Effect of small sample size on text categorization with support vector machines

Paweł Matykiewicz
Biomedical Informatics
Cincinnati Children's Hospital
3333 Burnet Ave
Cincinnat, OH 45220, USA
pawel.matykiewicz@gmail.com

John Pestian
Biomedical Informatics
Cincinnati Children's Hospital
3333 Burnet Ave
Cincinnat, OH 45220, USA
john.pestian@cchmc.org

Abstract

Datasets that answer difficult clinical questions are expensive in part due to the need for medical expertise and patient informed consent. We investigate the effect of small sample size on the performance of a text categorization algorithm. We show how to determine whether the dataset is large enough to train support vector machines. Since it is not possible to cover all aspects of sample size calculation in one manuscript, we focus on how certain types of data relate to certain properties of support vector machines. We show that normal vectors of decision hyperplanes can be used for assessing reliability and internal cross-validation can be used for assessing stability of small sample data.

1 Introduction

Every patient visit generates data, some on paper, some stored in databases as structured form fields, some as free text. Regardless of how they are stored, all such data are to be used strictly for patient care and for billing, not for research. Patient health records are maintained securely according to the provisions of the Health Insurance Portability and Accountability Act (HIPAA). Investigators must obtain informed consent from patients whose data will be used for other purposes. This means defining which data will be used and how they will be used. In addition to writing protocols and obtaining consent from patients, medical experts must either manually codify important information or teach a machine how to do it. All of these labor-intensive tasks are expensive. No one wants to collect more data than is necessary.

Our research focuses on answering difficult neuropsychiatric questions such as, “Who is at higher risk of dying by suicide?” or “Who is a good candidate for epilepsy surgery evaluation?” Large amounts of data that might answer these questions exist in the form of text dictated by clinicians or written by patients and thus unavailable. Parallel to the collection of such data, we explored whether small datasets can be used to build reliable methods of making this information available. Here, we investigate how text classification training size relates to certain aspects of linear support vector machines. We hypothesize that *a sufficiently large training subset will generate stable and reliable performance estimates of a classifier*. On the other hand, *if the dataset is too small, then even small changes to the training size will change the performance of a classifier and manifest unstable and unreliable estimates*. We introduce quantitive definitions for stability and reliability and give empirical evidence on how they work.

2 Background

How much data is needed for reliable and stable analysis? This question has been answered for most univariate problems, and a few solutions exist for multivariate problems, but no widely accepted answer is available for sparse and high-dimensional data. Nonetheless, we will review the few sample size calculation methods that have been used for machine learning.

Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012), pages 193–201,
Montréal, Canada, June 8, 2012.

Hsieh et al. (1998) described a method for calculating the sample size needed for logistic and linear regression models. The multivariate problem was simplified to a series of univariate two-sample t-tests on the input variables. A variance inflation factor was used to correct for the multi-dimensionality which quantifies the severity of multicollinearity in the least squares regression: collinearity deflates and non-collinearity inflates sample size estimation. Computer simulations were done on low-dimensional and continuous data, so it is not known whether the method is applicable to text categorization.

Guyon et al. (1998) addressed the problem of determining what size test set guarantees statistically significant results in a character recognition task, as a function of the expected error rate. This method does not assume which learner will be used. Instead, it requires specific parameters that describe handwriting data collection properties such as between-writers variance and within-writer variance. The downside of this method is that it must assume the worst-case scenario: a large variance in data and a low error rate for the classifier. For this reason larger datasets are recommended.

Dobbin et al. (2008) and Jianhua Hu (2005) focused only on sample size for a classifier that learns from gene expression data. No assumptions were made about the classifier, only about the data structure. All gene expressions were measured on a continuous scale that denotes some luminescence corresponding to the relative abundance of nucleic acid sequences in the target DNA strand. The data, regardless of size, can be qualified using just one parameter, fold change, which measures changes in the expression level of a gene under two different conditions. Furthermore, the fold change can be standardized for compatibility with other biological experiments: with a lower standardized fold change, more samples are needed, and with more genes, more samples are needed. There is a strong assumption about data makeup, but no assumption is made about the classifier. This solution allows for small sample sizes but does not generalize to text classification data.

Way et al. (2010) evaluated the performance of various classifiers and featured a selection technique in the presence of different training sample sizes. Experiments were conducted on synthetic data, with two classes drawn from multivariate Gaussian distributions with unequal means and either equal or unequal covariance matrices. The conclusion was that support vector machines with a radial kernel performed slightly better than the LDA when the training sample size was small. Only certain combinations of feature selection and classification methods work well with small sample sizes. We will use similar assumptions for sparse and high-dimensional data.

