On Learning Multicategory Classification with Sample Queries

Joel Ratsaby

Department of Computer Science, University College London, Gower Street, London WC1E 6BT, U.K.

Abstract

Consider the pattern recognition problem of learning multi-category classification from a labeled sample, for instance, the problem of learning character recognition where a category corresponds to an alphanumeric letter. The classical theory of pattern recognition assumes labeled examples appear according to the unknown underlying pattern-class conditional probability distributions where the pattern classes are picked randomly according to their *a priori* probabilities. In this paper we pose the following question: Can the learning accuracy be improved if labeled examples are independently randomly drawn according to the underlying class conditional probability distributions but the pattern classes are chosen *not* necessarily according to their *a priori* probabilities ? We answer this in the affirmative by showing that there exists a tuning of the subsample proportions which minimizes a loss criterion. The tuning is relative to the intrinsic complexity of the Bayes-classifier. As this complexity depends on the underlying probability distributions which are assumed to be unknown, we provide an algorithm which learns the proportions in an on-line manner utilizing sample querying which asymptotically minimizes the criterion. In practice, this algorithm may be used to boost the performance of existing learning classification algorithms by apportioning better subsample proportions.

Key words: Multicategory classification, On-line learning algorithm, Pattern recognition, Structural Risk Minimization, Stochastic gradient descent learning

1 Statement of the Problem

The general problem of learning pattern classification has been studied extensively in the literature of classical pattern recognition cf. Duda et. al. [2001],

Email address: J.Ratsaby@cs.ucl.ac.uk (Joel Ratsaby).
URL: http://www.cs.ucl.ac.uk/staff/J.Ratsaby/ (Joel Ratsaby).

Article published in Information and Computation (2003) Vol.185 , No.2 , pp.298–327

Fukunaga [1972], Vapnik [1982], Devroye et. al. [1996], under statistical decision theory and more recently in machine learning theory under the Probably Approximately Correct (PAC) model of Valiant [1984], Blumer et. al. [1989]. In the classical framework the problem is posed as follows: We are given M distinct pattern classes each with a class conditional probability densities $f_i(x)$, $1 \leq i \leq M$, $x \in \mathbb{R}^d$, and *a priori* probabilities p_i, $1 \leq i \leq M$. The functions $f_i(x)$, $1 \leq i \leq M$, are assumed to be unknown while the p_i are assumed to be known or unknown depending on the particular setting. The learner observes randomly drawn i.i.d. examples each consisting of a pair of a feature vector $x \in \mathbb{R}^d$ and a label $y \in \{1, 2, \ldots, M\}$, which are obtained by first drawing y from $\{1, \ldots, M\}$ according to a discrete probability distribution $\{p_1, \ldots, p_M\}$ and then drawing x according to the selected probability density $f_y(x)$.

Denoting by $c(x)$ a classifier which represents a mapping $c : \mathbb{R}^d \to \{1, 2, \ldots, M\}$ then the *misclassification error* of c is defined as the probability of misclassification of a randomly drawn x with respect to the underlying mixture probability density function $f(x) = \sum_{i=1}^{M} p_i f_i(x)$. This misclassification error is commonly represented as the expected 0/1-loss, or simply as the *loss*

$$L(c) = \mathrm{E}1_{\{c(x) \neq y(x)\}}$$

of c where expectation is taken with respect to $f(x)$ and $y(x)$ denotes the true label (or class origin) of the feature vector x. Note, in general $y(x)$ is a random variable depending on x and only in the case of $f_i(x)$ having non-overlapping probability 1 supports then $y(x)$ is a deterministic function [1].

The classical problem of pattern recognition is to learn, based on a finite randomly drawn labeled sample, the optimal classifier known in the literature as the Bayes classifier, which by definition has minimum loss.

The following notation will be used in the sequel: We write *const* to denote absolute constants or constants which do not depend on other variables in the mathematical expression. We denote by $\{(x_j, y_j)\}_{j=1}^{\overline{m}}$ an i.i.d. sample of labeled examples where $\overline{m}$ denotes the total sample size, y_j, $1 \leq j \leq \overline{m}$, are drawn i.i.d. and taking the integer value 'i' with probability p_i, $1 \leq i \leq M$, while the corresponding x_j are drawn according to the class conditional probability density $f_{y_j}(x)$. Denote by m_i the number of examples having a y-value of 'i'. Denote by $m = [m_1, \ldots, m_M]$ the sample size vector and let $\|m\| = \sum_{i=1}^{M} m_i \equiv \overline{m}$. The notation $\mathrm{argmin}_{k \in A} g(k)$ for a set A means the subset (of possibly more than one element) whose elements have the minimum value of g over A. A slight abuse of notation will be made by using it for countable sets where the

[1] According to the probabilistic data-generation model mentioned above, only regions in probability 1 support of the mixture distribution $f(x)$ have a well-defined class membership.

notation means the subset of elements k such that [2] $g(k) = \inf_{k'} g(k')$.

2 Learning Classification from Empirical Data

It is convenient to express the loss $L(c)$ in terms of the class-conditional losses $L_i(c)$

$$L(c) = \sum_{i=1}^{M} p_i L_i(c)$$

where $L_i(c) = \mathrm{E}_i 1_{\{c(x) \neq i\}}$, and E_i is the expectation with respect to the density $f_i(x)$. We may define the empirical counterparts of the loss and conditional loss as

$$L_m(c) = \sum_{i=1}^{M} p_i L_{i,m_i}(c) \tag{1}$$

where

$$L_{i,m_i}(c) = \frac{1}{m_i} \sum_{j:y_j=i} 1_{\{c(x_j) \neq i\}}.$$

A classifier c may be represented by different types of classifiers, for instance, a neural network, a labeled nearest-neighbor partition, linear discriminants and others. Usually in practice one is restricted to a single type of model, say nearest neighbor classifiers, in which case it is convenient (cf. Vapnik [1982], Devroye et. al. [1996]) to consider the family of classifiers as a nested structure of subclasses each of a fixed complexity $k \in \mathbb{Z}_+$. For instance, if we consider the space $\mathcal{C}$ of *all* nearest neighbor classifiers in $\mathbb{R}^d$ then k denotes the number of prototypes used in the classifier. The complexity of $\mathcal{C}$ is clearly infinite since it contains also classifiers with infinite number of prototypes. We leave the notion of complexity of a class of multi-category classifiers general and postpone its precise definition for later sections. The space $\mathcal{C}$ may be defined as the union of classes $\mathcal{C}_k$ of classifiers having a total number k of prototypes.

Each finite complexity class $\mathcal{C}_k$ contains an optimal classifier c_k^* which minimizes the loss $L(c)$ and is written as $c_k^* = \mathrm{argmin}_{c \in \mathcal{C}_k} L(c)$. The best performing classifier in $\mathcal{C}$ denoted as c^* is defined as $c^* = \mathrm{argmin}_{1 \leq k \leq \infty} L(c_k^*)$. Denoting

[2] In that case, technically, if there does not exists a k in A such that $g(k) = \inf_{k'} g(k')$ then we can always find an arbitrarily close approximating elements k_n, i.e., $\forall \epsilon > 0$ $\exists N(\epsilon)$ such that for $n > N(\epsilon)$ we have $|g(k_n) - \inf_{k'} g(k')| < \epsilon$.

by k^* the minimal complexity of a class which contains c^*, then depending on the problem and on the type of classifiers used, k^* may even be infinite as in the case when the Bayes classifier is not contained in $\mathcal{C}$. We will refer to k^* also as the intrinsic complexity of the Bayes classifier.

Similarly, denote by $\hat{c}_k$ the empirically-best classifier in $\mathcal{C}_k$, i.e., $\hat{c}_k = \text{argmin}_{c \in \mathcal{C}_k} L_m(c)$. We are going to assume that $\mathcal{C}_k$ is sufficiently rich such that for large enough k we can find a classifier which is *consistent* with the whole sample, i.e., has a zero empirical loss.

However the true loss $L(\hat{c}_k)$ does not necessarily decrease with k. This is a consequence of the well known bias v.s. variance tradeoff in statistics (see e.g. Kendall & Stuart [1994], Geman et. al. [1992], Meir [1994]) which in our context implies a tradeoff between learning accuracy (which is inversely proportional to the classifier class complexity k) and optimal loss $L(c_k^*)$, cf. Barron [1994], Lugosi & Zeger [1996]Ratsaby et. al. [1996].

2.1 Model Selection Criterion

The primary aim of learning should be to select a classifier $\hat{c}_k$ which does not necessarily achieve a zero empirical loss but one which generalizes well from the finite training sample, i.e., has a minimal true loss $L(\hat{c}_k)$. The latter depends on the unknown underlying pattern-class conditional probability distributions hence it is necessary to base the selection on some type of estimate of the true loss.

The area in statistics known as *model selection*, see for instance Linhart & Zucchini [1986], suggests numerous loss-estimates, also known as *criteria* for model selection, which include estimates based on leave-one-out cross validation, jackknife and bootstrap estimates, asymptotic upper bounds on maximum likelihood estimates (e.g., the Akaike Information Criterion) and others. Non-asymptotic upper bounds which hold uniformly over classes of estimators have been introduced by Vapnik & Chervonenkis [1981], and have since been used as a criterion for model selection known as Structural Risk Minimization (SRM), see Vapnik [1982], Devroye et. al. [1996], Shawe-Taylor et. al. [1998], Lugosi & Nobel [1999], Ratsaby et. al. [1996].

For the purpose of reviewing other published results we use m as a scalar sample size variable just for the remaining of this section. Many model selection criteria may be represented by a sum of the form $L_m(\hat{c}_k) + \epsilon(m, k)$ where $\epsilon(m, k)$ is some increasing function of k and is sometimes referred to as a *complexity penalty*, see for instance Barron [1994], Lugosi & Zeger [1996], Buescher

& Kumar [1996]. The classifier chosen by the criterion is then defined by

$$\hat{c}^* = \text{argmin}_{1 \leq k \leq \infty} \left(L_m(\hat{c}_k) + \epsilon(m,k) \right). \tag{2}$$

In SRM, the term $\epsilon(m,k)$ is related to the worst case deviations between the true loss and the empirical loss uniformly over all functions in some class $\mathcal{C}_k$ of a fixed complexity k which for the case of boolean classifiers (i.e., $M = 2$) is defined as the *Vapnik-Chervonenkis*-dimension [3] cf. Vapnik [1982], Devroye et. al. [1996]. We will take the penalty to be (cf. Vapnik [1982] Chapter 8, Devroye et. al. [1996])

$$\epsilon(m,k) = const \sqrt{\frac{k \ln m}{m}} \tag{3}$$

where again *const* stands for an absolute constant. This bound is central to the computations of the paper [4]. As will be later shown, a procedure of gradient descent will minimize a criterion (6) based on $\epsilon(m,k)$ and the *const* becomes unimportant as it appears symmetrically in all components of the gradient.

