How sure are you that large margin implies low VC dimension?

John Mount*

January 19, 2015

Abstract

A standard claim about Support Vector Machines (SVMs) is that a large margin built during training ensures (with high probability) good generalization error.[1]

The standard argument how large margin decreases generalization error (or excess error) is:

- Large margin implies bounded Vapnik–Chervonenkis (VC) dimension.
- Bounded VC dimension implies high probability of low generalization error.

Notice in the above that VC dimension is used only as a summary to connect large margin to low generalization error; so practitioners *can* avoid working with VC dimension if they are given a finished theorem relating large margin directly to generalization error.

The pedagogical issues are:

- Even people who have taken a good machine learning course (or consulted a good machine learning book) may have never seen the rigorous quantified versions of the above statements.
- The standard definition of VC dimension actually has a pretty hard time encoding concepts like margin.

So while large margin may indeed imply low VC dimension, we suspect most of our readers have never actually seen a rigorous proof of that step. In this article we talk a bit about what has become the standard treatment and its place in the textbook (or secondary) literature. We are going to center this write-up around [Cortes and Vapnik, 1995][2] (the paper that introduced soft margin support vector machines) and [Vapnik, 1998][3] (a primary source of results).

*email: mailto:jmount@win-vector.com web: http://www.win-vector.com/

[1]Note: for conciseness in this note we are using "generalization error" to mean the "excess error," or how much the unknown true error rate of a classifier exceeds the error frequency observed during training (this is the definition given in the Wikipedia). However, many references use generalization error to mean the unknown true error rate of a classifier (in particular [Hastie et al., 2009] and our own [Zumel and Mount, 2014]). This reflects different fields (statistics versus machine learning) historically using the term differently. For this note we are concerned with the change in error rate (not the total error rate), so it makes sense to use the definition closest to what we are trying to discuss.

[2]Cortes, C.; Vapnik, V. (1995) "Support-vector networks". Machine Learning 20 (3): 273.

[3]Vladimir N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

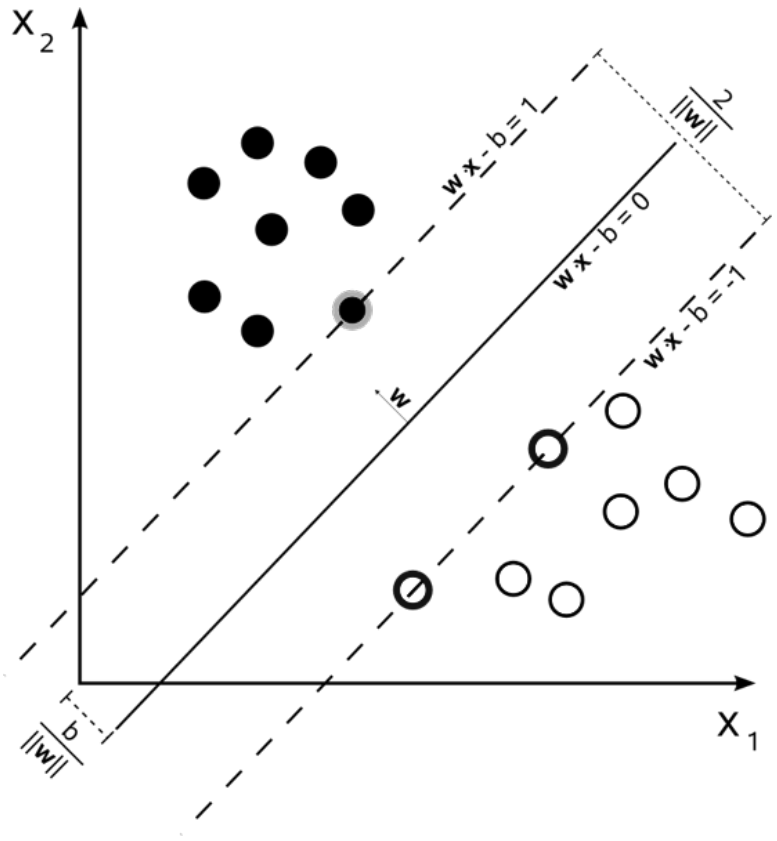

Figure 1: Margin illustrated (figure from Wikipedia: Support_vector_machine).

This write-up started as a recreational research project examining what the proofs look like if you go to a primary text [Vapnik, 1998]. We felt this book, given its authorship and history, could be considered a primary source; and in fact would be an easier and more unified way to get at early results than attempting to unify initial papers and proceedings. In reviewing sections of this book we found three key proofs about generalization error that are often only cited in reference (which we work through variations of in this write-up), *and* one mis-match or gap in argument (which we will talk about towards the end of this write-up). We have found other sources that have come to similar conclusions.

To promote understanding we are going to informally outline (and re-outline) the overall proof plan several times.[4] There are a lot of steps, and it takes a lot of careful attention to keep clear *both* what we are trying to prove in a given sub-step *and* why we need a give sub-step proven.

Obviously *this* work is at best merely a secondary source.

[4]In fact this note started as notes to keep all of the material in sensible order.

Contents

1 Defining terms

1.1 The problem

For this article we will consider the problem of learning a function that classifies data drawn from $\mathbb{R}^k$ into two classes. The idea is: we fit a support vector machine on training data (data where both the effective variables x and class assignments y are known) and then can use the learned classifier on new data that is exchangeable with the training data yet the class assignment is not visible. Any model (concept or hypothesis) will have an observed error frequency during training (call this r) and an (unobserved) true error rate on future data (call this q). We will call the difference d "the excess error rate" or "the generalization error" and we have $d = q - r$. Throughout we will specify a (small) probability p and ask for procedures that pick a model such that q is small with probability at least $1 - p$ (remember we observe r, but not q or d). This is done by appealing to the empirical risk minimization principle. We pick a hypothesis that has small r and try to bound d as also likely being small.

Some more symbols we will try to use consistently throughout this note are: k for the dimension of the space of effective (input) variables, n the number of training samples, and m for the number of hypotheses/concepts/models we are considering during training (when that set is finite).

A classic machine learning example is a collection of hand-written digits where the x-variables are vectors representing the rasterized images of the digits and the class assignment indicates whether the image represents the number 8 or not. If we can build a machine that correctly recognizes drawings of the number 8 we would say we have learned what a 8 looks like. This is the crux of supervised machine learning: by working with some labeled training examples we may deploy a procedure that works well assigning labels to future examples. Or from the US Post Office's point of

view: some time spent examining old envelopes can help build a machine that reads hand written addresses and zip-codes on future mail.

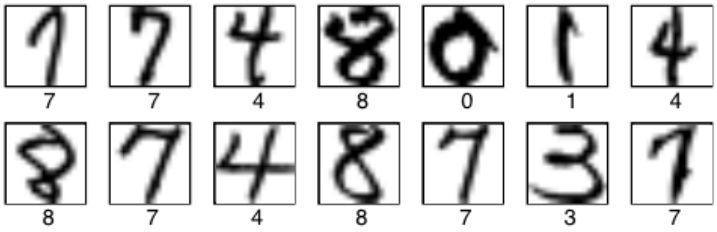

Figure 2: Examples of labeled rasterized digits (taken from figure 6 of [Cortes and Vapnik, 1995]).

1.2 Support Vector Machines

A support vector machine is a classification algorithm that decides if a given input-vector x should be labeled as being in the target class or not. There are support vector machines for regression, ranking, and multi-class classification; but we will limit ourselves to binary classification. The machine works by checking if the function

$$f(x) := w \cdot \phi(x) + b \tag{1}$$

is greater than zero or not. $\phi()$ is a function from $\mathbb{R}^k$ to a possibly bigger space, and w, b represent the learned separating hyperplane. The set of concepts/models/hypotheses that a SVM considers are half-spaces in the space that is the range of $\phi()$ (in figure 1 $\phi()$ is taken to be the identity, so the concept is a half space in terms of the original parameters x).

One of the clever steps of the method is the use of Mercer's theorem to write $f(x)$ as:

$$f(x) := \sum_{i \in S} a_i K(x_i, x) + b \tag{2}$$

where the index i runs over a subset of the training vectors x_i. The a_i are non-zero scalars and $K(u, v)$ is a positive semi-definite function from $\mathbb{R}^k \times \mathbb{R}^k$ to $\mathbb{R}$ called "the kernel" and is equal to $\phi(u) \cdot \phi(v)$. The set S of training examples that the index i selects is called the set of support vectors (hence the name: support vector machine). The point is: in some cases S can be very much smaller than the number of training examples, and this plus margin[5] help ensure (with high probability over re-draws of the training data) that $f(x)$ is a good approximation of the concept to be learned.

1.3 What is margin?

Informally margin is minimum distance from any datum to the decision surface. We would like to think about margin in a data independent way as being nearly half the minimum width of a moat drawn around the decision surface. In figure 1, $f(x)$ has been written as $w \cdot \phi(x) - b$ and w has been re-scaled to give a moat-width of 2 and a margin of 1. We emphasize that margin is in fact a dimensionless quantity naturally written in terms of the size of the coefficients of w (in this case $2/||w||$) or as a ratio of the moat width over the diameter of the smallest ball containing all of the training data. "Soft-margin" is a generalization of margin where a small proportion of training data is allowed in the moat (subject to penalty).

[5]We will define margin in a bit.