Most recently, Juckett (2012) developed a method for determining the number of documents needed for a gold standard corpus. The sample size calculation was based on the concept of capture probabilities. It is defined as the normalized sum of probabilities over all words of interest. For example, if the required capture probability is 0.95 for a set of medical words, when using larger corpora that contain these words, it must first be calculated how many documents are needed to capture the same probability in the target corpus. This method is specific to linguistic research on annotated corpora, where the probabilities of individual words in the sought corpora must match the probabilities of words in the target domain. This method focuses solely on the data structure and does not assume an algorithm or the task that it will serve. The downside is a higher sample size.

When reviewing various methods for sample size calculation, we found that as more assumptions can be made, fewer data are needed for meaningful analysis. Assumptions can be made about data structure and quality, the task the data serve, feature selection, and the classifier. Our approach exploits a scenario where the task, the feature selection, and the classifier are known.

3 Data

We used four data sets to test our hypothesis: versicolor and virginica samples from the Iris dataset (**VV**), newswires about corn and wheat from the ModApte split of the Reuters-21578 dataset (**WCT** and **WCE**), suicide notes reprinted in Shneidman and Farberow (1957) (**SN**), and ubiquitous questionnaire patient interviews (**UQ**). Properties of these data are summarized in Table 1.

The first dataset was created by Anderson (1935) and introduced to the world of statistics by Fisher (1936). Since then it has been used on countless occasions to benchmark machine learning algorithms. Each row of data has four variables to describe the shape of an iris calyx: sepal length, sepal width, petal length, and petal width. The dataset contains 50 measurements for each of three subspecies of the iris flower: setosa, versicolor, and virginica. All measurements of the setosa calyx are separable from the rest of the data and thus were not used in our experiments. Instead, we used data corresponding to versicolor and virginica (**VV**), which is more interesting because of a small class overlap. The noise is introduced mostly by sepal width and sepal length.

The second dataset was created by Lewis and Ringuette (1994) and is the one most commonly used to benchmark text classification algorithms. The collection is composed of 21,578 short news stories from the Reuters news agency. Some stories have manually assigned topics, like "earn," "acq," or "money-fx," and others do not. In order to make the dataset comparable across different uses, a "Modified Apte" ("ModApte") split was proposed by Apté et al. (1994). It has 9,603 training and 3,299 external testing documents, a total of 135 distinct topics, with at least one topic per document. The most frequent topic is "earn," which appears in 3,964 documents. Here, we used only the "wheat" and "corn" categories, which appear 212 and 181 times in the training set along with 71 and 56 cases in the test set. These topics are semantically related, so it is no surprise that 59 documents in the training set and 22 documents in test set have both labels. This gives a total of 335 unique training instances and 105 unique test instances. Interestingly, it is easier to distinguish "corn" news from "not corn just wheat" news than it is to distinguish "wheat" from "not wheat just corn." The latter seems to be a good dataset for benchmarking sample size calculation. We will refer to the "wheat" versus "not wheat" training set as **WCT** and the "wheat" versus "not wheat" external test set as **WCE**.

The third dataset was extracted from the appendix in Shneidman and Farberow (1957). It contains 66 suicide notes (**SN**) organized into two categories: 33 genuine and 33 simulated. The authors of the notes were matched in both groups by gender (male), race (white), religion (Protestant), nationality (native-born U.S. citizens), and age (25-59). Authors of the simulated suicide notes were screened for personality disorders or tendencies toward morbid thoughts that would exclude them from the study. Individuals enrolled in the study were asked to write a suicide note as if they were going to take their own life. Notes were anonymized, digitized, and prepared for text processing (Pestian et al., 2010).

The fourth dataset was collected in a clinical controlled trial at Cincinnati Children's Hospital Medical Center Emergency Department. Sixty patients were enrolled, 30 with suicidal behavior and 30 controls from the orthopedic service. The suicidal behavior group comprised 15 females and 15 males with an average age of ≈ 15.7 years (SD ≈ 1.15). The control group included 15 females and 15 males with an average age of ≈ 14.3 years (SD ≈ 1.21). The interview consisted of five open-ended ubiquitous questions (**UQ**): "Does it hurt emotionally?" "Do you have any fear?" "Are you angry?" "Do you have any secrets?" and "Do you have hope?" The interviews were recorded in an audio format, transcribed by a medical transcriptionist, and prepared for analysis by removing the sections of the interview where the questions were asked. To preserve the **UQ** structure, n-grams from each of the five questions were separated (Pestian et al., 2012).