We note that for the two pattern classification case, $M = 2$, cf. Devroye et. al. [1996] section 18.1, the error rate of the SRM-chosen classifier, henceforth denoted by $\hat{c}^*$ (which implicitly depends on the random sample of size m since it is obtained by minimizing the sum in (2)), satisfies

$$L(\hat{c}^*) > L(c^*) + const \sqrt{\frac{k^* \ln m}{m}} \tag{4}$$

infinitely often with probability 0 where c^* is the Bayes classifier which is assumed to be included in $\mathcal{C}$ and k^* is its intrinsic complexity. The assumption that the Bayes classifier is in $\mathcal{C}$ is not very severe as $\mathcal{C}$ may have an infinite VC dimension. From (4) it is apparent that aside from being consistent, the SRM-chosen classifier automatically locks onto the error rate as if k^* is *known* beforehand. Due to this nice property we choose SRM as the learning approach for the classification problem. We note in passing that recently there has been interest in data-dependent penalty terms for structural risk minimization which do not have an explicit complexity factor k but are related to

[3] For a class H of functions from a set X to $\{0,1\}$ and a set $S = \{x_1, \ldots, x_l\}$ of l points in X, denote by $H_{|S} = \{[h(x_1), \ldots, h(x_l)] : h \in H\}$. Then the Vapnik-Chervonenkis dimension of H denoted by $VC(H)$ is the largest l such that the cardinality $\left|H_{|S}\right| = 2^l$.

[4] There is actually an improved bound due to Talagrand, cf. Anthony & Bartlett [1999] Section 4.6, but when adapted for almost sure statements it yields $O(\sqrt{\frac{k+\ln m}{m}})$ which is insignificantly better than (3) at least for our work.

the class $\mathcal{C}_k$ by being defined as a supremum of some empirical quantity over $\mathcal{C}_k$, for instance the maximum discrepancy criterion [Bartlett et. al., 2002] or the Rademacher complexity [Kultchinskii, 2001].

The primary aim of this paper is to answer the following main question:

Question: Let $\overline{m}$ be the total number of examples available for training. Suppose that it is possible for a learner to query for i.i.d. labeled examples randomly drawn from *particular* pattern classes which are *not* necessarily selected at random according to their *a priori* probabilities p_i, $1 \leq i \leq M$. Can the error rate of learning via SRM be improved by such sample-querying ?

This question is not only interesting from a theoretical standpoint but is also motivated by real pattern classification problems (cf. Ratsaby [1998]). There, an application of the algorithm SQ (in Section 3.3) was realized based on k-NN classifiers. It improved the error rate compared to the setting where equal sized sub-samples are obtained *a priori* from every pattern class, this being the standard approach to on-line learning when the *a priori* class probabilities are unknown.

This non classical scheme of randomly drawing examples is related to *active learning* in which the learner actively engages in the process of sample selection. Some of the works in this area include Angluin [1988], Cohn et. al. [1994], Cohn [1996], Kulkarni et. al. [1993], Niyogi [1995], Rivest & Eisenberg [1990]. The common denominator here is the fact that some form of interaction between the learner and teacher which enables the learner to obtain labeled examples *not* only in a passive manner, as has been considered classically in the field of pattern recognition, leads to an improvement in the learning accuracy. Sample querying is also related to boosting [Freund & Schapire, 1995] since both seek a better weighting for the different parts of the sample except boosting is stuck with a given sample while here we allow the learner to acquire the sample as learning proceeds. Also related to that is stratified sampling which aims at getting a lower sampling error by oversampling smaller groups and then re-weighting, see also Japkowicz [2000].

In this paper we answer the above posed question in the affirmative which thereby provides further support in favor of active learning.

3 Querying for examples as means of improving the learning accuracy

A classifier $c(x)$ may be represented as a vector of M boolean classifiers $b_i(x)$, where $b_i(x) = 1$ if x is a pattern drawn from class 'i' and $b_i(x) = 0$ otherwise.

A union of such boolean classifiers forms a *well-defined classifier* $c(x)$ if for each $x \in \mathbb{R}^d$, $b_i(x) = 1$ for exactly one i, i.e., $\bigcup_{i=1}^{M}\{x : b_i(x) = 1\} = \mathbb{R}^d$ and $\{x : b_i(x) = 1\} \bigcap \{x : b_j(x) = 1\} = \emptyset$, for $1 \leq i \neq j \leq M$.

We also refer to these boolean classifiers as the component classifiers $c_i(x)$, $1 \leq i \leq M$, of a vector classifier $c(x)$.

With such a representation, the loss of a classifier c is the average of the losses of the component classifiers, i.e., $L(c) = \sum_{i=1}^{M} p_i L(c_i)$ where for a boolean classifier c_i the loss is defined as $L(c_i) = \mathrm{E}_i 1_{\{c_i(x)\neq 1\}}$, and the empirical loss is $L_{i,m_i}(c_i) = \frac{1}{m_i}\sum_{j=1}^{m_i} 1_{\{c_i(x_j)\neq 1\}}$ which is based on a subsample $\{(x_j, i)\}_{j=1}^{m_i}$ drawn i.i.d. from pattern class "i".

The class $\mathcal{C}$ of classifiers is decomposed into a structure $S = S_1 \times S_2 \times \cdots \times S_M$, where S_i is a nested structure (cf. Vapnik [1982]) of classes $\mathcal{B}_{k_i}$, $i = 1, 2, \ldots$, of boolean classifiers $b_i(x)$, i.e.,

$$S_1 = \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{k_1}, \ldots$$

$$S_2 = \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{k_2}, \ldots$$

up to

$$S_M = \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{k_M}, \ldots$$

where $k_i \in \mathbb{Z}_+$ denotes the VC-dimension of $\mathcal{B}_{k_i}$ and $\mathcal{B}_{k_i} \subseteq \mathcal{B}_{k_i+1}$, $1 \leq i \leq M$.

For any fixed positive integer vector $k \in \mathbb{Z}_+^M$ consider the class of vector classifiers

$$\mathcal{C}_k = \mathcal{B}_{k_1} \times \mathcal{B}_{k_2} \times \cdots \times \mathcal{B}_{k_M}. \tag{5}$$

Define by $\mathcal{G}_k$ the subclass of $\mathcal{C}_k$ of classifiers c that are well-defined (in the sense mentioned above). Note that every $c \in \mathcal{G}_k$ corresponds to a well defined classifier while any combination of $b_1 \in \mathcal{G}_{k_1}$, $b_2 \in \mathcal{G}_{k_2}$, ..., and $b_M \in \mathcal{G}_{k_M}$ is not necessarily a well-defined classifier $[b_1(x), \ldots, b_M(x)]$.

For vectors m and k in $\mathbb{Z}_+^M$, define

$$\epsilon(m, k) \equiv \sum_{i=1}^{M} p_i \epsilon(m_i, k_i) \tag{6}$$

where $\epsilon(m_i, k_i)$ is defined according to (3). For any $0 < \delta < 1$, we denote by $\epsilon(m_i, k_i, \delta) = \sqrt{\frac{k_i \ln m_i + \ln \frac{1}{\delta}}{m_i}}$ and $\epsilon(m, k, \delta) = \sum_{i=1}^{M} p_i \epsilon(m_i, k_i, \delta)$.

The next lemma states an upper bound on the deviation between the empirical loss and the loss uniformly over all classifiers in a class $\mathcal{G}_k$ and is a direct application of Theorem 6.7 Vapnik [1982].

Before we state it, it is necessary to define what is meant by an increasing sequence of vectors m.

Definition 1 (Increasing sample-size sequence) *A sequence $m(n)$ of sample-size vectors is said to increase if*

- *at every n, there exists a j such that $m_j(n+1) > m_j(n)$ and $m_i(n+1) \geq m_i(n)$ for $1 \leq i \neq j \leq M$,*
- *there exists an increasing function $T(N)$ such that for all $N > 0$, $n > N$ implies every component $m_i(n) > T(N)$, $1 \leq i \leq M$.*

Note that Definition 1 implies for all $1 \leq i \leq M$, $m_i(n) \to \infty$ as $n \to \infty$. We will henceforth use the notation $m \to \infty$ to denote such an ever-increasing sequence $m(n)$ with respect to an implicit discrete indexing variable n. The relevance of Definition 1 will become clearer later, in particular when considering Lemma 3.

Definition 2 (Sequence generating procedure) *A* sequence generating procedure *ϕ is one which generates increasing sequences $m(n)$ with a* fixed *function $T_\phi(N)$ as in Definition 1 and also satisfying the following: for all $N, N' \geq 1$ such that $T_\phi(N') = T_\phi(N)+1$ then $|N'-N| \leq const$, where const is dependent only on ϕ.*

The above definition simply states a lower bound requirement on the rate of increase of $T_\phi(N)$.

We now state the uniform strong law of large numbers for the class of well-defined classifiers.

Lemma 1 *For any $k \in \mathbb{Z}_+^M$ let $\mathcal{G}_k$ be a class of well-defined classifiers. Consider any sequence-generating procedure as in Definition 2 which generates $m(n)$, $n = 1, \ldots, \infty$. Based on examples $\{(x_j, y_j)\}_{j=1}^{\overline{m}(n)}$, each drawn i.i.d. according to an unknown underlying distribution over $\mathbb{R}^d \times \{1, \ldots, M\}$, define the empirical loss as in (1). Then*

- *For arbitrary $0 < \delta < 1$,*

$$\sup_{c \in \mathcal{G}_k} \left| L_{m(n)}(c) - L(c) \right| \leq const\ \epsilon(m(n), k, \delta)$$

  with probability $1 - \delta$ and
- *the events $\sup_{c \in \mathcal{G}_k} \left| L_{m(n)}(c) - L(c) \right| > const\ \epsilon(m(n), k)$, $n = 1, 2, \ldots$, occur infinitely often with probability 0,*

where $m(n)$ is any sequence generated by the procedure.

The proof is in Section A.

We will henceforth denote by c_k^* the optimal classifier in $\mathcal{G}_k$, i.e.,

$$c_k^* = \text{argmin}_{c \in \mathcal{G}_k} L(c) \tag{7}$$

and

$$\hat{c}_k = \text{argmin}_{c \in \mathcal{G}_k} L_m(c) \tag{8}$$

is the empirical minimizer over the class $\mathcal{G}_k$.

In the next section we consider several learning settings in which sample querying is useful.

3.1 Motivation

As part of the motivation for our work in succeeding sections, let us first answer the main question posed in Section 2 for the simplest case where there is only a single classifier class $\mathcal{G}_k$ with a complexity vector $k = [l, \ldots, l]$, for some finite positive integer l. In this setting, the problem of learning classification may be well represented by the PAC model (Valiant [1984], Haussler [1992], Blumer et. al. [1989]) as follows: For arbitrary accuracy parameter $\eta > 0$ and confidence parameter $0 < \delta < 1$, based on an i.i.d. labeled sample of size $\overline{m}$, the learner aims at outputting a hypothesis classifier $\hat{c}_k$, as defined in (8), such that $L(\hat{c}_k) \leq L(c_k^*) + \eta$ with confidence $1 - \delta$, where c_k^* is defined in (7). The classifier $\hat{c}_k$ is said to be an (η, δ)-good estimate of c_k^*.

