 4\frac{h(\ln\frac{2n}{h}+1)-\ln(\frac{\eta}{4})}{n},$$

where h is the VC dimension of the set of functions $\{V(f(\mathbf{x}),y)\}_{f\in G}$.

Then

Theorem (Vapnik): *For any distribution ρ on $\mathbb{R}^{d+1}$, and any loss function V, we have with probability at least $1-\delta$,*

$$\text{error } \epsilon \equiv R(f_n) - \inf_{f\in\mathcal{H}} R(f) \leq \frac{J\tau a(p)\sqrt{\mathcal{E}}}{\left(1-\tau a(p)\sqrt{\mathcal{E}}\right)} + O\left(\frac{1}{n\ln n}\right).$$

where $J = \inf_{f\in\mathcal{H}} R(f)$.

Algorithmic Error

Recall ϵ may be replaced by a norm error if G finite dimensional.

Remark: Note that for n large we have

$$\epsilon \leq K\sqrt{\mathcal{E}} = K\sqrt{4\frac{h(\ln\frac{2n}{h}+1)-\ln(\frac{\eta}{4})}{n}}$$

Like Monte Carlo with extra $\ln\frac{2n}{h}$ term in numerator;

Uniform Monte Carlo result - to find the overall minimizer f_0 of $R(f)$ within ϵ we need to estimate $R(f)$ uniformly in f, within ϵ - hence the ln term. Note also result is uniform in distribution ρ.

Algorithmic Error

ϵ-complexity: Now derive information complexity of risk minimization. We see that the above uniform bounds give an inverse relationship as follows.

Define the δ-probabilistic ϵ-complexity n of identifying the risk-minimizing function by:

$$n = \text{comp}(\epsilon)$$

$$= \inf_{n'}\{|R(f_{n'}) - R(f_0)| < \epsilon \text{ with probability at least } 1 - \delta\}$$

Algorithmic Error

Now invert above relationship between ϵ and n:

$$\epsilon \leq \frac{(J\tau a)\sqrt{\mathcal{E}}}{(1-\tau a\sqrt{\mathcal{E}})} + O\left(\frac{1}{n}\right)$$

yielding complexity

$$n(\epsilon) \leq 2(J\tau a)^2 h \frac{\ln(1/\epsilon^2)}{\epsilon^2} + o\left(\frac{\ln 1/\epsilon}{\epsilon^2}\right).$$

$=$ information complexity of approximating $f_0 = \arg\inf_f I_\rho(f)$ within error ϵ, using algorithm

$$\phi(N(\rho)) = \arg\inf_{f\in G} I_{\widehat{\rho}}(f)$$

which minimizes empirical risk function. Note that probability δ of failure of approximation appears in higher order terms.

5. Support vector machine: In this case we assume

$$G = \{\text{affine functions } f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b\}$$

We assume $y = \{\pm 1\}$ (classification).

Want $f \in G$ which minimizes the loss with

$V(f(\mathbf{x}), y) = (1 - f(\mathbf{x})y)_+$, i.e.,

$$R(f) = \int_{\mathbb{R}^{d+1}} (1 - f(\mathbf{x})y)_+ \, d\rho(\mathbf{x}, y).$$

The affine function f which minimizes the empirical risk $\frac{1}{n}\sum_{i=1}^{n} V(f(\mathbf{x}_i, y_i))$ forms a plane which separates the data well.

6. Is the SVM algorithm optimal?

Note: we have for SVM error,

$$\epsilon = O\left(\sqrt{\frac{\ln n}{n}}\right);$$

and complexity

$$n(\epsilon) = O\left(\frac{\ln 1/\epsilon}{\epsilon}\right)$$

(with prob. $1 - \delta$).

Note that the best possible case is for a function class G with only one non-trivial function f. In this case (standard Monte Carlo without uniformity in f) we have (again with probability $1 - \delta$) :

Is SVM Optimal?

$$\epsilon = \Omega\left(\frac{1}{\sqrt{n}}\right); \quad n(\epsilon) = \Omega(\epsilon^2)$$

Thus we conclude:

Theorem 1: The SVM is within a logarithm $\ln(1/\epsilon)$ term of being optimal, i.e., of having optimal ϵ-complexity with probability $1 - \delta$.

Can we improve on the logarithm term? Yes, at least in some cases.

7. Improvement of VC bounds

We can improve the bounds if we restrict ourselves to "almost" all of $G =$ {affine functions f}.

Specifically, let us consider the space G_M consisting of all affine functions $\mathbf{w} \cdot \mathbf{x} + b$ with slopes $|\mathbf{w}|$ less than or equal to M (or any other compact subset $G_M \subset G$).

On G_M the functional

$$B(f) = R_\rho(f) = \int_{\mathbb{R}^{d+1}} V(f(\mathbf{x}), y)\, d\rho(\mathbf{x}, y)$$

is continuous in $f \in G_M$.