	VV	**SN**	**UQ**	**WCT**	**WCE**
Samples (m)	100	66	60	335	105
Classes	2	2	2	2	2
Class balance	100%	100%	100%	58%	48%
Min row freq	100	2	2	3	0
Max row freq	100	66	60	335	105
Min cell value	1	0	0	0	0
Max cell value	7.9	102.045	64	117	892
Features (n)	4	60	7,282	7,132	7,132
Sparsity	0%	60%	92.3%	97%	98%

Table 1: Four very different benchmark data: versicolor and virginica (**VV**) from iris data, representing a dense, low-dimensional dataset; suicide notes (**SN**) from *Clues to Suicide* (Shneidman and Farberow, 1957), representing a mildly sparse, high-dimensional dataset; ubiquitous questionnaires, (**UQ**) representing a sparse, extremely high-dimensional dataset; and "wheat" versus "not wheat just corn" (**WCT** and **WCE**) from the "ModApte" split of Reuters-21578 data, representing an unbalanced, extremely sparse, high-dimensional dataset.

4 Methods

Feature extraction. Every text classification algorithm starts with feature engineering. Documents in the **UQ**, **WCT**, and **WCE** sets were represented by a bag-of-n-grams model (Manning and Schuetze, 1999; Manning et al., 2008). Every document was tokenized, and frequencies of unigrams, bigrams, and trigrams were calculated. All digit numbers that appeared in a document were converted to the same token ("NUMB"). Documents become row vectors and n-grams become column vectors in a large sparse matrix. Each n-gram has its own dimension, with the exception of **UQ** data, where n-grams are represented separately for each of the five questions. Neither stemming nor a stop word list were applied to the textual data. Suicide notes (**SN**) were not represented by n-grams. In previous studies, we found that the structure of the note and its emotional content are indicative of suicidality, not its semantic content. Hence, the **SN** dataset is represented by the frequency of 23 emotions assigned by mental health professionals, the frequency of 34 parts of speech, and by three readability scores: Flesch, Fog, and Kincaid.

Feature weighting. Term weighting was chosen *ad hoc*. **UQ**, **WCT**, and **WCE** had a logarithmic term frequency (log-tf) as local weighting and an inverse document frequency (idf) as global weighting but were derived only from the training data (Salton and Buckley, 1988; Nakov et al., 2001).

Feature selection. To speed up calculations, the least frequent features were removed from the **SN**, **UQ**, **WCT**, and **WCE** datasets (see minimum row frequency in Table 1). Further optimization of the feature space was done using an information gain filter (Guyon and Elisseeff, 2003; Yang and Pedersen, 1997). Depending on the experiment, some of the features with the lowest information gain were removed. For example, $IG = 0.4$ means that 40% of the features, those with a higher information gain, were kept, and the other 60%, those with a lower information gain, were removed. Lastly, all row vectors in **UQ**, **WCT**, and **WCE** were normalized to unit length (Joachims, 1998).

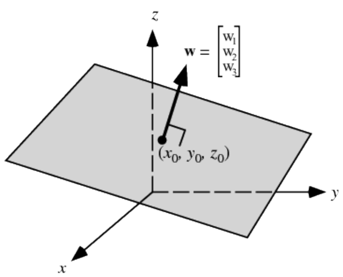
z
$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$
(x_0, y_0, z_0)
y
x

Figure 1: Normal vector $\mathbf{w}$ of a hyperplane.

Learning algorithm. We used linear support vector machines (**SVM**) to learn from the data. Support vector machines are described in great detail in Schlkopf and Smola (2001). We will focus on just two aspects: properties of the normal vector of decision hyperplane (see Figure 1) and internal cross-validation (see Figure 2). **SVM** is in essence a simple linear classifier:

$$f(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) \tag{1}$$

where $\mathbf{x}$ is an input vector that needs to be classified, $\langle \cdot, \cdot \rangle$ is the inner product, $\mathbf{w}$ is a weight vector with the same dimensionality as $\mathbf{x}$, and b is a scalar. The function f outputs $+1$ if $\mathbf{x}$ belongs to the first class or -1 if $\mathbf{x}$ belongs to the second class. **SVM** differs from other linear classifiers on how $\mathbf{w}$ is computed. Contrary to other classifiers, it does not solve $\mathbf{w}$ directly. Instead, it uses convex optimization to find vectors from the training set that can be used for creating the largest margin between training examples from the first and second class. Hence, the solution to $\mathbf{w}$ is in the form of the linear combination of coefficients and training vectors:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{2}$$

where m is the number of training vectors, $\alpha_i \geq 0$ are Lagrange multipliers, $y_i \in \{-1, 1\}$ are numerical codes for class labels, and $\mathbf{x}_i$ are training row vectors. Vector $\mathbf{w}$ is perpendicular to the decision boundary, and its proper name in the context of **SVM** is the normal vector of decision hyperplane[1] (see Figure 1). One of the properties of **SVM** is that outlying training vectors are not used in $\mathbf{w}$. These vectors have the corresponding coefficient $\alpha_i = 0$. In fact, these vectors can be removed from the training set and the convex optimization procedure will