From Lemma 1 it follows that

$$\begin{aligned} L(\hat{c}_k) &\leq L_m(\hat{c}_k) + \epsilon(m, k, \delta) \\ &\leq L_m(c_k^*) + \epsilon(m, k, \delta) \\ &\leq L(c_k^*) + 2\epsilon(m, k, \delta) \end{aligned}$$

with confidence $1-\delta$. Choosing *any* sample size vector m such that $\epsilon(m, k) \leq \frac{\eta}{2}$ yields a $\hat{c}_k$ which is (η, δ)-good. In order to find the minimizing subsample proportions which we denote by the sample size vector m^*, we minimize $\sum_{i=1}^M m_i$ under the constraint that $\sum_{i=1}^M p_i \epsilon(m_i, k_i) = \frac{\eta}{2}$. This yields $m_i^* = const \left(l \left(\frac{\overline{m} p_i}{\eta} \right)^2 \right)^{\frac{1}{3}} \log \left(\frac{\overline{m} p_i l}{\eta} \right)$ where $const > 0$ is an absolute constant,

$1 \leq i \leq M$, for a total sample size $\overline{m} = \sum_{i=1}^{M} m_i^*$. We note in passing that for the more restricted setting where the classifier class contains a Bayes optimal classifier with a zero misclassification probability then the bound of (3) can be strengthened to one without a square-root, cf. Section 4.5 in Anthony & Bartlett [1999].

As the next case consider a class $\mathcal{G}_k$ of classifiers which has an arbitrary but finite complexity $k \in \mathbb{Z}_+^M$, i.e., with elements k_i, $1 \leq i \leq M$, which are not necessarily all equal. Following the same reasoning as before we conclude that the minimizing subsample proportions m_i^* need to satisfy a set of non-linear equations $m_i = \overline{m} \frac{p_i \epsilon(m_i, k_i)}{\epsilon(m,k)}$ which depend on k and on the *a priori* class probabilities.

In both of the cases above the learner is forced to use a certain classifier class and knowing its complexity vector k he may then compute the best subsample sizes for batch learning, i.e., where by batch we mean all the labeled examples are obtained *prior* to running the empirical minimization algorithm. However in many instances of non-parametric classification (as well as parametric ones), cf. Ripley [1996], Duda et. al. [2001], the learner is theoretically free to use arbitrarily complex classes, e.g., nearest-neighbor classifiers having arbitrarily large number of prototypes as discussed in Section 2. In such circumstances the above settings of a single fixed hypothesis class do not apply and it is necessary to consider more flexible class structures as the ones introduced in the beginning of Section 3. Here the learning algorithm uses some form of model selection that automatically selects a class which balances the tradeoff between the empirical loss and the complexity penalty.

If we consider again the main question posed earlier but this time under this richer class setting, the answer is not at all obvious since the complexity chosen by a model selection criterion is determined only *after* the sample has been drawn leaving no room for querying for fine-tuned subsample sizes that minimize the upper bound on the loss of the chosen classifier. This difficulty is intrinsic to batch learning where querying needs to be done in advance.

However, as shown in this paper, it is possible to interleave sampling with learning and hence potentially obtain sub-samples, one per pattern class, of different sizes. The question remains as to what sub-sample size proportions m_i yield a better loss rate. In particular, if one resorts to a model-selection learning criterion then the complexity of the classifier class $\mathcal{G}_k$ can change as the sample increases hence one cannot apportion sample sizes as in the previous two settings.

In Section 2 we mentioned the nice property of the method of SRM which effectively yields a loss rate as if the minimal complexity class containing the Bayes classifier was known in advance. Essentially, the intrinsic unknown

complexity k^* of the Bayes classifier is automatically learned by the SRM criterion. Hence it should be possible to minimize an upper bound of the form of (4), but for vector sample size m, and to yield an even better selected classifier, i.e., one whose loss is not $\epsilon(m, k^*)$ but $\epsilon(m^*, k^*)$ where m^* minimizes the criterion.

This raises an interesting path to proceed as far as querying is concerned. It says that if the intrinsic complexity of the Bayes classifier could be predicted early in time then based on it an estimated criterion involving the sample size m can be defined. Querying should be done in a manner which minimizes this estimated criterion and which hopefully yields subsample proportions which are close to those minimizing the 'true' criterion, i.e., the one involving the intrinsic Bayes complexity k^*.

The remainder of the paper will be devoted to doing precisely that. It will be shown that the complexity of the classifier chosen by the method of SRM is a consistent estimator of the Bayes complexity k^*. We next outline our approach: First we present additional notation concerning some complexities that are associated with the method of SRM over structures of well-defined classifier classes. We then state Lemma 2 (proved in Section B) which establishes an estimate on the loss of the SRM-selected classifier and the convergence of its complexity with increasing i.i.d. sample sizes. Corollary 1 (proved in Section C) states the same results for samples that are i.i.d. only when conditioned on the pattern class thereby allowing the examples to be drawn from pattern classes even in a dependent manner. This allows introducing an on-line algorithm which combines SRM with sample querying and then establish (through Theorem 1) its optimality in a certain sense. In Section 4 we analyze the convergence properties of this on-line algorithm, first just applying the query-rule to a deterministic criterion (Lemma 4) and then to the realistic case of a random criterion estimate (Lemma 5). At that point the necessary results for proving the main Theorem 1 are in place (the proof is in Section E).

One comment concerning the convergence mode of random variables. Upper bounds are based on the uniform strong law of large numbers, see proof of Lemma 1 in Section A. Such bounds originated in the work of Vapnik [1982], for instance his Theorem 6.7. Throughout the current paper, *almost sure* statements are made by a standard application of the Borel-Cantelli lemma. For instance, taking m to be a scalar, the statement $\sup_{b\in B_r} |L(b) - L_m(b)| \leq const\sqrt{\frac{r\log m+\log\frac{1}{\delta}}{m}}$ with probability at least $1-\delta$ for any $\delta > 0$ is alternatively stated as follows by letting $\delta_m = \frac{1}{m^2}$: For the sequence of random variables $L_m(b)$, uniformly over all $b \in B$, we have $L(b) > L_m(b) + const\sqrt{\frac{r\log m+\log\frac{1}{\delta_m}}{m}}$ occur infinitely often with probability 0.

Finally, concerning our, perhaps, loose use of the word *optimal*, whenever not explicitly stated, optimality of a classifier or of a procedure or algorithm is only with respect to minimization of the criterion, namely, the upper bound on the loss. In particular, it is not intended to claim that the algorithm introduced later is optimal with respect to other sample querying approaches but that it minimizes the upper bound on the loss.

3.2 Structural Risk Minimization for Multi-Category Classifiers

We will henceforth make the following assumption.

Assumption 1 *The Bayes loss $L^* = 0$ and there exists a classifier c_k in the structure S with $L(c_k) = L^*$ such that $k_i < \infty$, $1 \leq i \leq M$. The* a priori *pattern class probabilities p_i, $1 \leq i \leq M$, are known to the learner.*

Before continuing we make a few remarks.

Remark 1 *It is assumed that the Bayes loss of the underlying classification problem is zero and that the structure S is rich enough and contains the Bayes classifier. The problem of learning classification under the restriction that the target Bayes classifier has a zero loss is not necessarily easy or trivial since it can have an arbitrarily complex decision border. Such problems have been extensively studied, for instance, in the Probably Approximately Correct (PAC) framework, cf. Blumer et. al. [1989], and the proceedings of conferences on computational learning theory (COLT), see also Devroye et. al. [1996] Section 12.7.*

Remark 2 *In practice the* a priori *pattern class probabilities can be estimated easily. In assuming that the learner knows the p_i, $1 \leq i \leq M$, one approach would have the learner allocate sub-sample sizes according to $m_i = p_i \overline{m}$ followed by doing structural risk minimization (this actually corresponds to passive learning where the teacher provides the samples according to the* a priori *pattern class probabilities). Note that this does not necessarily minimizes the upper bound on the loss of the SRM-selected classifier and hence is inferior in this respect as Principle 1 states later.*

Remark 3 *We note that if the classifier class was* fixed *and the intrinsic complexity k^* of the Bayes classifier was known in advance then because of Assumption 1 one would resort to a bound of the form $O(k^* \log m/m)$ and not the weaker bound of (3). However, as mentioned before, not knowing k^* and hence using structural risk minimization as opposed to empirical risk minimization over a fixed class, necessitates using (3) as the upper bound or complexity-penalty.*

Remark 4 *If one uses the expected value $EL(\hat{c})$ as a criterion, where $\hat{c}$ is the learnt classifier which is dependent on the random sample (the expectation here is taken with respect to this random sample) then the bound can be improved from (3) to $O(\log m/m)$ as noted in Problem 18.4 in Devroye et. al. [1996]. However, it is well-known that such expected losses are less interesting and less realistic since usually one has to live with the particular random data set at hand and does not have the luxury of obtaining multiple data sets with which to take averages. The loss $L_m(c)$ used in this paper is a random variable since it depends on a random data set and is not an expectation with respect to the data distribution. The upper bound defined in (3) is therefore not weak.*

We continue now with introducing some concepts that will be used for defining our sampling-criterion.

Consider the set

$$F^* = \{\text{argmin}_{k \in \mathbb{Z}_+^M} L(c_k^*)\} = \{k : L(c_k^*) = L^* = 0\} \tag{9}$$

which may contain more than one vector k. Following Assumption 1 we may define the *Bayes classifier* c^* as the particular classifier $c_{k^*}^*$ whose complexity is minimal, i.e.,

$$k^* = \text{argmin}_{\{k \in F^*\}} \{\|k\|_\infty\} \tag{10}$$

where $\|k\|_\infty = \max_{1 \le i \le M} |k_i|$. Note again that there may be more than one such k^*. The significance of specifying the Bayes classifier up to its complexity rather than just saying it is any classifier having a loss L^* will become apparent later in the paper.

For an empirical-minimizer classifier $\hat{c}_k$, define by the *penalized empirical loss* (cf. Devroye et. al. [1996]) $\tilde{L}_m(\hat{c}_k) = L_m(\hat{c}_k) + \epsilon(m, k)$. Consider the set

$$\hat{F} = \{\text{argmin}_{k \in \mathbb{Z}_+^M} \tilde{L}(\hat{c}_k)\} \tag{11}$$

which may contain more than one vector k. In structural risk minimization according to Vapnik [1982] the selected classifier is *any* one whose complexity index $k \in \hat{F}$.

For our purposes the original definition of the SRM-selected classifier is not sufficient since by its definition it may have any complexity as long as it minimizes the criterion, namely, the sum of the empirical loss and the penalty, over $k \in \mathbb{Z}_+^M$. The algorithm to be introduced later relies on the convergence

of the complexity $\hat{k}$ to some finite limiting complexity value with increasing[5] m. The selected classifier is one whose complexity satisfies

$$\hat{k} = \text{argmin}_{k \in \hat{F}} \|k\|_\infty. \tag{12}$$

That is, among all classifiers which minimize the penalized empirical error we choose the one having a minimal complexity magnitude. This *minimal-complexity SRM-selected classifier* will be denoted as $\hat{c}_{\hat{k}}$ or simply as $\hat{c}^*$. We sometimes write $\hat{k}_n$ and $\hat{c}_n^*$ for the complexity and for the SRM-selected classifier, respectively, in order to explicitly show the dependence on discrete time n.