1.4 What can be said about margin, VC dimension, and generalization error?

In [Vapnik, 1998] (and other places) we see very precise quantifiable statements about margin. The math can look involved but there are three fundamental theorems needed to complete the argument (all formulas merely schematic simplifications, or mere cartoons in this section):

1. The VC dimension of the model space can be bounded in terms of an appropriately scaled margin by roughly:
$$VCdimension \leq 1 + 1/margin^2 \tag{3}$$
([Vapnik, 1998] theorem 8.4).
Notice this is independent of the dimensions of both the range and domain of $\phi()$.
[Vapnik, 1998] actually states this as $VCdimension \leq \min(1 + 1/margin^2, \dim(\text{range}(\phi())))$ which is true as the concepts we are working with are essentially hyperplanes passing through the origin in range($\phi()$). Note the radial basis kernel is one of the most popular choices of $\phi()$ in practice and has infinite VC dimension. This emphasizes the importance of the *margin* term over the dim(range($\phi()$)) term for many applications.
2. With probability at least 1-p the expected excess error rate (d) is bounded by a term that decreases in n (the training sample size) similar to:
$$d \leq (VCdimension - \log(p))/n \tag{4}$$
(Vapnik2010 corollary to theorem 5.1).
3. The sample size (n) needed to guarantee with probability at least $1 - p$ that no model in our set of models has an increase in error rate by more than d units when moving from training to application data is about:
$$n \approx (VCdimension - \log(p))/d^2 \tag{5}$$
(standard fact on confidence bounds, for details see the section 2.2 in this write-up).
When there is a perfect hypothesis available (exactly predicts all data) then equation 5 can be strengthened to about:
$$n \approx (VCdimension - \log(p))/d \tag{6}$$
(similar to section 7.4.3 [Mitchell, 1997]).

We have simplified each of the above statements (left out some pre-conditions, constants, sqrts, logs, and so on). But in this literature you have pretty tight bounds with small constants.

1.5 What is typically said about margin, VC dimension, and generalization

Typically the machinery needed to correctly set up and state a correct VC result is considered too much to introduce in the middle of another work. You can not correctly apply such detailed materials by the mere looking for matches of terminology. You must (at least on scratch paper) re-work results in your problem notation to make sure the citation's actual implied pre-conditions and content actually match your application. The issue is: since written mathematics is not fully formalized many references use what are essentially homographs (same words, with different precise formal meaning) and you do not get correct deduction from a mere chaining of textual syllogisms. You have to check that the authors were in fact using terms in the same subtle senses you need.

An example of this is VC dimension itself: the standard definition only applies to families of models (or hypotheses) that are so-called non-partial functions (that is functions whose proper domain includes all possible values of the effective variables). This is why the definitions can be

stated using simple set theory. Yet we see many large-margin results stated as if they were facts about standard VC dimension. But an indicator function or a set with a region of no-answers (the margin region) is essentially a partial function (as it is not applicable on the whole of the original domain). Thus large margin VC dimension results have to be stated in terms of some extension of ideas such as the "working sample" (as in [Vapnik, 1998] section 8.1), or "fat shattering dimension" (as in section 4.3 of [Cristianini and Shawe-Taylor, 2000]).

Most often the benefit of large margin is typically motivated informally by appealing to one or more of the following ideas.

- One can think of large margin as a precaution against small perturbations of the data. The idea is: larger moats are harder to accidentally cross.
- One can think of large margin as a precaution against small perturbations in the estimation of the decision surface (a robustness goal or ϵ-insensitive treatment).
- One can argue the large margin is compatible with the largest measure of decision surfaces yielding the same training labeling and appeal to the principle of indifference.
- One can think of the large margin solution as being an average of many decision surfaces and appeal to ideas of variance reduction or bagging.
- A large margin surface has a harder time encoding tight bends or narrow kinks, so you can think of large margin as a regularization idea.
- In SVM optimization the largest margin solution is unique, possible evidence it is special. This can be thought of as an appeal to symmetry or as a way of avoiding under-determined or ill-posed problems.
- One can think of margin as a pricing trick: placing more emphasis on examples near the decision surface. By complementary slackness examples not at the minimal margin distance do not affect the chosen decision surface.

The above are all laudable goals, but it is unclear how many of them large margin actually simultaneously achieves.

1.6 Margin, VC dimension, and generalization error in the literature

Here we are going to quickly survey what is typically said about margin, VC dimension, and generalization error in the literature (mostly in the secondary literature or textbooks). This is mostly to support our comment that most sources don't have time and space to work through the technical details of VC dimension. This is not to pick on these authors (and we include our on book in the zoo). We are also restricting ourselves to books that mention support vector machines (so avoiding a lot of classic statistics books and business books) as books that don't bring the topic up don't end up influencing popular perception of the topic.

- [Cristianini and Shawe-Taylor, 2000]: has a chapter titled "Generalization Theory" which has good direct proofs of most of the VC and generalization theorems. I suspect the proofs have a couple of typos and have only been able to reproduce similar (not identical results). They also supply a good bibliography.
- [Cortes and Vapnik, 1995]: (a primary source) claims (and refers to) earlier results. Specific definition and discussion of VC dimension and its exact relation to generalization error is deferred to references. We will discuss this more in section 2.
- [Hastie et al., 2009] section 7.9 "Vapnik-Chervonenkis Dimension" defines VC dimension and states the expected excess error rate as a function of sample size. Margin is defined in chapter 12 "Support Vector Machines and Flexible Discriminants", but the discussion is not centered on generalization error.

- [Kuhn and Johnson, 2013]: introduces margin as a desirable SVM efficacy measure without quantifying it. Gives good references.
- [Mitchell, 1997] proves the PAC learning principles for finite concept spaces under the rubric of computational learning theory and sites (without proof) the correct theorem for exact learning of separable concepts in section 7.4.3
- [Murphy, 2012] has sections bounding risk and the large margin principle. Section 6.5.4 "Upper bounding the risk using statistical learning theory" seems to be the primary mention of VC dimension in the book and is limited to observation that there are theorems that use the VC dimension instead of the log of the size of the hypothesis space. Section 14.5.2.2 "The large margin principle" is limited to motivating large margin as being unique and intuitive, without direct reference to the VC dimension.
- [Provost and Fawcett, 2013]: Introduces margin as an anti-noise measure and also works examples in their chapter on over-fitting, demonstrating the robust nature of SVM fits in some situations.
- [Vapnik, 1998]: (a primary source) has all of the proofs. However initially the primary "large margin implies low VC dimension" theorem (theorem 8.4) is only proven in the restricted case where in addition to the training a set of additional points called "the working sample" (additional xs without class labels) is supplied. Theorem 10.3 later in the book states it is derived from theorem 8.4, but 10.3 is stated without the additional "working sample" (x points we will be queried at in addition to the training data). Some variation of theorem 10.3 may or may not follow from theorem 8.4, but that is (contrary to claims) not established in the book.
- [Vapnik, 2010]: defers to [Vapnik, 1998] and other references for most proofs. Theorem 5.1 seems to be referring to theorem 10.3 of [Vapnik, 1998]. Again the "working sample" condition required to meet the pre-conditions of [Vapnik, 1998] theorem 8.4 are not in force. The book discusses the difference between constructive and non-constructive bounds, but it is not clear the distinction is completely enforced.
- [Zumel and Mount, 2014] section 9.4.1 "Understanding support vector machines" states "large margin can actually ensure good behavior on future data (good generalization performance)" without quantifying it and without direct reference to VC dimension (but cites [Cristianini and Shawe-Taylor, 2000]).

I feel that for most of these works it is in fact appropriate (and necessary) to leave out some details. Part of the issue is: unlike colloquial mathematics, algorithms can be quickly and mechanically composed. A practitioner does need to know a lot about the representation issues and consequences of using a support vector machine, but a lot of the proofs and implementation details can (and should) be hidden as the algorithm implementer's responsibilities. Training a SVM (on previously loaded data) can (and should) look as simple as this R listing and the practitioner can move quickly to examining results (like the following recovery of spirals by the a radial basis function kernel SVM).

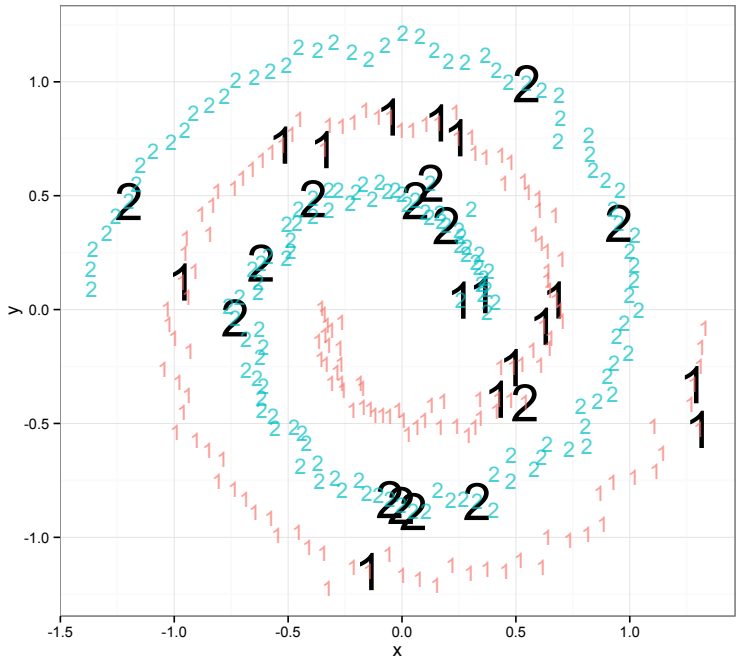

Figure 3: SVM recovering two classes of spiral example (figure 9.11 of Practical Data Science with R).