[1] If *R* with **SVM** from the *e1071* package is used, the command to obtain the normal vector is `w = c(t(model$coefs)%*%model$SV)`.

result in exactly the same solution. We can use this property to probe how reliable training data are for the classification task. If we have enough data that we can randomly remove some, what is left will result in $\mathbf{w}^* \approx \mathbf{w}$. On the other hand, if we do not have enough data, then random removal of training data will result in a very different equation, because the decision boundary changes and $\mathbf{w}^* \neq \mathbf{w}$.

Reliability of performance. The relationship between $\mathbf{w}^*$ and $\mathbf{w}$ can be measured. We introduce the **SVM** reliability index (**SRI**):

$$\begin{aligned} \mathrm{SRI}(\mathbf{w}^*, \mathbf{w}) &= |r(\mathbf{w}^*, \mathbf{w})| \qquad (3) \\ &= \frac{|\sum_{i=1}^{n}(w_i^* - \overline{\mathbf{w}}^*)(w_i - \overline{\mathbf{w}})|}{\sqrt{\sum_{i=1}^{n}(w_i^* - \overline{\mathbf{w}}^*)^2}\sqrt{\sum_{i=1}^{n}(w_i - \overline{\mathbf{w}})^2}} \end{aligned}$$

which is the absolute value of the Pearson product-moment correlation coefficient between convex optimization solution $\mathbf{w}^*$ corresponding to a training subset and $\mathbf{w}$ corresponding to the full dataset[2]. Pearson's correlation coefficient discovers linear dependency between two normally distributed random variables and has its domain on a continuous segment between -1 and $+1$. In our case, we are looking for a strong linear dependency between constituents of the training weight vector w_i^* and constituents of the full dataset weight vector w_i. Some numerical implementations of **SVM** cause the output values for the class labels to switch. We corrected for this effect by applying absolute value to the Pearson's coefficient, resulting in $\mathrm{SRI} \in [0, 1]$. We did not have a formal proof on how **SRI** relates to **SVM** performance. Instead, we showed empirical evidence for the relationship based on a few small benchmark data. **Stability of performance.** **SVM** generalization performance is usually measured using cross-validation accuracy. In particular, we use balanced accuracy because it gives better evidence for a drop in performance when solving unbalanced problems. Following Guyon and Elisseeff (2003) and many others, we divided the data into three sets: test, training, and validation. Mean test balanced accuracy $\overline{a}^T$ is estimated using stratified Monte Carlo cross-validation (**MCCV**), where the proportion of the training set to the test set is varied between 0.06 and 0.99. Mean validation balanced accuracy $\overline{a}^V$ (**MVA**) is estimated using K-fold cross-validation (also known as internal cross-validation), where $K = \frac{m}{2}$ and m is the number of training cases. In the case of the "wheat" versus "not wheat just corn" dataset, we have, in addition, the external validation set **WCE** and corresponding mean external balanced accuracy $\overline{a}^E$. Correct estimation of the learner's generalization performance should result in all three accuracies being equal: $\overline{a}^T \approx \overline{a}^V \approx \overline{a}^E$. Furthermore, we want all three accuracies to be the same regardless of the amount of data. If we have enough data that we can randomly remove some, what is left will result in $\overline{a}^{V^*} \approx \overline{a}^{V^{**}}$. On the other hand, if we do not have enough data, then random removal of training data will result in very different accuracy estimations: $\overline{a}^{V^*} \neq \overline{a}^{V^{**}}$.

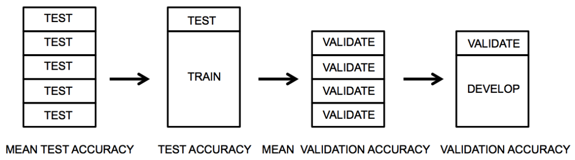

Figure 2: Estimation and resampling: mean test balanced accuracy and mean validation balanced accuracy should match. To prevent overfitting, tuning machine learning should be guided by mean validation accuracy and confirmed by mean test accuracy. This procedure requires the "develop" set to be large enough to give reliable and stable estimates.