The next lemma states that the complexity $\hat{k}$ converges to some (not necessarily unique) k^* corresponding to the Bayes classifier c^* defined in (10).

Lemma 2 *Based on $\overline{m}$ examples $\{(x_j, y_j)\}_{j=1}^{\overline{m}}$ each drawn i.i.d. according to an unknown underlying distribution over $\mathbb{R}^d \times \{1, \ldots, M\}$, let $\hat{c}^*$ be the chosen classifier of complexity $\hat{k}$ as in (12). Consider a sequence of samples $\zeta^{m(n)}$ with increasing sample-size vectors $m(n)$ obtained by a sequence-generating procedure as in Definition 2. Then*

- *the corresponding complexity sequence $\hat{k}_n$ converges a.s. to some k^* as defined in (10) which from Assumption 1 has finite components.*
- *For any sample $\zeta^{m(n)}$ in the sequence, the loss of the corresponding classifier $\hat{c}_n^*$ satisfies*

  $$L(\hat{c}_n^*) > const\ \epsilon(m(n), k^*)$$

  infinitely often with probability 0.

The proof is in Section B.

Remark 5 *For the more general case of $L^* > 0$ (but two-category classifiers) the upper bound becomes $L^* + const\ \epsilon(m, k^*)$, cf. Devroye et. al. [1996]. It is an open question whether in this case it is possible to guarantee convergence of $\hat{k}_n$ or some variation of it to a finite limiting value.*

That querying for randomly drawn examples from particular pattern classes may serve useful is seen from being able to minimize the loss rate of $\hat{c}^*$ with respect to the sample size vector m. The principal idea of our work is realizing that the subsample proportions may be tuned to the intrinsic Bayes complexity

[5] We will henceforth adopt the convention that a vector sequence $\hat{k}_n \to k^*$, a.s., means that every component of $\hat{k}_n$ converges to the corresponding component of k^*, a.s., as $m \to \infty$.

k^* thereby yielding an improved loss rate for $\hat{c}^*$. We formally state this in the following:

Principle 1 *Choose m to minimize the criterion $\epsilon(m, k^*)$ with respect to all m such that $\sum_{i=1}^{M} m_i = \overline{m}$, the latter being the a priori total sample size allocated for learning.*

There may be other proposed sampling criteria just as there are many criteria for model selection based on minimization of different upper bounds. Our proposed sample-querying can be viewed as paralleling the structural risk minimization approach of model selection.

If k^* was known then an optimal sample size $m^* = [m_1^*, \ldots, m_M^*]$ could be computed which yields a classifier $\hat{c}^*$ with the best (lowest) deviation *const* $\epsilon(m^*, k^*)$ away from Bayes loss. The difficulty is that $k^* = [k_1^*, \ldots, k_M^*]$ is usually unknown since it depends on the underlying unknown probability densities $f_i(x)$, $1 \leq i \leq M$. To overcome this we will minimize an estimate of $\epsilon(\cdot, k^*)$ rather than the criterion $\epsilon(\cdot, k^*)$ itself.

3.3 An On-Line Learning Algorithm

In this section we introduce an on-line learning algorithm which repetitively cycles between running SRM over the current sample and querying for more examples in a manner which asymptotically has the criterion estimate converging to the true unknown criterion. The interleaved querying step ensures that this true criterion is minimized eventually. As before, $m(n)$ denotes a sequence of sample-size vectors indexed by an integer $n \geq 0$ representing discrete time. When referring to a particular i^{th} component of the vector $m(n)$ we write $m_i(n)$.

The algorithm initially starts with uniform sample size proportions, i.e., $m_1 = m_2 = \cdots = m_M = const > 0$. Then at each time $n \geq 1$ the algorithm determines the SRM-selected classifier $\hat{c}_n^*$ defined as

$$\hat{c}_n^* = \text{argmin}_{\hat{c}_{n,k}: k \in \hat{F}_n} \|k\|_\infty \qquad \mathbf{S - step} \tag{13}$$

where

$$\hat{F}_n = \{k : \tilde{L}_n(\hat{c}_{n,k}) = \min_{r \in \mathbb{Z}_+^M} \tilde{L}_n(\hat{c}_{n,r})\}$$

and for any $\hat{c}_{n,k}$ which minimizes $L_{m(n)}(c)$ over all $c \in \mathcal{G}_k$ we define $\tilde{L}_n(\hat{c}_{n,k}) = L_{m(n)}(\hat{c}_{n,k}) + \epsilon(m(n), k)$ while $L_{m(n)}()$ stands for the empirical loss as defined in

(1) using the sample size vector $m(n)$ at time n. The complexity $\hat{k}_n$ of $\hat{c}_n^*$ will be shown later to converge to k^* hence $\epsilon(\cdot, \hat{k}_n)$ serves as a consistent estimator of the criterion $\epsilon(\cdot, k^*)$.

We will use a query rule which depends on the observed sample. While for any *fixed* $i \in \{1, 2, \ldots, M\}$ the examples $\{(x_j, i)\}_{j=1}^{m_i(n)}$ accumulated up until time n are all i.i.d. random variables, the total sample $\{(x_j, y_j)\}_{j=1}^{\overline{m}(n)}$ consists of *dependent* random variables since by the query decision the choice of the particular class-conditional probability distribution used to draw examples at each time instant l depends on the sample accumulated up until time $l-1$. As the next Corollary shows, this dependency does not alter the results of Lemma 2.

Corollary 1 *At time n, based on M subsamples $\{(x_j, i)\}_{j=1}^{m_i(n)}$, each of which contains an i.i.d. sample which is drawn according to $f_i(x)$, $1 \leq i \leq M$, let $\hat{c}_n^*$ be a chosen classifier as defined in (13). Consider* any *sequence of samples $\zeta^{m(n)}$ with increasing sequence $m(n)$ as $n \to \infty$ generated by a sequence-generating procedure. Then the corresponding complexity sequence*

$$\hat{k}_n \to k^*, \qquad a.s. \quad with\ n \to \infty$$

for some k^* as defined in (10) which from Assumption 1 has finite components. Furthermore, for any sample $\zeta^{m(n)}$ in the sequence, the loss of the corresponding $\hat{c}_n^*$ has

$$L(\hat{c}_n^*) > const\ \epsilon(m(n), k^*)$$

infinitely often with probability 0.

The proof is deferred to the Section C.

According to Principle 1, if k^* was known then a natural query-step would be to adapt the sample vector $m(n)$ in a direction which minimizes the criterion $\epsilon(\cdot, k^*)$. As k^* is unknown, we will instead base the query step on minimizing the estimate $\epsilon(\cdot, \hat{k}_n)$ of $\epsilon(\cdot, k^*)$. While doing that it must be assured that the query-step results in $m(n)$ increasing to $m(n+1)$ as defined in Definition 1. This is required since once having obtained a sample of size-vector $m(n)$ at time n it makes no sense to throw away some examples, i.e., decrease the sample size, in particular where for $\hat{k}_n$ to converge to k^* it is necessary to have an ever-increasing sample size sequence $m(n)$.

There are various ways of defining a query-step given the restrictions above. We choose here a greedy query rule which adapts only one component of m at a time, namely, it increases the component $m_{j_{max}(n)}$ which corresponds to

the direction of maximum descent of the criterion $\epsilon(\cdot, \hat{k}_n)$ at time n. This may be written as

$$m(n+1) = m(n) + \Delta\, e_{j_{max}} \qquad \mathbf{Q-step} \tag{14}$$

where the positive integer Δ denotes some fixed query step size and for any integer $i \in \{1, 2, \ldots, M\}$, e_i denotes an M-dimensional elementary vector with 1 in the i^{th} component and 0 elsewhere.

Thus at time n the query-step produces $m(n+1)$ which is used for drawing additional examples according to specific sample sizes $m_i(n+1)$, $1 \leq i \leq M$. Consequently the SRM-step (13) is repeated, this time using the newly acquired sample of size vector $m(n+1)$.

We now state the learning algorithm explicitly. The querying rule will be discussed in the next section where it is proved that it obtains the minimizing sample-size vector in the limit with respect to increasing n. The notation $a := b$ represents assigning the variable a with the value of the variable b.

Learning Algorithm SQ (SRM with Queries)
Let: $m_i(0) = const > 0$, $1 \leq i \leq M$.
Given: (a) M uniform-size samples $\{\zeta^{m_i(0)}\}_{i=1}^{M}$, where $\zeta^{m_i(0)} = \{(x_j, \text{`i'})\}_{j=1}^{m_i(0)}$, and x_j are drawn i.i.d. according to underlying class-conditional probability densities $f_i(x)$. (b) A sequence of classes $\mathcal{G}_k$, $k \in \mathbb{Z}_+^M$, of well-defined classifiers. (c) A constant query-step size $\Delta > 0$. (d) Known *a priori* probabilities p_j, $1 \leq j \leq M$.
Initialization: (Time $n = 0$**)** Based on $\zeta^{m_i(0)}$, $1 \leq i \leq M$, determine a set of candidate classifiers $\hat{c}_{0,k}$ minimizing the empirical loss $L_{m(0)}$ over $\mathcal{G}_k$, , $k \in Z_+^M$, respectively. Determine $\hat{c}_0^*$ according to (13) and denote its complexity vector by $\hat{k}_0$.
Output: $\hat{c}_0^*$.
Call Procedure Greedy-Query: $m(1) := GQ(0)$.
Let $n = 1$**.**
While (still more available examples) **Do:**
1. Based on the sample $\zeta^{m(n)}$, determine the empirical minimizers $\hat{c}_{n,k}$ for each class $\mathcal{G}_k$. Determine $\hat{c}_n^*$ according to (13) and denote its complexity vector by $\hat{k}_n$.
2. **Output:** $\hat{c}_n^*$.
3. **Call Procedure Query:** $m(n+1) := GQ(n)$.
4. $n := n + 1$.

End Do

Procedure Greedy-Query (GQ)

Input: Time n.

1. $j_{max}(n) := \text{argmax}_{1\leq j\leq M}\, p_j \frac{\epsilon(m_j(n),\hat{k}_{n,j})}{m_j(n)}$, where if more than one argmax then choose any one.
2. **Obtain:** Δ new i.i.d. examples from class $j_{max}(n)$. Denote them by ζ_n.
3. **Update Sample:** $\zeta^{m_{j_{max}(n)}(n+1)} := \zeta^{m_{j_{max}(n)}(n)} \bigcup \zeta_n$, while $\zeta^{m_i(n+1)} := \zeta^{m_i(n)}$, for $1 \leq i \neq j_{max}(n) \leq M$.
4. **Return Value:** $m(n) + \Delta\, e_{j_{max}(n)}$.

□

Algorithm SQ alternates between the SRM-step (13) and the Query-step (14) repetitively until finally exhausting the total sample size limit $\overline{m}$ which for most generality is assumed to be unknown to the learner.

The next lemma implies that the previous Corollary 1 applies also to Algorithm SQ.

Lemma 3 *Algorithm SQ is a sequence-generating procedure.*

The proof is deferred to Section D. Next, we state the main theorem of the paper.