Improving VC Bounds

Note for any fixed $f \in G_M$, from Monte Carlo:

$$|R_{\widehat{\rho}}(f) - R_\rho(f)| = \left| \frac{1}{n} \sum_{i=1}^{n} V(f(\mathbf{x}_i), y_i) - \int V(f(\mathbf{x}), y) d\rho(\mathbf{x}, y) \right|$$

$$= O\left(\frac{1}{\sqrt{n}} \right).$$

Now note if $V(f(\mathbf{x}_i, y_i)) = P(f(\mathbf{x}) - y)$ where $P =$ polynomial, then

$$R_\rho(f) = \int P(\mathbf{w} \cdot \mathbf{x} + b - y) d\rho(\mathbf{x}, y)$$

$$= \int \sum_i c_i a_i(x_1, \ldots, x_n, y) d\rho(\mathbf{x}, y)$$

Improving VC Bounds

where a_i are monomials in $x_1, \ldots, x_n, y$. Now note that since for each a_i:

$$|R_{\widehat{\rho}}(a_i) - R_\rho(a_i)| = O\left(\frac{1}{\sqrt{n}}\right),$$

the same follows for their finite sum $P(\mathbf{w} \cdot \mathbf{x} + b - y)$, uniformly in bounded **w.**

Improving VC Bounds

Thus as above with probability $1 - \delta$:

Theorem 2: *For any set $G_M \subset G$ of affine functions of bounded slope, and a polynomial V, we have*

$$R(f_n) - \inf_{f \in \mathcal{F}} R(f) = O\left(\frac{1}{\sqrt{n}}\right).$$

Thus, information complexity of ϵ-approximation is of order ϵ^{-2} for SVM, i.e., is almost optimal.

Further, the algorithm of empirical risk-minimization is complexity-almost optimal.

Thus log term in asymptotic error, if there, comes from specific small set of possible $f \in G_M$ for V as above.

Improving VC Bounds

We note that for a compactly supported ρ, we can approximate any V uniformly by polynomials, so that

Theorem 3: *For a complactly supported ρ and any continuous V, there exists a V^* which is arbitrarily close to V such that the probability $1-\delta$ information complexity of an SVM using error criterion V^* is of order $\frac{1}{\sqrt{n}}$.*

8. Use of scaled families of algorithms

Increased information generically corresponds to increased algorithmic complexity -

This occurs, for example in spline approximation - more data points means approximation in spline space with more knots.

A science of scaling the two is sometimes useful.

Example: If I have have a million data points then I don't want to try linear regression (i.e., an approximation space G only consisting of linear functions). I want to enlarge the space to include more parameters, e.g., quadratics and cubics.

One popular computational model is now to scale algorithms ϕ with cardinality of information via size h of range in G.

Scaled Algorithm Families

Specifically, for information $N : F \to Y$ of cardinality n, choose algorithm ϕ_n: $Y \to G$ whose range G_n has dimension $h(n)$, with scaling $h(n)$ chosen so that the error of approximation is minimized.

The error for such an algorithm

$$\epsilon = R(f_n) - R(f_0) = \underbrace{(R(\widehat{f}_n) - R(f_k))}_{\text{estimation error } \epsilon_{\text{est}}(n)} + \underbrace{(R(f_k) - R(f_0))}_{\text{approximation error } \epsilon_{\text{app}}(h)},$$

where

$\widehat{f}_n =$ minimizer of empirical risk with n data points

$f_n =$ closest element of G_n to true f_0

Note: $\epsilon_{\text{est}}(n)$ decreases at some rate as $n \to \infty$
$\epsilon_{\text{app}}(h)$ decreases at some rate as $n \to \infty$.

Scaled Algorithm Families

But: if h too large for n, have overfitting - estimation error $\to 0$ - we are in the wrong space.

Goal: increase $h = h(n)$ so not enough dimensions $h(n)$ in G_n for $\epsilon_{\text{est}} =$ zero.

This is scaling of complexity $h = h(n)$ of ϕ with information complexity n.

Colloquially: keep the number of free parameters $(h =$ alg. comp.) scaled to amount of data $(n =$ infomration complexity$)$

(Note: in e.g. Kon and Plaskota, 2000: algorithmic complexity $=$ neural complexity).

9. Scaling of algorithms: applications to SVM

More on scaling n and h: Recall

$$\epsilon \leq K\sqrt{\mathcal{E}} = K\sqrt{4\frac{h(\ln\frac{2n}{h}+1)-\ln(\frac{\eta}{4})}{n}}$$

This suggests: scale n with h so h/n is constant or decreasing

Note h must increase (not just n) for approximation error $e_{\text{app}}(k) \underset{k\to\infty}{\longrightarrow} 0$.)

Thus want $\widehat{f}_n = \phi_n(N_n(\rho)) \in G_n = \text{ran}(\phi_n)$, with $h = \text{VC dim}(G_n)$ scaled as follows.