Sample size calculation. We do not have a good way of predicting how much data will be needed to solve a problem with a small p-value, but this is a matter of convenience. Rather than looking to the future, we can simply ask if what we have now is enough. *If we can build a classifier that gives reliable and stable estimates of performance, we can stop collecting data.* Reliability is measured by **SRI**, while stability is measured by **MVA**, not as a single value but merely as a function of the training size:

$$SRI(t) = |r(\mathbf{w}^{tm}, \mathbf{w}^{m})| \quad \text{and} \qquad (4)$$

$$a^T(t) = a^{T^{tm}} \qquad (5)$$

where t is a proportion of the training data, $t \in (0, 1)$, m is size of the full dataset, and tm is the actual number of training instances. To quantify the

[2] We experimented with Pearson's correlation, Spearman's correlation, one-way intraclass correlation, Cosine correlation, Cronbach's coefficient, and Krippendorff's coefficients and found that Pearson's correlation coefficient works well with both low-dimensional and high-dimensional spaces.

ability of the dataset to produce classification models with reliable and stable performance estimates, we need two more measures: sample dispersion of **SRI** and sample dispersion of **MVA**:

$$c_{SRI}(t \geq p) = \frac{s_{SRI(t \geq p)}}{SRI(t \geq p)} \quad \text{and} \tag{6}$$

$$c_{MVA}(t \geq p) = \frac{s_{a^T(t \geq p)}}{a^T(t \geq p)} \tag{7}$$

defined as the coefficient of variation of all **SRI** or **MVA** measurements for training data sizes greater than $p\dot{m}$. For example, we want to know if our 10-fold cross-validation (**CV**) for a dataset that has 400 training samples is reliable and stable. 10-fold CV is 0.9 of training data, so we need to measure **SRI** and **MVA** for different proportions of training data, $t = \{0.90, 0.91, \ldots, 0.99\}$, and then calculate dispersion for $c_{SRI}(t \geq 0.9)$ and $c_{MVA}(t \geq 0.9)$. Numerical calculations will give us sense of good and bad dispersion across different datasets.

5 Results

Do I have enough data? The first set of experiments was done with untuned algorithms. We set the **SVM** parameter to $C = 1$ and did not use any feature selection. Figure 3 shows four examples of how **SVM** performance depends on the training set size. The performance was measured using mean test balanced accuracy, **MVA**, and **SRI**. Numerical calculations showed that **VV** needs at least 30 randomly selected training examples to produce reliable and stable results with high accuracy. $c_{SRI}(t \geq 0.75)$ is 0.005 and $c_{MVA}(t \geq 0.75)$ is 0.016. **SN** was not encouraging regarding the estimated accuracy; **SRI** dropped, suggesting that the **SVM** decision hyperplanes are unreliable. Mental health professionals can distinguish between genuine and simulated notes about 63% of time. Machine learning does it correctly about 73% of time if text structure and emotional content are used. Even so, the sample size calculation yields high dispersion ($c_{SRI}(t \geq 0.75) = 0.134$ and $c_{MVA}(t \geq 0.75) = 0.082$). **UQ** is small and high-dimensional, and yet the results were reliable and stable ($c_{SRI}(t \geq 0.75) = 0.015$ and $c_{MVA}(t \geq 0.75) = 0.023$). Patients enrolled in the **UQ** study also received the Suicide Ideation Questionnaire (Raynolds, 1987) and the Columbia-Suicide Severity Rating Scale (Posner et al., 2011). We found that **UQ** was no different from the structured questionnaires. **UQ** detects suicidality mostly by emotional pain and hopelessness, which were mildly present in four control patients. Other instruments returned errors because the same few teenagers reported risky behavior and morbid thoughts. **WCT** produced reliable and stable accuracy estimates, but no large amounts of data could be removed ($c_{SRI}(t \geq 0.75) = 0.010$ and $c_{MVA}(t \geq 0.75) = 0.053$). It seems that **WCE** is somehow different from **WCT**, or it might be a case of overfitting, which causes the mean test accuracy to diverge from **MVA** as the training dataset gets smaller. **Algorithm tuning.** No results should be regarded as satisfactory until a thorough parameter space search has been completed. Each step of a text classification algorithm can be improved. To attempt a complete description of the dependency of a minimal viable sample size on text classification would be both impossible and futile, since new methods are discovered every day. However, to start somewhere, we focused only on the feature selection and **SVM** parameter C [3]. Feature selection removes noise from data. Parameter C informs the convex optimization process about the expected noise level. If both parameters are set correctly, we should see an improvement in the reliability and stability of the results. There are several methods for tuning **SVM**; the most commonly used but computationally expensive is internal cross-validation (Duan et al., 2003; Chapelle et al., 2002). Figure 5 shows the results of the parameter tuning procedure. **VV** and **SN** are not extremely high-dimensional, so we tuned just parameter C. **MVA** maxima were found at $C = 0.45$ with **VV**, $C = 0.05$ with **SN**, $C = 0.4$ and $IG = 0.1584$ with **UQ**, and $C = 2.5$ and $IG = 0.8020$ with **WCT**. **Do I have enough data after algorithm tuning?** Internal cross-validation (**MVA**) did not improve dispersion universally (see Table 2). **VV** improved on reliability but not stability. **SN** scored much better on both measures, but we do not yet know what the cutoff for having a low enough dispersion is. **UQ** did worse on all measures after tuning. **WCT** improved greatly on mean

[3] Please note that most **SVM** implementations do not allow for simultaneous feature selection and internal cross-validation.