Theorem 1 *Assume that the Bayes complexity k^* is an* unknown *M-dimensional vector of finite positive integers. Let the step size $\Delta = 1$ in Algorithm SQ, resulting in a total sample size which increases with discrete time as $\overline{m}(n) = n$. Then the random sequence of classifiers $\hat{c}_n^*$ produced by Algorithm SQ is such that the events*

$$L(\hat{c}_n^*) > const\ \epsilon(m(n), k^*)\ \ or\ \|m(n) - m^*(n)\|_{l_1^M} > 1 \tag{15}$$

occur infinitely often with probability 0 *where $m^*(n)$ is the solution to the constrained minimization of $\epsilon(m, k^*)$ over all m of magnitude $\|m\| = \overline{m}(n)$.*

□

Remark 6 *In the limit of large n the bound const $\epsilon(m(n), k^*)$ is almost minimum (the minimum being at $m^*(n)$) with respect to all vectors $m \in \mathbb{Z}_+^M$ of size $\overline{m}(n)$. Note that this rate is achieved by Algorithm SQ without the knowledge of the intrinsic complexity k^* of the Bayes classifier. Compare this for instance to uniform querying where at each time n one queries for subsamples of the same size $\frac{\Delta}{M}$ from* every *pattern class. This leads to a different (deterministic) sequence $m(n) = \frac{\Delta}{M}[1, 1, \ldots, 1]n \equiv \underline{\Delta}\, n$ and in turn to a sequence of classifiers $\hat{c}_n$ whose loss $L(\hat{c}_n) \leq const\ \epsilon(\underline{\Delta}\, n, k^*)$, as $n \to \infty$, where here the upper bound is not even asymptotically minimal. A similar argument holds*

if the proportions are based on the a priori *pattern class probabilities since in general, letting* $m_i = p_i\overline{m}$ *does not necessarily minimize the upper bound. In Ratsaby [1998], empirical results display the inferiority of uniform sampling compared to an online sample-query approach based on Algorithm SQ.*

The proof of Theorem 1 is postponed to Section E. It is based on the previous Lemmas and on Lemma 4 and Lemma 5 of the next section both of which deal with the the convergence property of the greedy query rule stated above.

4 Technical Results

Algorithm SQ uses a query step which increments at time n the particular sample of pattern class $j_{max}(n)$ where $j_{max}(n)$ corresponds to the component of m along which the criterion function $\epsilon(m, \hat{k}_n)$ decreases the fastest. From (15), in order to analyze the convergence properties of the loss sequence $L(\hat{c}_n)$ it suffices to obtain convergence results on the random sequence of sample size vectors $m(n)$ generated by Algorithm SQ.

First, letting t, as well as n, denote discrete time $t = 1, 2, \ldots$, we adopt the notation $m(t)$ for a *deterministic* sample size sequence governed by the deterministic criterion $\epsilon(m, k^*)$. We write $m(n)$ to denote the *random* sequence governed by the stochastic criterion $\epsilon(m, \hat{k}_n)$. Thus t or n distinguish between a deterministic or a random sample sequence $m(t)$ or $m(n)$, respectively.

We then define precisely the meaning of an optimal minimizing trajectory $m^*(t)$ for the deterministic case which corresponds to the setting where k^* is known, and prove (in Lemma 4) that the query-rule 'learns' this minimizing trajectory, i.e., that $m(t) \to m^*(t)$, $t \to \infty$, in a sense to be described below.

Consequently based on the convergence of $\hat{k}_n$ to k^* we conclude (in Lemma 5) that applying the query rule to the criterion function $\epsilon(\cdot, \hat{k}_n)$, instead of $\epsilon(\cdot, k^*)$, yields a random sequence $m(n)$ such that $\|m(n) - m^*(n)\|_{l_1^M} \leq \Delta$, a.s., as $n \to \infty$, where Δ is the step size used in Algorithm SQ.

We start with the following definition.

Definition 3 (Optimal trajectory) *Let* $\overline{m}(t)$ *be any positive integer-valued function of* t *which denotes the total sample size at time* t*. The optimal trajectory is a set of vectors* $m^*(t) \in \mathbb{Z}_+^M$ *indexed by* $t \in \mathbb{Z}_+$*, defined as*

$$m^*(t) = argmin_{\{m \in \mathbb{Z}_+^M : \|m\| = \overline{m}(t)\}} \epsilon(m, k^*).$$

□

We now proceed to study the convergence properties of the sequence $m(t)$ generated by the query rule in the deterministic setting where k^* is known.

4.1 The case where k^* is known

First let us solve the following constrained minimization problem. Fix a total sample size $\overline{m}$ and minimize the error $\epsilon(m, k^*)$ under the constraint that $\sum_{i=1}^{M} m_i = \overline{m}$. This amounts to minimizing

$$\epsilon(m, k^*) + \lambda(\sum_{i=1}^{M} m_i - \overline{m}) \tag{16}$$

over m and λ. Denote the gradient by $g(m, k^*) = \nabla\epsilon(m, k^*)$. Then the above is equivalent to solving

$$g(m, k^*) + \lambda[1, 1, \ldots, 1] = 0 \tag{17}$$

for m and λ. The vector valued function $g(m, k^*)$ may be approximated by

$$g(m, k^*) \simeq \left[-\frac{p_1\epsilon(m_1, k_1^*)}{2m_1}, -\frac{p_2\epsilon(m_2, k_2^*)}{2m_2}, \ldots, -\frac{p_M\epsilon(m_M, k_M^*)}{2m_M}\right]$$

where we used the approximation $1 - \frac{1}{\log m_i} \simeq 1$ for $1 \leq i \leq M$. The approximation is appropriate as it is applied in the same manner for all components and the GQ rule treats the components symmetrically. Moreover the statements made throughout the paper are for large m.

Using this approximation for $g(m, k^*)$ and denoting the minimizing values by m_i^*, $1 \leq i \leq M$, and λ^*, we then obtain the set of equations $2\lambda^* m_i^* = p_i\epsilon(m_i^*, k_i^*)$, $1 \leq i \leq M$, and $\lambda^* = \epsilon(m^*, k^*)/(2\overline{m})$. The solution may be obtained using standard non-linear optimization methods see for instance Dixon [1972]. We are interested not in obtaining a solution for a fixed $\overline{m}$ but obtaining, using local gradient information, a sequence of solutions for the sequence of minimization problems corresponding to an increasing sequence of total sample-size values $\overline{m}(t)$.

We restate the GQ rule but now applied to a deterministic sample-size sequence with a fixed complexity k^*. Note that in this section, k^* is assumed to be known and may therefore be used for querying. The rule modifies the sample size vector $m(t)$ at time t in the direction (among all directions along the elementary vectors e_i, $1 \leq i \leq M$) of steepest descent of $\epsilon(m, k^*)$.

Greedy Query Rule (GQ) Let $\Delta > 0$ be any fixed constant. At discrete times $t = 1, 2, \ldots$, let $j^*(t) = \text{argmax}_{1 \le j \le M} \frac{p_j \epsilon(m_j(t), k_j^*)}{m_j(t)}$ and in case of more than one argmax (e.g., if all M values are identical) choose any one to be $j^*(t)$. Let

$$m_{j^*(t)}(t+1) = m_{j^*(t)}(t) + \Delta \tag{18}$$

while the remaining components of $m(t)$ remain unchanged, i.e.,

$$m_j(t+1) = m_j(t), \forall j \neq j^*(t).$$

The value of the derivative with respect to continuous time t evaluated at $t = 1, 2, \ldots$, is chosen as $\dot{m}_{j^*(t)}(t) = \Delta$ and $\dot{m}_j(t) = 0$ for $j \neq j^*(t)$. $\square$

The next lemma shows that the rule achieves the desired result, namely, the deterministic sequence $m(t)$ converges to the optimal trajectory $m^*(t)$.

Lemma 4 *For any initial point $m(0) \in \mathbb{R}^M$, satisfying $m_i(0) \geq 3$, there exists some finite integer $0 < N' < \infty$ such that for all discrete time $t > N'$ the trajectory $m(t)$ corresponding to a repeated application of the adaptation rule GQ, is no farther than Δ (in the l_1^M-norm) from the optimal trajectory $m^*(t)$.*

PROOF. Recall that $\epsilon(m, k^*) = \sum_{i=1}^M p_i \epsilon(m_i, k_i^*)$ where $\epsilon(m_i, k_i) = \sqrt{\frac{k_i \ln m_i}{m_i}}$, $1 \leq i \leq M$. The derivative $\frac{\partial \epsilon(m,k^*)}{\partial m_i} = p_i \frac{k_i^*}{2\epsilon(m_i,k_i^*)} \frac{1-\ln m_i}{m_i^2} \simeq p_i \frac{1}{2\epsilon(m_i,k_i^*)} \frac{-k_i^* \ln m_i}{m_i^2}$ which equals $-p_i \epsilon(m_i, k_i^*)/(2m_i)$. We denote by $x_i = p_i \epsilon(m_i, k_i^*)/(2m_i)$, and note that $\frac{dx_i}{dm_i} \simeq -\frac{3}{2} \frac{x_i}{m_i}$, $1 \leq i \leq M$.

There is a one-to-one correspondence between the vector x and m. Thus we may refer to the optimal trajectory also in x-space. First, let us consider the set $T = \{x = c[1, 1, \ldots, 1] \in \mathbb{R}_+^M : c \in \mathbb{R}_+\}$ which is not a trace (with parameter t) but the 'static' set corresponding to the trace of the optimal trajectory in x-space. We refer to T' as the corresponding set in m-space.

Define the Liapunov function

$$V(x(t)) = V(t) = \frac{x_{max}(t) - x_{min}(t)}{x_{min}(t)}$$

where for any vector $x \in \mathbb{R}_+^M$, $x_{max} = \max_{1 \le i \le M} x_i$, and $x_{min} = \min_{1 \le i \le M} x_i$, and write m_{max}, m_{min} for the elements of m with the same index as x_{max}, x_{min}, respectively.

Denote by $\dot{V}$ the derivative of V with respect to t. The notation $\dot{V}(x)$ denotes the derivative of V with respect to t evaluated at x. We first claim the following stability property:

Claim 1 *If $x \notin T$ then $V(x) > 0$ and $\dot{V}(x) < 0$. If $x \in T$ then $V(x) = 0$ and $\dot{V}(x) = 0$.*

We prove the claim next. For $x \notin T$ we have $x \neq c[1, 1, \ldots, 1]$, for any $c \in \mathbb{R}_+$. Thus $x_{max} - x_{min} > 0$ which implies $V(x) > 0$. While for $x \in T$, $x = c[1, 1, \ldots, 1]$ for some $c \in \mathbb{R}_+$ hence $x_{max} = x_{min}$ which implies $V(x) = 0$.