2 Reasoning about Support Vector Machines

The soft-margin support vector machines currently in use originated in [Cortes and Vapnik, 1995] (see Wikipedia: Support_vector_machine). This paper introduces a lot at once (linear support vectors, convexity of the learning problem, Mercer's theorem, the digits example, and soft-margins for inseparable data). The paper leaves the issue of generalization error (ensuring that f() works well on new data, and not just on training data) to a demonstration and a quick appeal to VC dimension. The exact definition of VC dimension and the exact quantified structure of the generalization bound are left to a reference: [Vapnik, 1982].

As concise as it is this paper states the important intuition very well (formula 38):

$$Pr(\text{test error}) \leq Frequency(\text{training error}) + \text{Confidence Interval} \tag{7}$$

That is: on new data you should expect an error rate that is at least what you saw during training plus another term that looks like a confidence interval. The role of statistical learning theory is to set up provable conditions that the confidence interval represents a number that is with high probability very small (immediately giving you probably approximately correct or PAC-style statements). We are going to explain the applications of this confidence interval and then link it to the VC dimension.

Note: [Cortes and Vapnik, 1995] does make the additional claim (formula 5):

$$E[Pr(\text{error})] \leq E[\text{number of support vectors}]/(\text{number of training vectors}) \tag{8}$$

It looks like [Vapnik, 1998] theorem 10.5 is the follow-up on this statement. I assume this is stated either in the separable case or in terms of excess error, as what is called error is going to zero. In [Vapnik, 1998] there is the additional caveat that we have "expectation taken over both training

and test data", so the error rate is not quite the unknown true error rate or even a pure (not biased by training examples) test-rate estimate. Beyond this and a few non-quantified general comments [Cortes and Vapnik, 1995] doesn't spend much space on VC dimension and generalization error. The core of the paper is the optimization technique and a demonstration.

There are a lot of steps to understanding a support vector machine. But the overall proof technique is in fact quite well motivated, it just requires some strong lemmas. We are going to ignore the very interesting optimization issues (which are the issues most important to actually implementing or justifying the efficient implementation of such a classifier) and work through the issue of generalization error.

2.1 Bounding the confidence interval

There are two primary ways to estimate a bound on the confidence interval we discussed above. Seeing how this bound is derived will also help us understand what is being proven by the bound. The two common derivations are:

1. Appeal to the theory of repeated experiments. If f() is chosen by inspecting 1,000,000 candidate models then under the rubric of multiple comparison issue we need to apply a Bonferroni correction (or appeal to the union bound) and estimate the confidence interval as possibly being as large as 1,000,000 times the expected size of the confidence interval expected from inspecting a single model. This is bad, as picking our support vectors and weights represents a very large (even uncountable) number of candidate models, and we want the confidence interval to represent with high probability a small quantity.
2. Provide a uniform bound for all models. We can show all possible models that we are considering share a common bound on their generalization error. Thus it is irrelevant (with respect to our bound on generalization error) which model we picked and how we pick it.

In the second case the uniform bound argument has the following schematic.

1. Pick a set of possible models: M. In our example we could specify the $\phi()$, which determines the kernel function and pick a radius D and insist that $||a|| \leq D$ where the a_i are support vector weights as in Mercer's theorem (with the convention $a_i = 0$ for i not in our set of support vectors S). We would further specify any other constants that control the SVM implementation (typically they have names C,k, and so-on). Together these conditions give us a sufficiently constrained set of possible models (coefficients not too large, margin not too small, not too many examples in the moat) so that we could expect to be able to prove theorems about this set.
2. Use declared constraints on the set of possible models to prove a uniform detailed confidence bound of the form:

$$\text{For all } f() \in M\text{:}Pr(\text{true error rate of } f()) \leq Frequency(\text{training error of } f()) + C(f) \quad (9)$$

3. Use the detailed structure of $C(f)$ to prove a new confidence shared (or uniform) confidence bound C for all models simultaneously. If (with high probability) no model has a large increase in error rate, then picking a model that is good on training is then a provably good procedure.

The more stringent the constraints in step 1 the easier it is to prove a good common confidence interval in step 2.

2.2 Deriving confidence bounds

Let's work an idealized example. Let M be the set of all models or functions f() we are willing to consider, and suppose M is finite of size m. Suppose we draw n training examples from a fixed (but unknown) joint distribution on effective variables x and class labels. We want to build a confidence interval bounding "d": the excess error rate experienced when we move from the training data to new examples (drawn from the same unknown distribution).

The trick is: we treat $f()$ as being fixed and chosen before we looked at the training data. Then $f()$ has a fixed (unknown) error rate q (on all data) and an observed error frequency "r" on the training data. r is in fact is in fact a simple unbiased noisy measurement of q. We then have the excess error d is: $d = q - r$.

From Hoeffding's inequality we know

$$P[r \leq q - d] \leq exp(-2d^2 n) \tag{10}$$

Or equivalently:

$$P[q - r \geq d] \leq exp(-2d^2 n) \tag{11}$$

Notice $exp(-2d^2 n)$ doesn't contain terms q or r that depend on unknowns or on our choice of $f()$. So this is exactly the kind of uniform bound we are looking for. This is the grace of Hoeffding's inequality, it is weaker than the related Chernoff bound, but that is because it has suppressed the terms you may not know.

We apply this as follows. If we want to ensure that for whichever model we pick from M we have only a p-probability of having an error rate d-higher on new data than on training data, then it is enough to have:

$$exp(-2d^2 n) \leq p/m \tag{12}$$

That is if we make the probability of any one model showing large generalization error as no more than p/m then (by the union bound) the probability of any of m-models showing large generalization error can not be more than $m(p/m) = p$. So if we have at least

$$n \geq (-\log(p) + \log(m))/(2d^2) \tag{13}$$

items of training data then we know with probability at least 1-p that all possible models in M are simultaneously performing near their ideal error rates.

Thus (once we have enough training data) picking a best model by evaluating training performance is in fact a good procedure. This is essentially the argument given in section 7.3.1 of [Mitchell, 1997] (in a chapter named "Computational Learning Theory"). These proofs may seem simple and limited (compared to the more powerful results established using VC dimension), but they are typical of the PAC learning reasoning style and illustrates what one should try to prove using the VC dimension tools. One should also check out [Mitchell, 1997] section 7.3 where stronger bounds of the form $n \geq (-\log(p) + \log(m))/d$ are derived (notice the better dependence on d) for the case where there is a perfect (0-error) concept in the hypothesis space. Oddly enough these stronger bounds don't come from more powerful math (they just use the fact that $(1-\epsilon)^n \leq exp(-\epsilon n)$ instead of having to invoke the Hoeffding inequality) but from the fact that the only way we could fail to pick the correct model is if some wrong model is lucky enough to look perfect on our training data).

We *could* try to apply this same idea to the repeated experiments bound: use the detailed structure of the confidence interval to overcome the effects of the union bound. One might expect to be able to prove an error bound of the form $exp(-2d^2 n) \leq p/Z$ where Z is the number of models considered (say the number of models evaluated during an optimization algorithm such as conjugate gradient optimization). The problem is that the probabilities we are talking about are the

probabilities of picking a bad model under repetition of drawing the training set. So the bound would only be valid when Z is the number of items that could possibly have been inspected by the optimization algorithm under re-draws of the training set. So without more detailed lemmas about the nature of your optimization procedure you don't have a good bound on Z in this case (Z is *not* in fact necessarily bounded by the expected run time of a single run of your optimization procedure).

2.3 VC dimension as a measure of model family complexity

The miracle of VC dimension arguments is that they allow us to correctly derive similar bounds where the $\log(|M|)$ term is essentially replaced with some function of the VC dimension. This allows the uniform bound argument to be applied to large (even uncountable) spaces of models.

In section 7.4.3 [Mitchell, 1997] cites the following (without proof) as a large enough sample size to exactly learn a separable concept with a failure probability no more than p:

$$n \geq (-4\log_2(p/2) + 8VC(M)\log_2(13/d))/d \tag{14}$$

First notice the dependence on d is improved from the original $1/d^2$. This is because we assume we are learning a concept that is in fact separable *and* there is a model with an error rate of zero in set of concepts/hypotheses (a very strong assumption). For finite M we know the VC dimension of M is such that $VC(M) \leq \log(|M|)$, so we should consider this a very successful generalization.

For the soft-margin case (where the data may not be separable) [Vapnik, 2010] (corollary to theorem 5.1) states:

$$d \leq (E/2)(1 + \sqrt{1 + 4r/E}) \tag{15}$$

$$\text{where } E = 4(VC(M)(\log(2n/VC(M)) + 1) - \log(p)/4))/n \tag{16}$$

This has the excess error rate falling roughly linearly in the sample size and the expected near linear dependence on $VC(M)$ and $-\log(p)$.

Notice that VC dimension is superior to minimum description length MDL in that complexity is ascribed to a family of models and not to individual models or individual representations of models. Also VC dimension is superior to thinking of everything in terms of multiple comparisons as it doesn't care how many times you consider the same (or similar models). Though you still must be careful to understand all probabilities are over repetition of data draws and all preparation (not just the SVM optimization step), so the effective VC dimension of your problem may include contributions from variables you have suppressed during pre-processing (under the usual Frequentist counter-factual concern that they could have come in or might come in on a re-run). We now have a couple of examples of how VC dimension is applied to computational learning theory, let's now work more on the formal derivation.

3 Some neglected proofs

The steps usually cited and not actually re-done in secondary sources include:

1. Proof that too many points packed into a sphere can not be convex well separated (section 3.1).[6]
2. Proof that large margin implies low VC dimension (section 3.2).
   To get this we apply the previous result to data points in a fixed sphere and establish no subset of our data above a given size is convex well separated. This in turn implies no subset of our data beyond this size is shatterable and (by definition) gives us a bound on the VC dimension.