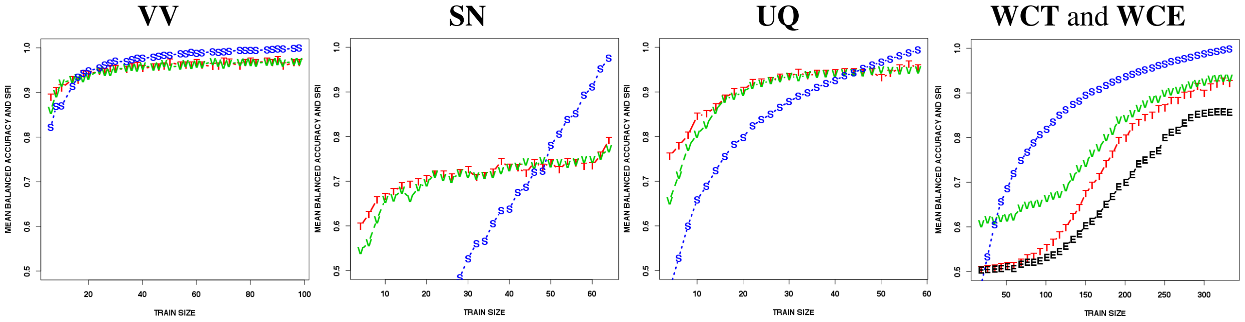

Figure 3: **SRI** index (**S**), **MVA** accuracy (**V**) and mean test accuracy (**T**) averaged over 120 repetitions and different training data sizes. Linear **SVM** with $C = 1$ and no feature selection. **VV** ($c_{SRI}(t \geq 0.75) = 0.005$ and $c_{MVA}(t \geq 0.75) = 0.016$), **UQ** ($c_{SRI}(t \geq 0.75) = 0.015$ and $c_{MVA}(t \geq 0.75) = 0.023$), and **WCT** ($c_{SRI}(t \geq 0.75) = 0.010$ and $c_{MVA}(t \geq 0.75) = 0.053$) gave stable and reliable estimates, but **SN** did not ($c_{SRI}(t \geq 0.75) = 0.134$ and $c_{MVA}(t \geq 0.75) = 0.082$).

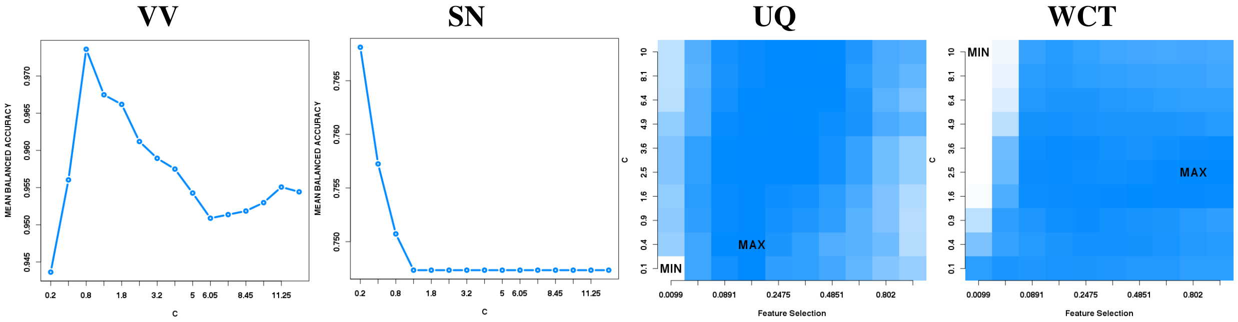

Figure 4: **MVA** (internal cross-validation) parameter tuning results. Maxima were found at $C = 0.45$ with **VV**, $C = 0.05$ with **SN**, $C = 0.4$ and $IG = 0.1584$ with **UQ**, and $C = 2.5$ and $IG = 0.8020$ with **WCT**.