We also have

$$\dot{V} = \frac{dV}{dt} = \sum_{j=1}^{M} \frac{d}{dx_j} V \frac{dx_j}{dt}. \tag{19}$$

Now, according to Rule GQ at any time t only x_{max} changes. Thus the right side of (19) equals

$$\frac{dV}{dx_{max}} \frac{dx_{max}}{dm_{max}} \dot{m}_{max}. \tag{20}$$

According to Rule GQ, $\dot{m}_{max} = \Delta$. Also, $dV/dx_{max} = 1/x_{min}$. Thus (20) becomes

$$\begin{aligned} -\frac{3}{2} \frac{\Delta}{m_{max}} \frac{x_{max}}{x_{min}} &= -\frac{3}{2} \frac{\Delta}{m_{max}} \frac{x_{max} - x_{min}}{x_{min}} - \frac{3}{2} \frac{\Delta}{m_{max}} \\ &\leq -\frac{3}{2} \frac{\Delta}{m_{max}} \frac{x_{max} - x_{min}}{x_{min}} \\ &= -\frac{3}{2} \Delta \frac{V(x)}{m_{max}} \leq -\frac{3}{2} \frac{\Delta\, V(x)}{t\, \Delta} = -\frac{3}{2} \frac{V}{t} \end{aligned}$$

the latter follows since $m_{max} \leq \sum_{i=1}^{M} m_i(t) = \Delta\, t$ using the fact that $\dot{m}_{max} = \Delta$. Thus we now have the following differential equation

$$\dot{V} \leq -\frac{3}{2} \frac{V}{t}. \tag{21}$$

Since for $x \notin T$, $V(x(t)) > 0$ it follows that $\dot{V}(x(t)) < 0$ while for $x \in T$, $V(x) = 0$ implies $\dot{V}(x) = 0$, which together with the above proves Claim 1. $\square$

We have proved that as long as $m(t)$ is not on the optimal trajectory then $V(t)$ decreases. In order to show that the trajectory is an attractor we need

to show that $V(t)$ decreases fast enough to zero.

Solving (21) yields

$$V(t) \leq const \left(\frac{1}{t}\right)^{\frac{3}{2}}. \tag{22}$$

As we now show, this rate of decrease suffices to guarantee the convergence of $x(t)$ to the optimal trajectory. Denote by $\text{dist}(x, T) = \inf_{y \in T} \|x - y\|_{l_1^M}$, where l_1^M denotes the Euclidean vector norm.

Claim 2 *As $t \to \infty$, the distance*

$$dist(m(t), T') \to 0.$$

Fix a time t such that $V(x(t)) \leq \epsilon$. For this x we have $x_{max} - x_{min} \leq \epsilon x_{min}$. Denote by $\overline{x} = \frac{1}{M} \sum_{i=1}^{M} x_i$. Take

$$\tilde{x} = [\overline{x}, \ldots, \overline{x}] \tag{23}$$

and denote the vector corresponding to $\tilde{x}$ by $\tilde{m}$.

Then the distance $|\tilde{x}_i - x_i| \leq \epsilon x_{min}$, for every $1 \leq i \leq M$. Using the Mean Value Theorem for the function $x_i(m_i) = \epsilon(m_i, k_i^*)/m_i$, applied to the points $\tilde{m}_i$ and m_i we have for every $1 \leq i \leq M$,

$$|\tilde{m}_i - m_i| = \frac{|\tilde{x}_i - x_i|}{\frac{3x_i'}{2m_i'}} \leq \frac{\epsilon x_{min}}{\frac{3x_i'}{2m_i'}} \tag{24}$$

where x_i' corresponds to the point m_i' which satisfies $\min\{\tilde{m}_i, m_i\} \leq m_i' \leq \max\{\tilde{m}_i, m_i\}$. Now, we have

$$x_i' \geq \min\{\tilde{x}_i, x_i\} = \min\{\overline{x}, x_i\} \geq x_{min}.$$

Combining the above we have

$$|\tilde{m}_i - m_i| \leq \frac{2}{3}\epsilon m_i' \leq \frac{2}{3}\epsilon \max\{\tilde{m}_i, m_i\}. \tag{25}$$

Note the simple inequality $\max\{\tilde{m}_i, m_i\} \leq m_i + |\tilde{m} - m_i|$. This yields

$$|\tilde{m}_i - m_i| \leq \frac{\frac{2}{3}\epsilon m_i}{1 - \frac{2}{3}\epsilon}$$

which holds for any $1 \leq i \leq M$. Hence $\|\tilde{m} - m\|_{l_1^M} \leq \frac{2}{3} M \frac{\epsilon m_{max}}{1 - \frac{2}{3}\epsilon}$.

Now, choose a t such that $const(1/t)^{3/2} = \epsilon$, where the constant $const$ is from (22). For such t, we have $V(t) \leq \epsilon$ hence the above inequality applies. Moreover, $m_{max} \leq \sum_{i=1}^{M} m_i(t) = t\,\Delta$. So, making now the dependence on t explicit, we have

$$\|\tilde{m}(t) - m(t)\|_{l_1^M} \leq \frac{\frac{2}{3} M \left(\frac{1}{t}\right)^{\frac{3}{2}} t\,\Delta}{1/const - \frac{2}{3}\left(\frac{1}{t}\right)^{\frac{3}{2}}} = \frac{\frac{2M}{3\sqrt{t}}\,\Delta}{1/const - \frac{2}{3}\frac{1}{t^{\frac{3}{2}}}} \to 0 \tag{26}$$

as $t \to \infty$. We also have

$$\mathrm{dist}(m(t), T') = \inf_{y \in T'} \|m(t) - y\|_{l_1^M} \leq \|m(t) - \tilde{m}(t)\|_{l_1^M} \to 0, \quad t \to \infty$$

since $\tilde{m} \in T'$. This proves Claim 2. $\square$

So $m(t)$ gets closer to the set T' with increasing time t. Denote by the t^{th} *problem* the minimization of $\epsilon(y, k^*)$ under the constraint $\sum_{i=1}^{M} y_i = \overline{m}(t)$. Denote its solution by $m^*(t)$. We next show that $m(t)$ gets closer to $m^*(t)$ as $t \to \infty$.

Letting $\beta(t) = \|\tilde{m}(t) - m(t)\|_{l_1^M}$, then from above, $\beta(t) \to 0$ with $t \to \infty$. It follows that $\|m(t)\|_{l_1^M} - \beta(t) \leq \|\tilde{m}(t)\|_{l_1^M} \leq \|m(t)\|_{l_1^M} + \beta(t)$. Since $\overline{m}(t) = \|m(t)\|_{l_1^M}$, and denoting by $\hat{m}(t) = \|\tilde{m}(t)\|_{l_1^M}$, then it follows from (17) and (23) that $\tilde{m}(t)$ is the *solution* to the minimization of $\epsilon(y, k^*)$ under a constraint $\sum_{i=1}^{M} y_i = \hat{m}(t)$, where $|\hat{m}(t) - \overline{m}(t)| \leq \beta(t)$. By the continuity of the mapping which takes the constraint value $\overline{m}$ to the solution vector m^* it follows that the two solution vectors $\tilde{m}(t)$ and $m^*(t)$ of the two minimization problems under constraints $\sum_{i=1}^{M} y_i = \hat{m}(t)$ and $\sum_{i=1}^{M} y_i = \overline{m}(t)$, respectively, become arbitrarily close in the l_1^M-norm as $t \to \infty$. The rate of convergence of $\tilde{m}(t) \to m^*(t)$ depends on the complexity vector k^*.

Combining this with (26) it follows that as $t \to \infty$, $m(t)$ gets closer in the l_1^M-norm to the solution $m^*(t)$ of the t^{th} minimization problem. As both $m(t)$ and $m^*(t)$ are multi-integers, there is some finite discrete time N' such that $m(N') = m^*(N')$. At that point the Rule GQ will adapt, i.e., increase m, in *any* one of the component directions m_i, $1 \leq i \leq M$, since all the components of $x(N')$ are equal. This results in a step of size Δ away from the optimal trajectory followed by, at time $N' + 1$, a renewed convergence of $m(t)$ to $m^*(t)$, $t > N'$, which follows from the above analysis. Hence for all discrete time $t > N'$, $\|m(t) - m^*(t)\|_{l_1^M} \leq \Delta$. $\square$

Thus we conclude that for the case of known k^*, the Rule GQ ensures that the sample size vector $m(t)$ converges to a Δ-band around the optimal trajectory. In the next section we show that the same rule may also be used in the setting where k^* is unknown.

4.2 The Realistic Case—k^* is unknown

In the previous section we determined the convergence property of the sequence $m(t)$ generated by Rule GQ which asymptotically was shown to minimize the criterion $\epsilon(m, k^*)$ under the constraint that $\|m(t)\|_{l_1^M} = \overline{m}(t)$.

In this section we are concerned with the convergence of the *random* sequence $m(n)$ generated by Algorithm SQ, see (14), which adapts $m(n)$ to minimize a *random* criterion $\epsilon(\cdot, \hat{k}_n)$. This bears similarity to stochastic approximation under non-exogenous noise where noise depends on the state variable which is adapted at each time instance, cf. Kushner & Clark [1978]. In our case however, the random sequence $\hat{k}_n$ converges to a deterministic value k^* (see Corollary 1) thereby admitting a simpler analysis.

The next lemma states that even when k^* is unknown, it is possible, by using Algorithm SQ, to generate a sample-size vector sequence which converges to the optimal $m^*(n)$ trajectory asymptotically in time (again, the use of n instead of t just means we have a random sequence $m(n)$ and not a deterministic sequence $m(t)$ as was investigated in the previous section).

Lemma 5 *Fix any $\Delta \geq 1$ as a step size used by Algorithm SQ. Given a sample size vector sequence $m(n)$, $n \to \infty$, generated by Algorithm SQ, assume that $\hat{k}_n \to k^*$ almost surely, where k^* is the Bayes complexity as defined in (10). Let $m^*(n)$ be the optimal trajectory as in Definition 3. Then the events*

$$\|m(n) - m^*(n)\|_{l_1^M} > \Delta$$

occur infinitely often with probability 0.

PROOF.

From Lemma 3, $m(n)$ generated by Algorithm SQ is an increasing sample-size sequence. Therefore by Corollary 1 we have $\hat{k}_n \to k^*$, a.s., as $n \to \infty$. This means that $P(\exists n > N, |\hat{k}_n - k^*| > \epsilon) = \delta_N(\epsilon)$ where $\delta_N(\epsilon) \to 0$ as $N \to \infty$. Now, since $\hat{k}_n$, k^* are multi-integers there exists a small enough $\epsilon > 0$ and some large enough $N(\epsilon)$ such that for all $n > N(\epsilon)$ we have $|\hat{k}_n - k^*| \leq \epsilon$ implying $\hat{k}_n = k^*$. Combining the above, it follows that for all $\delta > 0$, there

is a finite $N(\delta,\epsilon) \in \mathbb{Z}_+$ such that with probability $1-\delta$ for all $n \geq N(\epsilon,\delta)$, $\hat{k}_n = k^*$.

It follows that with the same probability for all $n \geq N$, the criterion $\epsilon(m,\hat{k}_n) = \epsilon(m,k^*)$, uniformly over all $m \in \mathbb{Z}_+^M$, and hence the trajectory $m(n)$ taken by algorithm SQ, governed by the criterion $\epsilon(\cdot,\hat{k}_n)$, equals the trajectory $m(t)$, $t \in \mathbb{Z}_+$, taken by Rule GQ, see (18), under the deterministic criterion $\epsilon(\cdot,k^*)$. Moreover, this probability of $1-\delta$ goes to 1 as $N \to \infty$ by the a.s. convergence of $\hat{k}_n$ to k^*.