[6] Convex well separation will be defined in a bit.

3. Proof the number of ways a finite set of data can be cut by a low VC dimension set of hypotheses is not too large (this called bounding "the growth function", section 3.3).
   This is used to show an arbitrary large concept space behaves a lot like a concept space of as small as $exp(VCdimension)$ hypotheses.
4. Proof that low VC dimension implies good generalization error (section 3.4).
   The argument is essentially a repeat of section 2.2 with some modifications to use $VCdimension$ in place of $\log(m)$ (m having been the total number of hypotheses we are considering when the concept space was finite).

Chaining these steps together gives the complete result: large margin SVMs have good generalization error. We will work through the individual results to try and learn a bit more about their actual character.

3.1 Bounds on packing points in a sphere

A lemma we are going to need is that you can not pack too many points into a sphere while keeping the set of points *well convex separated*. This lemma will control which sets we consider shatterable in section 3.5.2.

Call a finite set of points "well convex separated" if for any split of the set into two sets: the convex hulls of the two sets are at least a given minimum distance apart. If a set of data points is not well convex separated then for some cut of the set into two subsets any separating half-space must pass near some of the data.

The lemma needed is: any set of r-points inside the sphere of radius D in $\mathbb{R}^k$ ($k \geq r-1$) has minimum convex separation no greater than around $cD/\sqrt{r-1}$. It changes nothing to assume $k = r-1$ (as any r-points are in an $r-1$ dimensional flat) and $D = 1$. So we want to show the minimum convex separation of r points inside the unit sphere in $\mathbb{R}^{r-1}$ is no more than $c/\sqrt{r-1}$. [7]

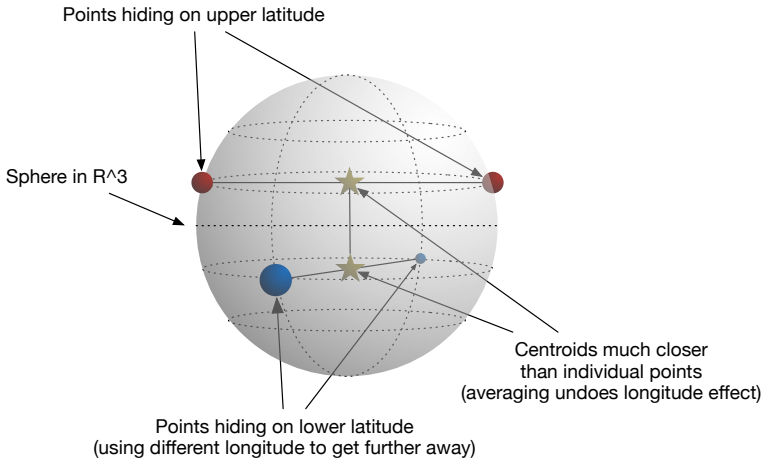

Figure 4: Cartoon example illustrating why centroids of two convex subsets tend to be close.

The very rough idea is to do a lot of work (calculus of variations and game theory) to rigorously establish points on a regular simplex inscribed on the surface of the unit sphere are the extreme case. Then to work out the minimal separation of this case. [Hush and Scovel, 2001] argues this by working with $k = r$ (not $k = r-1$) and using picking vertices of a dimension k-1 simplex as the vectors $E1 \ldots Ek$ (that is Ei is the vector in $\mathbb{R}^k$ such that $Ei_i = 1$ and all other positions are zero,

[7] Note: the individual points are easy to mutually separate, it is the convex hulls of sets that are hard to pull apart.

often called an elementary vector). These points all lie on the surface of a dimension $k-1$ sphere centered around their common average $(1/k,\ldots,1/k)$ (so they are a sphere of radius $\sqrt{(k-1)/k}$, not 1, the original source scales to correct this- but we will ignore it here). Assume k is even and take the first k/2 points as our first set and the rest as our second. Averaging all the points in each of the two sets (separately) gives us two new centroids: (2/k, ...,2/k,0, ...,0) and (0, ...,0,2/k, ...,2/k). The point is: these centroids are in the two convex hulls and the distance between them is $\sqrt{k(2/k)^2} = 2/\sqrt{k}$ which is pretty much what we needed to show.

3.2 Using margin to control VC dimension

By section 3.5.2 we know no more than $cD/\sqrt{r-1}$ points (for some small c) can be packed into a sphere of radius D with convex separation at least u (independent of dimension). Or (solving for r) the largest set of data that is well convex separated is of size no more than $1+(cD/u)^2$.

A set that is not well convex separated can not be shattered by hyperplanes (or half-space cuts, the set of concepts of an SVM) that have all points far away from the decision boundary (as such cuts would establish well convex separation). So the largest sets we consider shatterable by large margin hyperplanes are no larger than $1+(cD/u)^2$. So if we *restricted our hypothesis space to only to large margin concepts* we would have our large margin VC dimension is no more than $1+(cD/u)^2$. The dimensionless quantity u/D is "margin" and we would have VC dimension bounded by $1+1/margin^2$.

Unfortunately there is gap in the above argument. We haven't been precise and how we would restrict a SVM implementation to consider only large margin hypotheses. As we have seen in section 2.2 (and will see again in section 3.4) the proof that low VC dimension implies good generalization error depends not just on the hypothesis that the optimization procedures chooses, but on the entire set of hypotheses it could have potentially considered (even under re-draws of the training data). This is a critical aspect of the proof that allows us to ignore details of the optimization procedure. So to apply such results we need to be a bit more precise about how margin is controlled.

The issue is fixable. Our preferred fix is to use a revised definition of VC dimension such as "fat shattering dimension" that can directly track margin (see [Hush and Scovel, 2001] and [Cristianini and Shawe-Taylor, 2000]). Vapnik's preference was to introduce additional data points (called "the working set") to help witness large margin and allow the original definition of shatterable to be used.[8]

We will discuss our objections to the "working set" argument in section 3.5. For now let us finish the promised lemmas.

3.3 VC dimension and growth functions

My favorite proof that low VC dimension implies slow growth functions is taken from the description of Frankl's shifting technique argument found in Hung Q. Ngo's SUNY at Buffalo, Fall 2010 CSE 711 course notes. We work as follows. Let U be an arbitrary set of possible data points (such as $\mathbb{R}^k$) Let H be a (possibly infinite) set of subsets of U (called our system of hypotheses, models, or concepts). For any finite subset X of U define "the split of X by H" as

$$s(H,X) := \{X \cap g | g \in H\}. \tag{17}$$

[8]Even though the classic definition of shatterable is often stated in terms of indicator functions, its most natural statement is in terms of set systems as in section 3.3. When shatterability is stated in terms of sets is obvious that it is a problem in extremal combinatorics and looks to be in a very general form (with obvious analogies to order ideas, filters, and topology). Fat shattering is necessarily stated in terms of real-valued functions, which obscures the combinatorial nature of the ideas. However, in my opinion, the more powerful results available when using fat shattering as your basis for reasoning far outweigh the loss of generality. This trade-off is common in mathematics: the weaker more detailed definition (fat shattering) allows for stronger results than the stronger more general definition (set shattering).

So $|s(H,X)|$ is the number of different subsets of X the set system H can distinguish.

Call a set X "shattered by H" if $|s(H,X)| = 2^{|X|}$. The empty set is always considered shattered.

Define the VC dimension of H as:

$$VCdimension(H) := \sup\{|X||X \text{ is shattered by } H\}. \tag{18}$$

These definitions are essentially as in section 4.9 of [Vapnik, 1998] (here preferring sets to indicator functions, to emphasize the combinatorial nature of the problem).[9] In this form the problem definitely belongs to the field of extremal combinatorics. This kind of statement looks a lot like Erdos's 1945 generalization of Sperner's theorem with the controlling simplicity condition being "shattered by H" instead of a "no long chains" condition (see [Erdos, 1945]).

What we are trying to prove is called the Sauer–Shelah lemma (also independently discovered by Vapnik-Chervonenkis). The statement is: if the VC dimension of H is the finite integer h then for any finite set X contained in U we have $|s(H,X)| \leq \sum_{i=0,\ldots,h} \binom{|X|}{i}$. (This is particularly useful because $\sum_{i=0,\ldots,h} \binom{|X|}{i} \leq (|X|+1)^h$.)[10]

3.3.1 A toy example

These ideas can be a bit hard to think about on first exposure. An example can help.

For a toy example take:

- Take U as the set of positive integers.
- Take H as the set of concepts "divisible by the prime p" for each prime p. That is H is the set

$$\{\{p, 2p, 3p, 4p, \cdots\}|p \text{ prime}\}. \tag{19}$$

- Take $X = \{2, 3, 4\}$.

Now look at $s(H,X)$, this turns out to be $\{\{2,4\},\{3\},\emptyset\}$. So X was not shattered by H (we failed to generate all $2^3 = 8$ possible subsets).

There is a set of size three that is shattered: $\{5 \times 11 \times 13 \times 17, 3 \times 7 \times 13 \times 17, 2 \times 7 \times 11 \times 17\}$. Any of the possible eight subsets of this three item set can be extracted by checking the element for divisibility by a prime in the range 2 through 19. Such shatterable encodings exist for any finite set set size, so this toy problem has unbounded or infinite VC dimension. We would not be able to prove good generalization error over this set system using VC dimension arguments.

Conversely, if we had taken H as the set of all contiguous intervals of positive integers ($H = \{\{a, a+1, \cdots a+b\}|a \geq 1, b \geq 0\}$) it is easy to show no set of size 3 or more is shattered- so the VC dimension in this case is 2. So the set of intervals is in fact a system of concepts that is easy to learn over.