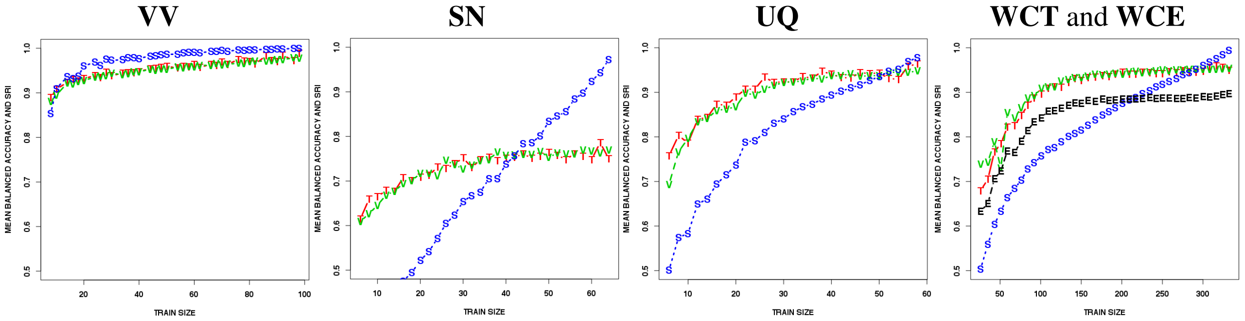

Figure 5: **SRI** index (**S**), **MVA** accuracy (**V**), and mean test accuracy (**T**) averaged over 60 repetitions and different training data sizes. Tuned classification algorithms: **VV** with $C = 0.45$ and no feature selection, **SN** with $C = 0.05$ and no feature selection, **UQ** with $C = 0.4$ and $IG = 0.1584$, and **WCT** with $C = 2.5$ and $IG = 0.8020$. Stability and reliability: **VV** had $c_{SRI}(t \geq 0.75) = 0.003$ and $c_{MVA}(t \geq 0.75) = 0.018$), **SN** had $c_{SRI}(t \geq 0.75) = 0.085$ and $c_{MVA}(t \geq 0.75) = 0.075$, **UQ** had $c_{SRI}(t \geq 0.75) = 0.025$ and $c_{MVA}(t \geq 0.75) = 0.024$, and **WCT** had $c_{SRI}(t \geq 0.75) = 0.025$ and $c_{MVA}(t \geq 0.75) = 0.011$.

test accuracy, mean external validation, and stability dispersion (see Figure 5). It would be interesting to see if improvement on both reliability dispersion and stability dispersion would bring mean test accuracy and mean external validation even closer together.

	$\overline{a^T(t \geq 0.75)}$	$c_{SRI}(t \geq 0.75)$	$c_{MVA}(t \geq 0.75)$
VV no tuning	0.965	0.005	**0.016**
SN no tuning	0.744	0.134	0.082
UQ no tuning	**0.946**	**0.015**	**0.023**
WCT no tuning	0.862	**0.010**	0.053
VV with tuning	**0.970**	**0.003**	0.018
SN with tuning	**0.755**	**0.085**	**0.075**
UQ with tuning	0.941	0.025	0.024
WCT with tuning	**0.946**	0.025	**0.011**

Table 2: Sample size calculation before and after tuning with internal cross-validation (**MVA**). Even though mean test accuracy ($\overline{a^T(t \geq 0.75)}$) improved for **VV**, **SN**, and **WCT**, reliability and stability did not improve universally. Internal cross-validation alone might not be adequate for tuning classification algorithms for all data.

6 Discussion

Sample size calculation data for a competition and for problem-solving. In general, there might be two conflicting objectives when calculating whether what we have collected is a large enough dataset. If the objective is to have a shared task with many participants and, thus, many unknowns, the best course of action is to assume the weakest classifier: unigrams with no feature weighting or selection trained using the simplest logistic regression. On the other hand, if the problem is to be solved with only one classifier and the least amount of data, then the strongest assumptions about the data and the algorithm are required.

The fallacy of untuned algorithms. After years of working with classification algorithms to solve difficult patient care problems, we have found that a large amount of data is not needed; usually samples measured in the hundreds will suffice, but this is only possible when a thorough parameter space search is conducted. It seems that reliability and stability dispersions are good measures of how well the algorithm is tuned to the data without overfitting. Moreover, we now have a new direction for thinking about optimizing classification algorithms: instead of focusing solely on accuracy, we can also measure the dispersion and see whether this is a better indicator of what would happen with unevaluated data. There is a great deal of data available, but very little that can be used for training.

What to measure? VC-bound, span-bound, accuracy, F_1, reliability, and stability dispersions are just a few examples of indicators of how well our models fit. What we have outlined here is how one of the many properties of **SVM**, the property of the normal vector, can be used to obtain insights into data. Normal vectors are constructed using Lagrangian multipliers and support vectors; accuracy is constructed using a sign function on decision values. It is feasible that other parts of **SVM** may be more suited to algorithm tuning and calculation of minimum viable training size.