Finally, by Lemma 4, there exists a $N' < \infty$ such that for all discrete time $t > N'$, $\|m(t)-m^*(t)\|_{l_1^M} \leq \Delta$. It follows that for all $n > \max\{N,N'\}$, the random sequence $m(n)$ generated by Algorithm SQ satisfies $\|m(n)-m^*(n)\|_{l_1^M} \leq \Delta$ with probability going to 1 as $\max\{N,N'\} \to \infty$. Stated more formally: Let $N'' = \max\{N,N'\}$ then $P\left(\exists n > N'', \hat{k}_n \neq k^* \text{ or } \left\|m(t)_{|t=n} - m^*(t)_{|t=n}\right\|_{l_1^M} > \Delta\right) = \delta_{N''}$ where $\delta_{N''} \to 0$ as $N'' \to \infty$. The latter means that the event $\hat{k}_n \neq k^*$ or $\|m(n)-m^*(n)\|_{l_1^M} > \Delta$ occurs infinitely often with probability 0. The statement of the lemma then follows. $\square$

5 Conclusions

In this work we considered the problem of learning multi-category classification of M pattern classes with the assumption that the Bayes classifier has zero loss. We proposed a criterion according to which there are sample sizes m_i^*, $1 \leq i \leq M$, which minimize an upper bound on the loss of an estimator of the Bayes classifier. These sample sizes depend on the unknown intrinsic complexity of the Bayes classifier and as such cannot be computed directly. For this reason we introduced an on-line algorithm which chooses at each time instant the particular pattern class from which to draw randomly labeled examples. The choice is governed by a stochastic gradient descent rule which minimizes a random criterion and for all large enough time is shown to generate these minimizing sample sizes.

There are various possible extensions including the treatment of the case of having a Bayes loss greater than zero and trying to improve the rate of convergence of $m(n)$ to $m^*(n)$ by allowing the step size Δ to vary somehow with time n. For this, it appears though that the rate of convergence of $\hat{k}_n$ to k^* needs to be known.

Acknowledgements:
The author acknowledges Nahum Shimkin for pointing out the Liapunov function for Lemma 4, and thanks Avrim Blum and one anonymous referee for

useful comments

A Proof of Lemma 1

As in the beginning of Section 2, for $k \in \mathbb{Z}_+^M$ let $\mathcal{C}_k$ denote a class of classifiers of the form $c(x) = [c_1(x), \ldots, c_M(x)]$, $x \in \mathbb{R}^d$ where $c_i(x) \in \{0, 1\}$, $1 \leq i \leq M$, are the boolean component classifiers of c. Denote by $\mathcal{G}_k \subset \mathcal{C}_k$ the set of well-defined classifiers in $\mathcal{C}_k$.

For a class $\mathcal{B}_r$ of boolean classifiers with $VC(\mathcal{B}_r) = r$ it is known (cf. Devroye et. al. [1996] ch. 6, Vapnik [1982] Theorem 6.7) that a bound on the deviation between the loss and the empirical loss uniformly over all classifiers $b \in \mathcal{B}_r$ is

$$\sup_{b \in \mathcal{B}_r} |L(b) - L_m(b)| \leq const \sqrt{\frac{r \ln m + \ln\left(\frac{1}{\delta}\right)}{m}} \tag{A.1}$$

with probability $1 - \delta$ where m denotes the size of the random sample used for calculating empirical loss $L_m(b)$. Choosing for instance $\delta_m = \frac{1}{m^2}$ implies that the bound of $const\sqrt{\frac{r \ln m}{m}}$, with a different constant *const*, does not hold infinitely often with probability 0. We will refer to this as the uniform strong law of large numbers result. This bound was defined as $\epsilon(m, r)$ in (3).

We begin with proving the first part of the lemma.

PROOF. From above we have for arbitrary $\delta' > 0$ and for each $1 \leq i \leq M$,

$$\mathbf{P}\left(\sup_{c_i \in \mathcal{C}_{k_i}} |L_i(c_i) - L_{i,m_i}(c_i)| > const\ \epsilon(m_i, k_i, \delta')\right) \leq \delta' \tag{A.2}$$

provided m_i is larger than some finite value.

$$\begin{aligned}
&\mathbf{P}\left(\sup_{c \in \mathcal{C}_k} |L(c) - L_m(c)| > \epsilon(m, k, \delta')\right) = \\
&= \mathbf{P}\left(\sup_{c \in \mathcal{C}_k} \left|\sum_{i=1}^M p_i \left(L(c_i) - L_{i,m_i}(c_i)\right)\right| > \sum_{i=1}^M p_i \epsilon(m_i, k_i, \delta')\right) \\
&\leq \mathbf{P}\left(\sup_{c \in \mathcal{C}_k} \sum_{i=1}^M p_i \left|L(c_i) - L_{i,m_i}(c_i)\right| > \sum_{i=1}^M p_i \epsilon(m_i, k_i, \delta')\right)
\end{aligned}$$

$$
\begin{aligned}
&= \mathbf{P}\left(\exists c \in \mathcal{C}_k : \sum_{i=1}^{M} p_i \left|L(c_i) - L_{i,m_i}(c_i)\right| > \sum_{i=1}^{M} p_i \epsilon(m_i, k_i, \delta')\right) \\
&\leq \mathbf{P}\left(\exists c \in \mathcal{C}_k : \exists 1 \leq i \leq M, |L(c_i) - L_{i,m_i}(c_i)| > \epsilon(m_i, k_i, \delta')\right) \\
&\leq \sum_{i=1}^{M} \mathbf{P}\left(\exists c \in \mathcal{C}_k : |L(c_i) - L_{i,m_i}(c_i)| > \epsilon(m_i, k_i, \delta')\right) \\
&= \sum_{i=1}^{M} \mathbf{P}\left(\exists c \in \mathcal{C}_{k_i} : |L(c) - L_{i,m_i}(c)| > \epsilon(m_i, k_i, \delta')\right) \\
&\leq M\delta' \equiv \delta.
\end{aligned}
$$

We also have

$$
\sup_{c \in \mathcal{C}_k} |L(c) - L_m(c)| \leq \alpha \Rightarrow \sup_{c \in \mathcal{G}_k} |L(c) - L_m(c)| \leq \alpha
$$

since $\mathcal{G}_k \subseteq \mathcal{C}_k$. The first statement of the lemma then follows. $\square$

For the second part of the lemma, by the premise, consider any fixed complexity vector k and any sequence-generating procedure ϕ. Define the following set of sample size vector sequences: $A_N \equiv \{m(n) : n > N, m(n) \text{ is generated by } \phi\}$. As the space is discrete, for any finite N, the set A_N contains all possible paths except a finite number of length-N paths. We will show that the events $E_n \equiv \{\sup_{c \in \mathcal{G}_k} \left|L(c) - L_{m(n)}(c)\right| > \epsilon(m(n), k, \delta) : m(n) \text{ generated by } \phi\}$ occur infinitely often with probability 0, where $\epsilon(m, k, \delta)$ is defined just below (6) and choosing δ as a function of m.

Let us define $\delta^*_m = \frac{1}{\max_{1 \leq j \leq M} m_j^2}$. We write $\{\exists m(n) \in A_N : \text{property holds}\}$ to mean there exists a sequence $m(\cdot) \in A_N$ such that there exists $n > N$ such that the property holds for the point $m(n)$. We have

$$
\begin{aligned}
&\mathbf{P}\left(\exists m(n) \in A_N : \sup_{c \in \mathcal{G}_k} \left|L(c) - L_{m(n)}(c)\right| > \epsilon(m(n), k, \delta^*_{m(n)})\right) \\
&\leq \mathbf{P}\left(\exists m(n) \in A_N : \sup_{c \in \mathcal{C}_k} \left|L(c) - L_{m(n)}(c)\right| > \epsilon(m(n), k, \delta^*_{m(n)})\right) \\
&\leq \mathbf{P}\left(\exists m(n) \in A_N : \sup_{c \in \mathcal{C}_k} \sum_{j=1}^{M} p_j \left|L(c_j) - L_{j,m_j(n)}(c_j)\right| > \sum_{j=1}^{M} p_j \epsilon(m_j(n), k_j, \delta^*_{m(n)})\right) \\
&\leq \mathbf{P}\left(\exists m(n) \in A_N : \exists 1 \leq j \leq M, \sup_{c_j \in \mathcal{C}_{k_j}} \left|L(c_j) - L_{j,m_j(n)}(c_j)\right| > \epsilon(m_j(n), k_j, \delta^*_{m(n)})\right)
\end{aligned}
$$

where we used again the fact that $\mathcal{G}_k \subseteq \mathcal{C}_k$. Now, $m(n) \in A_N$ implies there exists a point m such that $\min_{1 \leq j \leq M} m_j > T_\phi(N)$ where $T_\phi(N)$ is increasing

with N. This follows from Definition 1 and from $m(n)$ being generated by ϕ which means it is an increasing sequence.

Continuing from above we have,

$$\mathbf{P}\left(\exists m(n) \in A_N : \exists 1 \leq j \leq M, \sup_{c_j \in \mathcal{C}_{k_j}} \left|L(c_j) - L_{j,m_j(n)}(c_j)\right| > \epsilon(m_j(n), k_j, \delta^*_{m(n)})\right) \tag{A.3}$$

$$\begin{aligned}
&= \mathbf{P}\left(\exists m \in \mathbb{Z}_+^M : \min_{1\leq i\leq M} m_i > T_\phi(N),\ \exists 1 \leq j \leq M, \sup_{c_j \in \mathcal{C}_{k_j}} \left|L(c_j) - L_{j,m_j}(c_j)\right| > \epsilon(m_j, k_j, \delta^*_m)\right) \\
&\leq \sum_{j=1}^{M} \mathbf{P}\left(\exists m \in \mathbb{Z}_+^M : \min_{1\leq i\leq M} m_i > T_\phi(N), \sup_{c_j \in \mathcal{C}_{k_j}} \left|L(c_j) - L_{j,m_j}(c_j)\right| > \epsilon(m_j, k_j, \delta^*_m)\right) \\
&\leq \sum_{j=1}^{M} \mathbf{P}\left(\exists m \in \mathbb{Z}_+^M : m_j > T_\phi(N), \sup_{c_j \in \mathcal{C}_{k_j}} \left|L(c_j) - L_{j,m_j}(c_j)\right| > \epsilon(m_j, k_j, \delta^*_m)\right)
\end{aligned}$$

where by going to (A.3) we have eliminated the need for n using the function $T_\phi(N)$ which depends only on the generating procedure ϕ and holds for all possible sequences generated by ϕ. By definition of δ^*_m we have,

$$\epsilon(m_j, k_j, \delta^*_m) = \sqrt{\frac{k_j \ln m_j + \ln \frac{1}{1/\max_{1\leq j\leq M} m_j^2}}{m_j}} > \sqrt{\frac{k_j \ln m_j + \ln \frac{1}{1/m_j^2}}{m_j}} = \epsilon\left(m_j, k_j, \frac{1}{m_j^2}\right)$$