---

[9] In section 1.2 [Vapnik, 1998] defines the space of hypotheses by a parameter set Λ and an indicator function $g(,)$ where $g(z,\alpha)$ is the prediction the hypothesis named $\alpha \in \Lambda$ makes on the data example z. Indicator functions are stated to be functions whose range is contained in the set $\{0,1\}$.

The introduction of hypothesis/model parameters allows the generalization of allowing for two different hypotheses that assign the exact same values to all possible examples (which is not possible for functions or sets, by axiom of extensionality). For computing one-sided bounds we do not need this generalization and can think of the hypotheses as a set of indicator functions or set of sets.

Section 1.2 defines $\mathcal{Q}(z,\alpha)$ as a specific family of indicator functions *returning the loss* defined as: zero if $g(z,\alpha) = \text{truth}(z)$, and one otherwise. By the time we get to section 4.9 $\mathcal{Q}(z,\alpha)$ seems to be a general set of indicator functions meant to model $\{g(,\alpha)|\alpha \in \Lambda\}$ (and no longer referring to loss).

We only mention this to indicate that, other than some change in notation, we are trying to use the source definitions.

[10] [Vapnik, 1998] section 4.9 actually completely characterizes the growth function, but for clarity we are going to only state and prove the bound actually used to establish our later results.

3.3.2 Proof of the Sauer–Shelah lemma

Proof: Take a finite set X contained in U. Let F = s(H,X). Set G=F. Repeat the following until no progress is possible:

for $x \in X$:
  for $f \in F$:
    if $f \setminus x \notin G$:
      $G = (f \setminus x) \cup (G \setminus f)$

We claim this terminates in a finite number of steps. We also claim:

1. $|G| = |F|$ (this is ensured because our exchange set doesn't change the size of G).
2. The final G is a "lower set" (an order ideal). That is if f in G then all subsets of f are also in G (this is a consequence of our stopping condition).
3. If A is shattered by by G then A is shattered by F. Suppose A is shattered after one repetition elimination of i step (the outer loop), we then argue it must have been shattered before this step. Let G' be the state of G before this processing (and G the state just after). So we are assuming A is shattered by G and need to show it is shattered by G'.
   Let x be the element the outer for-loop selected for elimination. If x not in A the conclusion is obvious (as A can't tell the difference between G' and G). So assume x is in A and let R be an arbitrary subset of A. Then there is g in G such that $R = g \cap A$. We show in R can be produced by A and G', since R was arbitrary this would imply A is shattered by G'. We have two cases to examine.
   - If $x \in R$ then $g \in G'$ (as we were only removing xs not adding them). So R can be built from A and G'.
   - If $x \notin R$, proceed as follows. We know A is shattered by G, so there is a $t \in G$ such that $t \cap A = R \cup \{x\}$. So $t \setminus \{x\}$ is in G' (else t would have been removed from G by our loop) and $(t \setminus \{x\}) \cap A = R$ as needed.

Now notice all sets in G are shattered by G (consequence of G being a lower set). Thus all sets in G are shattered by F (what we just proved). In fact all sets in G are shattered by H. Be the definition of VC dimension we then know all sets in G have size no more than the VC dimension (assumed to be h) and therefore G itself has size no more than $\sum_{i=0,\ldots,h} \binom{|X|}{i}$.

3.4 Low VC dimension implies good generalization error

This proof is adapted from [Cristianini and Shawe-Taylor, 2000] and similar to the argument from [Mitchell, 1997] that we have already shown in section 2.2. The idea is that for a single hypothesis picked before examining the training data: over fitting (performing worse on future application data) is exactly the same event as over-performing or getting lucky on the training data. So all that is left is to work out a way to reason about groups of hypotheses.

Fix p (a probability of failure), d (the excess error we can tolerate), and let n denote sample size. For a given hypothesis H with (unknown) true error rate of q define two events.

- $E1(n, h)$: In a sample size n: hypothesis H makes $(q - d)n$ or fewer errors. Or we have an undetected over-fitting event.
- $E2(n, z)$: In a sequence of size n where each entry is independently chosen to be 1 (representing an error) with probability z: we see $(z - d)n$ or fewer ones.

Both events are defining failures (abstractions of excess generalization errors). We have $P[E1(n, h)] = P[E2(n, q)] \leq max_z P[E2(n, z)]$ (as some value of z equals q). We can bound

$max_z P[E2(n,z)]$ easily. $max_z P[E2(n,z)]$ is no more than the odds of drawing no more than $n(z-d)$ ones in n trials when the true odds are z. This in turn is easy to bound by Hoeffding's inequality as:

$$P[count \leq n(z-d) | odds \geq z] \leq exp(-2nd^2) \tag{20}$$

So for each hypothesis H there is at no more than probability $exp(-2nd^2)$ chance that we see more than d generalization error. With n data points (and assuming finite VC dimension h) we apply the Sauer–Shelah lemma to find there are at most $\sum_{i=0,...,h} \binom{n}{i}$ different equivalence classes of hypotheses possible. We only have to apply the union bound correction proportional to the number of *different* hypotheses we try (repetitions do not cost). We know $\sum_{i=0,...,h} \binom{n}{i} \leq (n+1)^h$, so the probability that even one hypothesis class (and hence at least one hypothesis is bad) is no more than the product of these two quantities. We want our bound to be no larger than our target failure probability p. This gives us:

$$p \geq (n+1)^h exp(-2nd^2) \tag{21}$$

or

$$n \geq (h \log(n+1) - \log(p))/(2d^2) \tag{22}$$

is a large enough sample to ensure with probability at least 1-p all hypotheses only perform at a rate d worse in application than in training (allowing us to with high probability pick an approximately best model by the empirical risk minimization principle). And we have then established that low VC dimensions ensures (with high probability) low generalization error.

Notice this proof only establishes the result. It doesn't introduce any additional constructions (ϵ-nets, level-set approximation of functions) or measures (ϵ-entropy) to attempt to demonstrate *how* VC dimension moderates generalization error.

3.5 Critique of the working set argument

This is where we have a proof gap (in fact even a failure of definitions).

3.5.1 The issue

Many useful SVMs have very high dimensional concept spaces (the range of $\phi()$), so a direct appeal of VC dimension of the set of half-spaces of these spaces (the concepts SVMs work over) will not give good bounds on generalization error.[11] To overcome this we appeal to a concept of large margin as in section 3.1 which can in turn usefully bound VC dimension (and thus allow us to use a VC dimension argument to put a good confidence interval on generalization error).

The problem is: "large margin" as used in theorem 8.4 of [Vapnik, 1998] is defined as a joint property of the hypothesis set, the training data, *and* the set of future data we are going to apply the SVM to (called "the working set"). Yet most applications clearly apply "large margin" as if it was an intrinsic property of the hypothesis set alone. Notice how this is in contrast to VC dimension itself. VC dimension is defined in terms of the hypothesis set alone and not in terms of a single proposed finite data set.[12]

We will work thought the data dependent proof, and point out some of the undesirable consequences of working with a data dependent definition. We also will cite some later reference that try to work around these issues by using slightly modified definitions.

[11] The VC dimension of a the family of half-space concepts is the dimension of the ambient space plus 1.

[12] The definition of VC dimension does make reference to the set U of possible data points, but traditionally this is taken as some set known prior to examining the training data such as $\mathbb{R}^k$, and *not* a finite set of points known *after* the training data is made available. And it certainly does not encode a "working set" or a finite set of points that are the only ones the SVM will ever be run on.

3.5.2 The working set (or data dependent) argument

The definition we gave for VC dimension in section 3.3 is pretty much the standard one. In particular it allows for only complete (not partial classifiers) and is stated in terms of sets (or equivalently indicator functions). It doesn't strictly makes sense to call a hypothesis or classifier "large margin" in this formalism as there is no encoding of distance or of "don't know" in the set system (or system of indicator functions). [13]

Most applications of large-margin learning seem to use theorem 10.3 of [Vapnik, 1998]. This in turn is based on theorem 8.4. The meat of the (data dependent transductive) proof of theorem 8.4 is given in section 8.5 of [Vapnik, 1998] and is outlined as follows.

To control margin Vapnik introduces a new data set called the "complete sample" which is defined as a finite set that is the union of training examples and additional points the SVM will be applied to in the future (called the "working sample").[14]

The dependence of results on the complete sample may come as a surprise to many as theorem 10.3 of [Vapnik, 1998] (which seems to be the theorem used in most references), does not establish the necessary "complete sample" pre-conditions needed to correctly apply theorem 8.4 (which it is turn based upon). So it is hard to say the typical citations of the theorem are in fact completely satisfactory.

10.4.1 Proof of Theorem 10.3

To estimate the VC dimension of the Δ-margin hyperplanes, one has to estimate the maximal number r of vectors that can be separated in all 2^r possible ways by hyperplanes with the margin Δ. This bound was obtained in Theorem 8.4. Therefore the proof of this theorem coincides with the proof of Theorem 8.4 given in Chapter 8, Section 8.5.

Figure 5: Justification of theorem 10.3 [Vapnik, 1998].

3.5.3 Some consequences of using a "working set"

Two of the major components of a SVM are the proofs of generalization (with no assumptions about how optimization is performed) and the efficient implementation of the optimization or learning procedure. Adding the idea of an unlabeled working set has one set of effects on the generalization proofs and another set of effects on the efficient optimization procedure.

- Regarding generalization.
  In our opinion the "working set" is a key concept in the originally presented proofs.