Let $P_k =$ polynomials of degree k on $\mathbb{R}^d$.

Nonlinear SVM

Usual SVM algorithm:

$$\phi_1 : \text{ data } \{\mathbf{z}_i\} \to \text{affine polynomials } P_1.$$

Target space P_1 (along with y and composed with V) has VC dimension $\leq d + 2$.

For **nonlinear SVM**: Extend P_1 to P_k of appropriate dimension

done by extending data vector

$\mathbf{x} = (x_1, \ldots, x_n) \to$
$\tilde{\mathbf{x}} = (\text{all possible monomials of degree } k \text{ in components } x_i)$

Then use standard SVM algorithm on $\tilde{\mathbf{x}}$.

Nonlinear SVM

Let dim $P_k = D(k)$.

Therefore, scale k so :

algorithmic complexity $= VC$ dim. $= h \leq$ dim $P_k + 2$
scales with information complexity n

For this scaled family of algorithms

Theorem 4: *For the above scaled family of SVM algorithms, we have that the δ-probabilistic error satisfies*

$$\epsilon_n \leq O\left(\frac{\tau a(p)\sqrt{\mathcal{E}}}{1-\tau a(p)\sqrt{\mathcal{E}}}\right) + O\left(\frac{1}{n}\right) + \epsilon_{app}(h),$$

where $\mathcal{E} = 2\frac{\ln(\alpha+1)}{\alpha} - 4\frac{\ln(\delta/4)}{n}$, *with* $\alpha = \frac{2n}{h}$.

Nonlinear SVM

Note in some cases dimensional reduction appropriate to prune the set $G_n = \text{Ran } \phi_n$, so nonlinear SVM (of appropriate dimensionality) on such reduced data sets can be attempted.

Example: (bioinformatic data)

Advantage of NLSVM (i.e., polynomial separators) reduces in typical (Gaussian) situation to question:

given two multivariate Gaussian distributions $\rho_1 \equiv N(\mu_1, \Sigma_1)$ (those **x** for which $y = 1$) and $\rho_2 \equiv N(\mu_2, \Sigma_2)$ (those with $= -1$), what is shape of optimal separator between the two?

Nonlinear SVM

Optimal separator $=$ function $f(\mathbf{x})$ s.t. the probability of error

$$R(f) = P_1(f(\mathbf{x}) < 0) + P_2(f(\mathbf{x}) > 0)$$

is minimized.

Or: weighting of false positives versus false negatives, where the new risk function is

$$R_1(f) = \alpha_1 P_1(f > 0) + \alpha_2 P_2(f < 0),$$

with $\alpha_1 + \alpha_2 = 1$.

To extent risk for linear SVM large, decision to use a NL SVM makes sense.

Nonlinear SVM

Note: to identify optimal f among *all* functions, observe that the surface $\rho_1(\mathbf{x}) = \rho_2(\mathbf{x})$ is optimal. This is determined by the identity

$$-\ln\alpha_1 + \ln \det \Sigma_1 + \frac{1}{2}\langle \mathbf{x} - \mu_1, \Sigma_1^{-1}(\mathbf{x} - \mu_1)\rangle$$

$$= -\ln\alpha_2 + \ln \det \Sigma_2 + \frac{1}{2}\langle \mathbf{x} - \mu_2, \Sigma_2^{-1}(\mathbf{x} - \mu_2)\rangle,$$

Note in this case surface is quadratic, and use of quadratic (P_2) SVM is appropriate.

In fact we expect generically, in cases where distributions of positive and negative classes have different covariances, may be significant improvement using quadratic SVM over a linear one.

Nonlinear SVM

This suggests a criterion for determining whether a quadratic SVM is appropriate for a given data set : if we determine empirical covariance matrices $\widehat{\Sigma}_1$ and $\widehat{\Sigma}_2$ for the two data sets (assuming they are sufficiently large to allow for accurate estimates), then if $\widehat{\Sigma}_1 - \widehat{\Sigma}_2$ large, we expect a quadratic discrimination surface.

10. Example: bioinformatic data

Wisconsin cancer database,

1. Standard SVM applied to the 9 input variables to predict cancer malignancy (± 1)

Data summarized below.

349 randomly chosen examples from 699 total,

349 randomly chosen elements of test set.

The first test via SVM had an error rate of 13.75% on the test set.

When the three most useful variables were extracted, they themselves had an SVM error rate of 32.39%.

Example

When the fully nonlinear SVM of degree 2 was applied to these three variables, the total error rate went down to 8.60%.

Machine \ Error rate	FP	FN	TP	TN	ERR	%ERR
9-variable SVM	37	11	107	194	48	.1375
3-variable SVM	41	72	44	192	113	.3239
3-variable NL SVM	29	1	117	202	30	.0860

F/TP-false/true positive; F/TN-false/true negative; ERR-total errors;

Example

Currently applying the same methodologies to identifying transcription initiation sites in the genome from genetic behaviors.