7 Conclusion

Power and sample size calculations are very important in any domain that requires extensive expertise. We do not want to collect more data than necessary. There is, however, a scarcity of research in sample size calculation for machine learning. Nonetheless, the existing results are consistent: the more that can be assumed about the data, the problem and the algorithm, the fewer data are needed.

We proposed two independent measures for evaluating whether available datasets are sufficiently large: reliability and stability dispersions. Reliability dispersion measures indirectly whether the decision hyperplane is always similar and how much it varies, while stability dispersion measures how well we are generalizing and how much variability there is. If the sample size is large enough, we should always get the same decision hyperplane with the same generalization accuracy.

With little empirical evidence, we can conclude that classifier performance measured by just a single K in a cross-validation test is not sufficient. K must be be varied, and other measures must be present, such as the SVM reliability index, that support or contradict the generalization accuracy estimates. We suggest that other measures for sample size calculation and algorithm tuning may exist and there is still much to be learned about the mechanics of support vector machines.

References

Edgar Anderson. 1935. The irises of the gaspe peninsula. *Bulletin of the American Iris Society*, 59:2–5.

Chidanand Apté, Fred Damerau, and Sholom M. Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Trans. Inf. Syst.*, 12(3):233–251, July.

Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. 2002. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159.

Kevin K. Dobbin, Yingdong Zhao, and Richard M. Simon. 2008. How large a training set is needed to develop a classifier for microarray data? *Clinical cancer research : an official journal of the American Association for Cancer Research*, 14(1):108–114, January.

Kaibo Duan, S. Sathiya Keerthi, and Aun Neow Poo. 2003. Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51:41–59.

Ronald A. Fisher. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188.

Isabelle Guyon and Andre Elisseeff. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March.

Isabelle Guyon, John Makhoul, Richard Schwartz, and Vladimir Vapnik. 1998. What size test set gives good error rate estimates? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(1):52–64, January.

Fushing Y. Hsieh, Daniel A. Bloch, and Michael D. Larsen. 1998. A simple method of sample size calculation for linear and logistic regression. *Statistics in Medicine*, 17(14):1623–1634, December.

Fred A. Wright Jianhua Hu, Fei Zou. 2005. Practical fdr-based sample size calculations in microarray experiments. *Bioinformatics*, 21(15):3264–3272, August.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In Claire Ndellec and Cline Rouveirol, editors, *Machine Learning: ECML-98*, volume 1398, pages 137–142. Springer-Verlag, Berlin/Heidelberg.

David Juckett. 2012. A method for determining the number of documents needed for a gold standard corpus. *Journal of Biomedical Informatics*, page In Press, January.

David D. Lewis and Marc Ringuette. 1994. A comparison of two learning algorithms for text categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93.

Christopher D. Manning and Hinrich Schuetze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1 edition, June.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July.

Preslav Nakov, Antonia Popova, and Plamen Mateev. 2001. Weight functions impact on lsa performance. In *EuroConference RANLP'2001 (Recent Advances in NLP)*, pages 187–193.

John Pestian, Henry Nasrallah, Pawel Matykiewicz, Aurora Bennett, and Antoon Leenaars. 2010. Suicide note classification using natural language processing: A content analysis. *Biomedical Informatics Insights*, pages 19–28, August.

John Pestian, Jacqueline Grupp-Phelan, Pawel Matkiewicz, Linda Richey, Gabriel Meyers, Christina M. Canter, and Michael Sorter. 2012. Suicidal thought markers: A controlled trail examining the language of suicidal adolescents. *To Be Determined*, In Preparation.

Kelly Posner, Gregory K. Brown, Barbara Stanley, David A. Brent, Kseniya V. Yershova, Maria A. Oquendo, Glenn W. Currier, Glenn A. Melvin, Laurence Greenhill, Sa Shen, and J. John Mann. 2011. The ColumbiaSuicide severity rating scale: Initial validity and internal consistency findings from three multisite studies with adolescents and adults. *The American Journal of Psychiatry*, 168(12):1266–1277, December.

William M. Raynolds, 1987. *Suicidal Ideation Questionnaire - Junior*. Odessa, FL: Psychological Assessment Resources.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Bernhard Schlkopf and Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 1st edition, December.

Edwin S. Shneidman and Norman Farberow. 1957. *Clues to Suicide*. McGraw Hill Paperbacks.

Ted W. Way, Berkman Sahiner, Lubomir M. Hadjiiski, and Heang-Ping Chan. 2010. Effect of finite sample size on feature selection and classification: a simulation study. *Medical Physics*, 37(2):907–920, February.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 412–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.