[13] There are, of course, ways to calculate the margin of such a hypothesis *with respect to a given data set*. But this only yields a data-dependent result. I haven't seen the original [Vapnik, 1982] (and it looks like the 2006 2nd edition incorporates a great amount of post-2000 work), but I assume if [Vapnik, 1982] had a non-transductive argument (which would be the stronger argument) we would see this argument in [Vapnik, 1998]. So we will work with the argument given in [Vapnik, 1998]. For an alternative: [Shawe-Taylor et al., 1998] claims a general result using the "fat shattering" idea (see also [Hush and Scovel, 2001] and [Cristianini and Shawe-Taylor, 2000]).

[14] In [Vapnik, 1998] the complete sample is defined in section 8.1 at the top of page 341, and the stated splitting scheme is defined in section 8.5 page 351 right above equation 8.26.

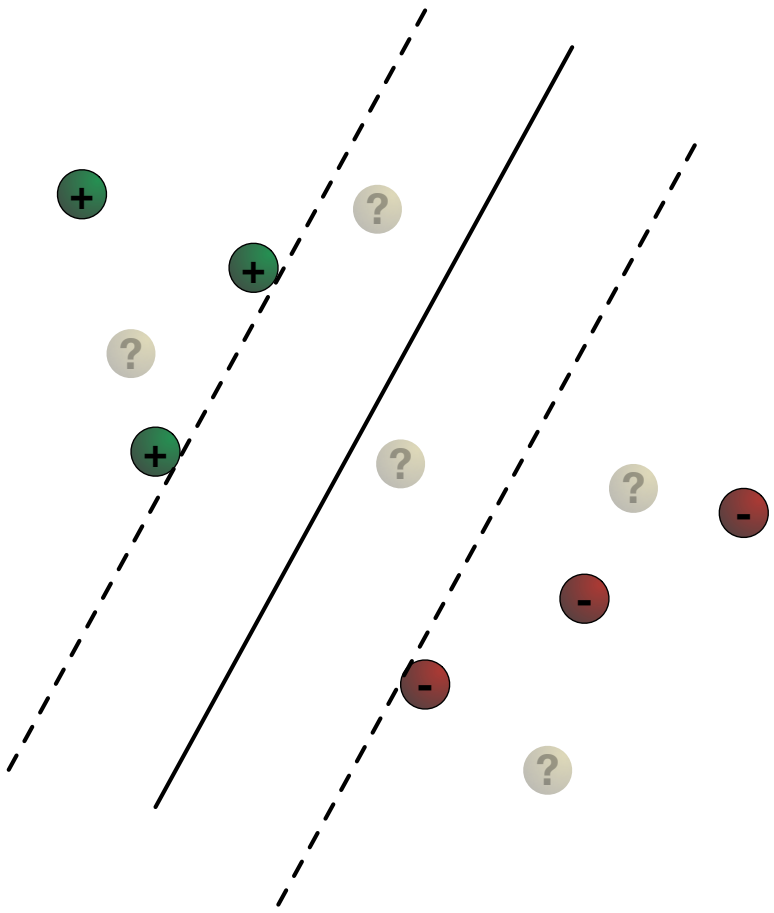

Figure 6: Separator and margin determined by only marked training examples (+ and - points).

The working set determined which models were considered to have large margin, so without explicit access to the working set the learning procedure can not guarantee the selected model is large margin in the sense of theorem 8.4. Or: your SVM's large margin is *not* guaranteed to be the one used in theorem 8.4 if you did not supply at the time of training every point you ever intend to make a prediction for. Even if you knew these points they may force a much narrower margin than the one you would see for looking at only training points. So generalization theorems results are at best in terms of a narrower margin and therefore weaker than you would expect.

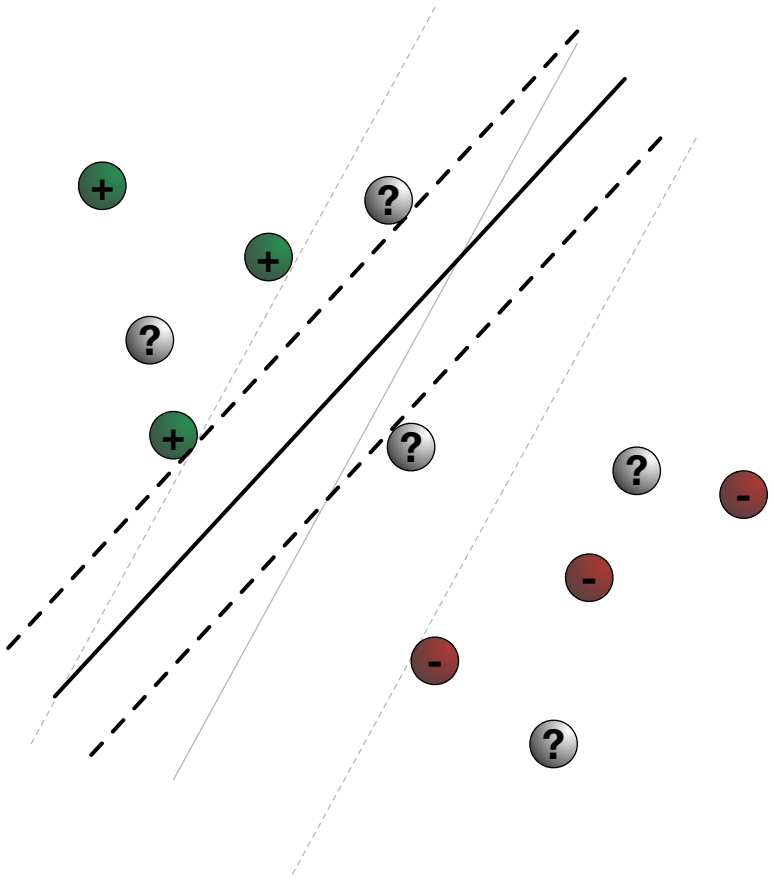

Figure 7: Separator and narrower margin determined by training examples (+ and - points) *and* working set (? points).

Figures 6 and 7 illustrate the issue (for simplicity, in terms of hard-margin; but the effect will be present in the soft margin situation also).
You *can* in fact expect good generalization results in terms of the potentially wider margin determined by the training data alone, but to establish that I would refer to [Shawe-Taylor et al., 1998], [Cristianini and Shawe-Taylor, 2000], or [Hush and Scovel, 2001].

- Regarding efficient optimization.
  The optimization specification of the soft margin optimization problem as stated in formulas 49 and 50 of [Cortes and Vapnik, 1995] (this seems to become equation 10.27 and some predecessors in [Vapnik, 1998]) only refer to labeled data. In fact once we introduce the additional unmarked data we no longer even have a guarantee of a unique maximum margin separator. Figure 8 illustrates the sort of symmetry breaking (introduced by the unmarked or "?" points) causing difficulty.

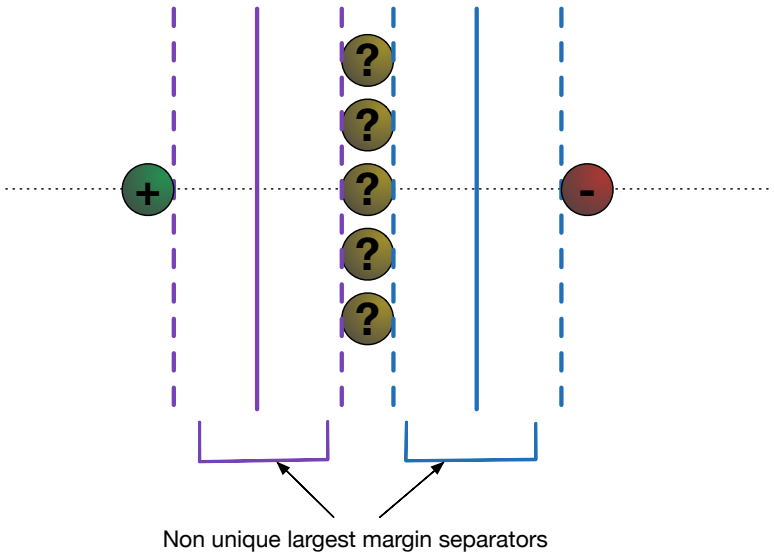

Figure 8: Extreme example of non-unique maximal +/- separators.

Figure 8 illustrates the issue.
The issue is: asking for something to be on a particular side of a surface, a particular distance away *in a pre-specified direction*, or close to a target are all easy to express in a convex program. Asking something to be at least a given distance away *without specifying the direction* is not (in general) easy to encode as a convex or even semi-definite program. Note: the degree of train/work incompatibility (the large stack of unknown working points deep in the separating channel) seen in figure 8 is actually very unlikely as [Vapnik, 1998] chapter 8 *does* assume the training and working data are exchangeable. But the issue still remains: the maximum margin separator is not necessarily unique as we see in figure 9.[15]

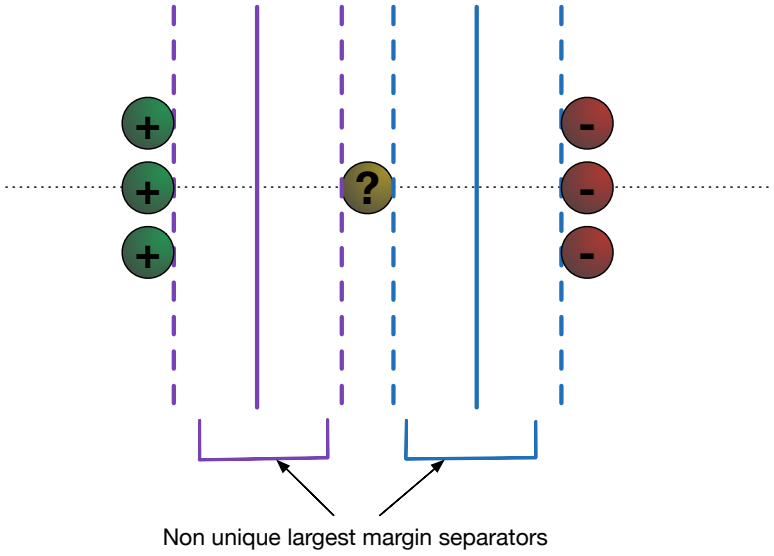

Figure 9: More likely example of non-unique maximal +/- separators.

[15]The problem is obvious in the hard-margin case, and a bit more involved in the soft-margin case as some of the soft-margin budget can be used to essentially move unlabeled data out of the way. But for some values of the soft-margin penalty the soft-margin solution family approximates the hard-margin solution family and the problem re-occurs.

We also claim most of the common SVM *classification* libraries do not in fact accept additional un-labeled data when training classification. Consider the scikit learn and R manual entries for SVM as examples of what data is typically required/accepted for SVM training. So you can not treat these libraries as if they are realizations of theorem 8.4.

To sum up: at some point you have to firmly decide if the additional working set points are really an explicit part of the computational problem or mere notional entities. Both decisions lead to problems, but you can't sit in superposition of both decisions when chaining proofs together.

If the working set points are really given as part of the problem you would have operational issues:

- The SVM optimization problem no longer necessarily has a unique connected component of optimal solutions (and so is no longer convex, and no longer efficiently solved by convex optimizers).
- You would need to actually supply the working set data during training.
- You would have to in principle re-build your SVM classifier every time you were asked to classify a novel example.

If the working set points are only notional then:

- The decision surface and margin returned by the SVM may not be the ones picked in theorem 8.4.
- Any generalization bound claimed is possibly about a *different* decision surface than the one returned by the SVM, so may not apply to your actual classifier.
- Any correct generalization bound is proven is in terms of the theoretical margin of the notional complete sample, so you can't use the optimizer returned margin to bound generalization error.

4 A quick recap of the key ideas

We have spent a lot of time on supporting theorems and examples. For actual understanding it is important to get back to the actual over all plan of proof. The theory behind support vector machines includes both issues of generalization error (margin controlling VC dimension, and that in turn controlling generalization error), and issues of optimization. The fundamental ideas establishing construction and known properties of support vector machines are as follows (grouped roughly by concept):

- Generalize the Guvenko-Cantelli-Kolmogovov theorem to prove sharp bounds on convergence of observed distribution (and summaries) to (unknown) ideal distributions in more situations.
- Introduce Vapnik–Chervonenkis (VC) dimension as the controlling summary in proofs.
- Discover the Sauer–Shelah/Vapnik–Chervonenkis lemma characterizing growth functions of bounded VC dimension set systems.
- Introduce PAC-learning/computational-learning theory (and its emphasis on distribution-free results).
- Relate VC dimension to generalization error.
- Relate margin to VC dimension.
- Relate margin to set size (by correctly bounding the minimum convex separation of r points inside a unit sphere).
- Apply Mercer's theorem on the representation of kernels.
- State large margin separation (and in particular soft margin) as a well-formed convex program.

We list these to explain: even at the length we have attempted here, we clearly must skip a few important topics. We really should appreciate how many good ideas went into inventing and proving

the properties of SVMs. Also, despite the complexity of the ideas there is (in hindsight) a naturalness to it: once you decide margin is your generalization control you would want to maximize margin. And the natural answer to the rigidity of linear machines turns out to be kernel methods. In a sense, despite needing a great number of creative steps, SVMs can be thought of as a consequence of these two concepts.

4.1 A rough chronology

The following is a list of references chosen to place [Cortes and Vapnik, 1995] (where soft margin SVMs are introduced) into context. I have chosen this set to be minimal (so a *lot* is left out) and to try to place them in *rough* chronological order (note: many of these works spent years in preparation and review, so timing and priority are delicate questions). I am deliberately biasing the list to works by Chervonenkis, Cortes, and Vapnik (or referred to by them) to try and indicate the history of their collaborations (and to emphasize statistical works). Even under this bias I felt I had to include more material on kernel representation ([Mercer, 1909]), extremal combinatorics ([Erdos, 1945], [Sauer, 1972], [Shelah, 1972]), PAC-learning/COLT ([Valiant, 1984], [Blumer et al., 1989]), and later proofs of margin/VC results ([Shawe-Taylor et al., 1998], [Hush and Scovel, 2001]). Even after these inclusions I have unfairly omitted a lot the PAC-learning/COLT community's contributions (the field I come from). I suggest [Cristianini and Shawe-Taylor, 2000] as a good source for additional references.

- [Mercer, 1909]: Mercer, J., "Functions of positive and negative type and their connection with the theory of integral equations", Philosophical Transactions of the Royal Society A 209, 1909.
- [Erdos, 1945]: Erdos, P., "On a lemma of Littlewood and Offord", Bulletin of the American Mathematical Society 51: 898–902, 1945.
- [Vapnik and Chervonenkis, 1971]: V. Vapnik and A. Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities" Theory of Probability and its Applications, 16(2):264–280, 1971.
- [Sauer, 1972]: Sauer, N. "On the density of families of sets", Journal of Combinatorial Theory, Series A 13: 145–147, 1972.
- [Shelah, 1972]: Shelah, Saharon "A combinatorial problem; stability and order for models and theories in infinitary languages", Pacific Journal of Mathematics 41: 247–261, 1972.
- [Vapnik, 1982]: V. N. Vapnik, *Estimation of Dependences Based on Empirical Data*, Addendum 1, Springer-Verlag, New York, 1982.
- [Valiant, 1984]: L. Valiant, "A theory of the learnable" Communications of the ACM, 27, 1984.
- [Vapnik and Chervonenkis, 1989]: N. Vapnik, A. Ya. Chervonenkis "The necessary and sufficient conditions for consistency of the method of empirical risk minimization" Yearbook of the Academy of Sciences of the USSR, on Recognition, Classification, and Forecasting, Vol 2, Naukia Moscow, pp. 207-249, 1989.
- [Blumer et al., 1989]: A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth "Learnability and the Vapnik–Chervonenkis dimension" Journal of the ACM, 36(4):929–865, 1989.
- [Cortes and Vapnik, 1995]: Cortes, C.; Vapnik, V. (1995) "Support-vector networks" Machine Learning 20 (3): 273.
- [Vapnik, 1998]: Vladimir N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.
- [Shawe-Taylor et al., 1998]: Shawe-Taylor, J., Bartlett, P. L., Williamson, R. C., & Anthony, M. "Structural risk minimization over data-dependent hierarchies" IEEE Transactions on Information Theory, 44(5), 1926–1940, (1998).
- [Hush and Scovel, 2001]: Don Hush, Clint Scovel, "On the VC Dimension of Bounded Margin Classifiers", Machine Learning, 45, 33–44, 2001.

5 VC dimension in practice

One can ask: what does VC dimension buy you that cross-validation does not?

The ideas are not as different as one would like to think. Proponents often sell VC dimension as having the advantage of being a prior bound. It gives you a proof system that tells you certain classifiers should generalize better than others. However, it doesn't prove that one of the model spaces that has low VC dimension will in fact even achieve good training performance (let alone perform well on later application data). So you are in fact in a situation not that different than test set based validation. You fit your training data and then (in principle) check that the quality of your training fit minus your model family complexity term gives you acceptable performance. This is not entirely different than building a model on training data and then checking that your hold-out performance is acceptable. In one case you know your expected performance loss earlier, but in neither case do you know ahead of time if your final performance will be acceptable.

Also note: the generalization error theorems we have been working with here are all of the form: excess classification error falls as $1/n$, pricing false positives and false negatives the same. So to get the SVM/margin generalization errors to prefer a given sensitivity/specificity or precision/recall trade-off you need one more trick (such as replicating, or re-weighting data).

(I also don't think there is a standard expected generalization error diagnostic (other than number of support vectors used) returned by any of the common SVM libraries sklearn.svm.SVC, sklearn.svm.NuSVC, kernlab, SVMlight, or LIBSVM.)

The thing that VC dimension (and margin in particular) does give you is: a dial you know works. If you over-fit with a random forest you can play with some of the parameters (shallow up the trees, increase the number of trees, and drop some redundant variables) but you don't know that will cure the over-fitting. With a support vector machine, if your SVM doesn't work well on held-out test data you can increase the margin, decrease the number of support vectors and expect to decrease excess error (however this can still fail to fit training data).

6 Conclusion

The proofs relating margin to VC dimension, to generalization error tend to be longer and more involved than most textbook writers have space for. And there is a large amount of additional fascinating material needed to deal properly with the optimization side of support vector machines (not even mentioned here: Lagrangians, Karush–Kuhn–Tucker conditions, convexity, dual forms, properties of kernels, semi-definite programming, the content of Mercer's theorem, vector calculus, optimization, and more). In this write-up I tried to confirm if the standard texts have a clean extractable line of reasoning that takes us from margin to VC dimension to generalization error.

This project started with idea of treating myself to a couple of the original books and researching the original published chain of reasoning, and this certainly colors its point of view.

References

[Blumer et al., 1989] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989). Learnability and the vapnik–chervonenkis dimension. *Journal of the ACM*, 36(4):929–965.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3).

[Cristianini and Shawe-Taylor, 2000] Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge.

[Erdos, 1945] Erdos, P. (1945). On a lemma of littlewood and offord. *Bulletin of the American Mathematical Society*, (51):898–902.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer-Verlag.

[Hush and Scovel, 2001] Hush, D. and Scovel, C. (2001). On the vc dimension of bounded margin classifiers. *Machine Learning*, 45:33–44.

[Kuhn and Johnson, 2013] Kuhn, M. and Johnson, K. (2013). *Applied Predictive Modeling*. Springer-Verlag.

[Mercer, 1909] Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, (209).

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning, A Probabilistic Perspective*. MIT Press.

[Provost and Fawcett, 2013] Provost, F. and Fawcett, T. (2013). *Data Science for Business*. O'Reilly.

[Sauer, 1972] Sauer, N. (1972). On the density of families of sets. *Journal of Combinatorial Theory, Series A*, (13):145–147.

[Shawe-Taylor et al., 1998] Shawe-Taylor, J., Bartlett, P. L., Williamson, R. C., and Anthony, M. (1998). Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940.

[Shelah, 1972] Shelah, S. (1972). A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, (41):247–261.

[Valiant, 1984] Valiant, L. (1984). A theory of the learnable. *Communications of the ACM*, (27).

[Vapnik and Chervonenkis, 1989] Vapnik, N. and Chervonenkis, A. Y. (1989). The necessary and sufficient conditions for consistency of the method of empirical risk minimization. In *Yearbook of the Academy of Sciences of the USSR, on Recognition, Classification, and Forecasting*, volume 2, pages 207–249, Moscow. Naukia.

[Vapnik and Chervonenkis, 1971] Vapnik, V. and Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280.

[Vapnik, 1982] Vapnik, V. N. (1982). *Estimation of Dependences Based on Empirical Data*, volume Addendum 1. Springer-Verlag, New York.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.

[Vapnik, 2010] Vapnik, V. N. (2010). *The Nature of Statistical Learning Theory*. Springer-Verlag, 2nd edition.

[Zumel and Mount, 2014] Zumel, N. and Mount, J. (2014). *Practical Data Science with R*. Manning.

A Soft margin is not as good as hard-margin

Another issue in reading through the published results is how to relate hard-margin results (most of what is proven) to soft-margin classifiers (that which are implemented)

Soft margin results are derived from hard margin results in corollary on page 408 of section 10.2.1 of [Vapnik, 1998]. The content is essentially that a support vector machine with a given margin can be expected to have an error rate on new data (identically distributed as the training data) that is no more than the error Δ-margin error rate seen on training data plus a factor that is roughly $VCdimension/n$ where n is the number of training examples. From the definitions of section 10.2.1 it appears the Δ-margin error rate can be taken to be all errors made on training data including all data that falls too close to the decision surface as also being in error (this appears to be implicit in the definition of a Δ-margin separating hyperplane).

Let's explore a bit how to use such a bound. For our problem set up the very simple artificial one dimensional problem with x_i being the effective variable and y_i being the outcome. Our training data is $y_i = +1$ for x_i uniformly distributed in $(0, 1]$ and $y_i = -1$ for x_i uniformly distributed in $[-1, 0)$. We imagine we draw n samples independently from these distributions (picking from the + and - classes with equal probability).

We are going to make the problem simple and assume we are only estimating a single scalar w and our model is $sign(w \cdot x)$. In this case we see for any $w > 0$ we have a perfect model (i.e. see no errors).

Let's look how the standard soft-margin SVM procedure would pick a w. The standard soft-margin SVM optimization problem for this set of concepts (left/right division of the number line at the origin) is (see [Vapnik, 1998] section 10.2.2):

$$\begin{aligned} &\text{minimize :} \\ &\qquad w \cdot w/2 + C \sum_{i=1}^{n} \xi_i \\ &\text{subject to:} \\ &\qquad \xi_i >= 0 \\ &\qquad y_i(x_i \cdot w) >= 1 - \xi_i \end{aligned}$$

Where C is a parameter usually picked before looking at the data, and often picked as $C = 1$.

The stated conditions insist $|x_i|w >= 1 - \xi_i$ and therefore $\xi_i >= 1 - |x_i|w$. We then expect the fraction of our data that needs a $\xi_i > 0$ corrections to be $min(1, 1/w)$ and the total mass of corrections to be $Cn \int_{x=0}^{min(1,1/w)} (1 - xw)dx$.

From the theory of SVMs we can bound the error seen on new data to be no more than Δ-margin error rate seen during training plus a $(1 + 1/margin^2)/n$ bound on excess generalization error. Our margin is of radius $1/w$ (measured from the center). So we expect our error bound to be no more than the empirical error rate of an appropriate hard-margin classifier that is considered wrong in the margin/moat area (or about a $1/w$ fraction of the time) plus the excess error bound $((1 + 1/margin^2)/n = (1 + w^2)/n)$. So our total error-bound is $1/w + (1 + w^2)/n$. Note we are assuming that the dimension of the concept space the support vector machine is working in is not obviously known (so we can't apply the obvious bound $VCdimension \leq 1$, and instead have to rely on the margin derived portion of the bound).

Let's see what value of C will minimize this error bound estimate.

```
# Python ipython
import sympy
x,w,C,n = sympy.symbols(['x','w','C','n'])

NormTerm = w*w/2
print('NormTerm',NormTerm)
EMarginTerm = C*n*sympy.integrate(1-x*w,(x,0,1/w))
print('EMarginTerm',EMarginTerm)
Objective = NormTerm + EMarginTerm
print('Objective',Objective)
Derivative = sympy.diff(Objective,w)
print('Derivative',Derivative)
OptimalWsGivenC = sympy.solve(Derivative,w)
print('OptimalWsGivenC',OptimalWsGivenC)
BestWGivenC = [ si for si in OptimalWsGivenC if sympy.N(si.subs([(C,1),(n,1)]))][0]
print('BestWGivenC',BestWGivenC)
ErrorBound = 1/w + (1 + w**2)/n
FnOfC = sympy.simplify(ErrorBound.subs(w,BestWGivenC))
print('FnOfC',FnOfC)
D2 = sympy.simplify(sympy.diff(FnOfC,C))
print('D2',D2)
CPicks = sympy.solve(D2,C)
print('CPicks',CPicks)

('NormTerm', w**2/2)
('EMarginTerm', C*n/(2*w))
('Objective', C*n/(2*w) + w**2/2)
('Derivative', -C*n/(2*w**2) + w)
('OptimalWsGivenC', [2**(2/3)*(C*n)**(1/3)/2,
    -2**(2/3)*(C*n)**(1/3)/4 - 2**(2/3)*sqrt(3)*I*(C*n)**(1/3)/4,
    -2**(2/3)*(C*n)**(1/3)/4 + 2**(2/3)*sqrt(3)*I*(C*n)**(1/3)/4])
('BestWGivenC', 2**(2/3)*(C*n)**(1/3)/2)
('FnOfC', 2**(1/3)*C/(2*(C*n)**(1/3)) + 2**(1/3)/(C*n)**(1/3) + 1/n)
('D2', 2**(1/3)*(C - 1)/(3*C*(C*n)**(1/3)))
('CPicks', [1])
```

What we did is look for minima of the optimization objectives (the SVM optimization problem and then the error-bound) by inspecting extreme of these functions by looking at zeros of derivatives. This let us confirm the common choice $C = 1$ is in fact an optimal pick for this toy problem.

Let's substitute $C = 1$ into our expressions and see what the SVM optimization procedure chooses for w, and what generalization bound this gives us.

```
CPick = CPicks[0]
wPick = BestWGivenC.subs(C,CPick)
print('wPick',wPick)
print('ErrorBoundw',sympy.simplify(ErrorBound.subs(w,wPick)))

('wPick', 2**(2/3)*n**(1/3)/2)
('ErrorBoundw', 1/n + 3*2**(1/3)/(2*n**(1/3)))
```

We see with $C = 1$ we get $w = 2^{2/3}n^{1/3}/2$, which is growing as the number of training examples n increases (but not shrinking very fast). Our error bound is then $1/n + 3 \times 2^{1/3}/(2n^{1/3})$, which is slowly shrinking as n grows.

In fact the error bound is shrinking much slower than one might initially expect. The issue is our data doesn't have any intrinsic large margin (a common problem) for real data, but we are having to build our own margin using the soft-margin method.

For a constant margin VC dimension theory tells us excess generalization error shrinks linearly in n, but picking the optimal margin (i.e. shrinking the margin as we get more data) only gives us an overall error-rate bound that is shrinking as $1/n^{1/3}$. Note this is only the bound, as the actual error-rate is zero for any $w > 0$ for this toy problem.

For the fixed $C = 1$ the optimization algorithm is (correctly) picking smaller margins as more data becomes available. This shrinking margin unfortunately obscures some of the sample-size driven improvement in generalization error (the excess generalization error or VC dimension part of the bound). But the trade-off also allows the training algorithm to consider a larger fraction of the training data as being out of the moat (or not in the margin around the decision surface). This is correct behavior on the part of the optimization procedure, and is shrinking the error bound as fast as possible. But "fast" is much slower than any (incorrect) intuition that a fixed C implies a fixed margin as the amount of training data increases. Also note, it is not common to actually explicitly calculate the implied error bound when using SVMs (we just used the obvious structure of our toy problem to allow such a calculation). And further note, we these bounds are approximate as we used simplified forms and not the full equations from the reference (though we are confident we get the overall behavior correct).

This means we expect a halving of the error-bound only each time we multiply our available data by a factor of eight. This is slower than any rule of thumb that states the error-bound shrinks linearly with the amount of training data. Just keep this in mind when deciding how much data you may need for a good SVM result.