for any $1 \leq j \leq M$. Continuing we have,

$$\begin{aligned}
&\sum_{j=1}^{M} \mathbf{P}\left(\exists m \in \mathbb{Z}_+^M : m_j > T_\phi(N), \sup_{c_j \in \mathcal{C}_{k_j}} \left|L(c_j) - L_{j,m_j}(c_j)\right| > \epsilon(m_j, k_j, \delta^*_m)\right) \\
&\leq \sum_{j=1}^{M} \mathbf{P}\left(\exists m \in \mathbb{Z}_+^M : m_j > T_\phi(N), \sup_{c_j \in \mathcal{C}_{k_j}} \left|L(c_j) - L_{j,m_j}(c_j)\right| > \epsilon\left(m_j, k_j, \frac{1}{m_j^2}\right)\right) \\
&= \sum_{j=1}^{M} \mathbf{P}\left(\exists m_j > T_\phi(N) : \sup_{c_j \in \mathcal{C}_{k_j}} \left|L(c_j) - L_{j,m_j}(c_j)\right| > \epsilon\left(m_j, k_j, \frac{1}{m_j^2}\right)\right) \\
&\leq \sum_{j=1}^{M} \sum_{m_j > T_\phi(N)} \frac{1}{m_j^2} \qquad (A.4) \\
&\equiv \sum_{j=1}^{M} \eta(N) \equiv s_N
\end{aligned}$$

where (A.4) follows from the uniform strong law result under (A.1). Note that the set $\{m_j : m_j > T_\phi(N)\}$ is strictly increasing, i.e., $T_\phi(N)+1, T_\phi(N)+$

$2, \ldots$, as opposed to $\{m_j(n) : m_j(n) > T_\phi(N)$ which is not necessarily strictly increasing but may have repetitions, i.e., $T_\phi(N)+1, \cdots, T_\phi(N)+1, T_\phi(N)+2, \cdots, T_\phi(N)+2, \ldots$. Having eliminated n since step (A.3) means we deal with the former set. The quantity $\eta(N)$, and hence s_N, is strictly decreasing with respect to N. We have therefore shown that

$$\mathbf{P}(\exists m(n) \in A_N : \sup_{c \in \mathcal{G}_k} \left| L(c) - L_{m(n)}(c) \right| > \epsilon(m(n), k, \delta^*_{m(n)})) \leq s_N$$

and it follows that the same holds if we replace $\epsilon(m(n), k, \delta^*_{m(n)})$ with $\epsilon(m(n), k)$ (see (3)) since there exists a constant *const* such that for all $m(n)$ and $1 \leq i \leq M$, we have $\max_{1 \leq j \leq M} m_j(n) \leq const\ m_i(n)$ based again on ϕ being a sequence generating procedure which places a lower bound on the rate of increase of $T_\phi(N)$.

So we have

$$\lim_{N \to \infty} \mathbf{P}(\exists m(n) \in A_N : \sup_{c \in \mathcal{G}_k} \left| L(c) - L_{m(n)}(c) \right| > \epsilon(m(n), k)) = 0$$

which implies that the sequence of events

$$E_n \equiv \left\{ \sup_{c \in \mathcal{G}_k} \left| L(c) - L_{m(n)}(c) \right| > \epsilon(m(n), k) \right\}, n = 1, 2, \ldots$$

where $m(n)$ is any sequence generated by ϕ, occurs infinitely often with probability 0. This proves the second part of the lemma 1. $\square$

B Proof of Lemma 2

First we prove the convergence of $\hat{k} \to k^*$, where k^* is some vector of minimal norm over all vectors k for which $L(c^*_k) = 0$. We henceforth denote for a vector $k \in \mathbb{Z}^M_+$, by $\|k\|_\infty = \max_{1 \leq i \leq M} |k_i|$. Throughout the proof all sequences and convergence statements are made with respect to the increasing sequence $m(n)$. The indexing variable n is sometimes left hidden for simpler notation.

The set $\hat{F}$ defined in (11) may be rewritten as $\hat{F} = \{k : \tilde{L}(\hat{c}_k) = \tilde{L}(\hat{c}^*)\}$, i.e., it is the set of complexities corresponding to all empirical loss minimizers whose penalized loss is the minimum over all $k \in \mathbb{Z}^M_+$. The cardinality of $\hat{F}$ is finite since for all k having at least one component k_i larger than some constant implies $\tilde{L}(\hat{c}_k) > \tilde{L}(\hat{c}^*)$ because $\epsilon(m, k)$ will be larger than $\tilde{L}(\hat{c}^*)$. This implies that the set of k for which $\tilde{L}(\hat{c}_k) \leq \tilde{L}(\hat{c}^*)$ is finite. Now for any $\alpha > 0$, define $\hat{F}_\alpha = \{k : \tilde{L}(\hat{c}_k) \leq \tilde{L}(\hat{c}^*) + \alpha\}$. We recall F^*, which was defined in (9) as

$F^* = \{k : L(c_k^*) = L^* = 0\}$, and define $F_\alpha^* = \{k : L(c_k^*) \leq L^* + \alpha\}$, where the Bayes loss is $L^* = 0$. Recall that the chosen classifier $\hat{c}^*$ has a complexity $\hat{k} = \text{argmin}_{k \in \hat{F}} \|k\|_\infty$. By Assumption 1 there exists a $k^* = \text{argmin}_{k \in F^*} \|k\|_\infty$ all of whose components are finite.

We start with the following Claim.

Claim 3 *$\hat{F} \not\subseteq F^*_{\epsilon(m,k^*)}$, i.o. with probability 0*

where *i.o.* stands for infinitely often.

PROOF.

$$\begin{aligned}
&\mathbf{P}\left(L(c_k^*) > \epsilon(m,k^*) \ \ i.o.\right) \\
\leq &\mathbf{P}\left(L(\hat{c}_k) > \epsilon(m,k^*) \ \ i.o.\right) && \text{(B.1)} \\
= &\mathbf{P}\left(L(\hat{c}_k) > L_m(\hat{c}_{k^*}) + \epsilon(m,k^*) \ \ i.o.\right) && \text{(B.2)}
\end{aligned}$$

where (B.1) follows since $L(\hat{c}_k) \geq L(c_k^*)$, (B.2) follows by Assumption 1 which by $L(c_{k^*}^*) = L^* = 0$ implies that $L_m(c_{k^*}^*) = 0$ for any sample size vector m and by definition of an empirical loss minimizer

$$L_m(\hat{c}_{k^*}) \leq L_m(c_{k^*}^*) = 0. \tag{B.3}$$

We continue from (B.2). For any $k \in \hat{F}$

$$\begin{aligned}
&\mathbf{P}\left(L(\hat{c}_k) > L_m(\hat{c}_{k^*}) + \epsilon(m,k^*) \ \ i.o.\right) \\
= &\mathbf{P}\left(L(\hat{c}_k) > \tilde{L}(\hat{c}_{k^*}) \ \ i.o.\right) \\
\leq &\mathbf{P}\left(L(\hat{c}_k) > \tilde{L}(\hat{c}^*) \ \ i.o.\right) && \text{(B.4)} \\
= &\mathbf{P}\left(L(\hat{c}_k) > \tilde{L}(\hat{c}_k) \ \ i.o.\right) && \text{(B.5)} \\
= &\mathbf{P}\left(L(\hat{c}_k) > L_m(\hat{c}_k) + \epsilon(m,k) \ \ i.o.\right) = 0 && \text{(B.6)}
\end{aligned}$$

where (B.4) follows from the definition of $\hat{c}^*$, (B.5) follows from by definition of $\hat{F}$ and (B.6) follows from Lemma 1.

As the cardinality of $\hat{F}$ is finite, it follows that $L(c_k^*) > \epsilon(m,k^*)$, i.o., with probability 0 simultaneously for all $k \in \hat{F}$. Hence $\hat{F} \not\subseteq F^*_{\epsilon(m,k^*)}$, i.o. with probability 0. □

Claim 4 *$k^* \in \hat{F}$.*

PROOF. For any $\alpha > 0$, we have $k^* \in \hat{F}_\alpha$. To see this first let m be large enough such that $\epsilon(m, k^*) = \alpha$. Then it follows that $k^* \in \hat{F}_{\epsilon(m,k^*)}$ since $\tilde{L}(\hat{c}_{k^*}) = L_m(\hat{c}_{k^*}) + \epsilon(m, k^*) \leq \epsilon(m, k^*) \leq \tilde{L}(\hat{c}^*) + \epsilon(m, k^*)$, where we used again the fact that $L_m(\hat{c}_{k^*}) \leq L_m(c^*_{k^*}) = 0$. It follows that $k^* \in \lim_{\alpha \to 0} \hat{F}_\alpha = \hat{F}$. $\square$

We now claim the following:

Claim 5 *For all m large enough, $k^* = argmin_{k \in F^*_{\epsilon(m,k^*)}} \|k\|_\infty$.*

PROOF.

There exists a $\tilde{k} = [l, \ldots, l]$ such that the cube $C = \{k : k_i \leq l, 1 \leq i \leq M\}$ contains the vector k^*, and there exists a $\beta > 0$, such that for all $0 < \alpha < \beta$, $F^*_\alpha \bigcap C = F^* \bigcap C$. To see this, note that for any $\alpha > 0$, the set $F^*_\alpha \bigcap C$ contains a finite number of vectors $k \in \mathbb{Z}^M_+$. Each of these vectors corresponds to a certain classifier c^*_k with a certain loss $L(c^*_k)$ and there must be at least one such vector with a corresponding classifier having a loss of zero since $k^* \in C$. There clearly exists a $\beta > 0$ small enough such that only those $k \in F^*_\alpha \bigcap C$ for which $L(c^*_k) = 0$ satisfy $L(c^*_k) \leq \beta$. It follows that for all $0 < \alpha < \beta$, $F^*_\alpha \bigcap C = F^*_\beta \bigcap C = \{k : L(c^*_k) = 0, k \in C\} = F^* \bigcap C$ as claimed.

We continue with the proof of Claim 5 assuming that $0 < \alpha < \beta$. For every $k \in F^*_\alpha \setminus C$ there exists a component $k_i > l$ for some $i \in \{1, 2, \ldots, M\}$. Hence $\|k\|_\infty > l$. Moreover, since $k^* \in C$ then $k^*_i \leq l$, $1 \leq i \leq M$. Hence it follows that $\|k\|_\infty \geq \|k^*\|_\infty$ for all $k \in F^*_\alpha \setminus C$. Hence the $k \in F^*_\alpha$ which has minimal norm must be a k which has a minimum norm over $F^*_\alpha \bigcap C$. But the latter is equivalent to the set $F^* \bigcap C$. Now, since $k^* = \text{argmin}_{k \in F^*} \|k\|_\infty$ and since $k^* \in C$ it follows that $k^* = \text{argmin}_{F^* \bigcap C} \|k\|_\infty$. Hence it follows that k^* minimizes $\|k\|_\infty$ over all $k \in F^*_\alpha$. Letting $\alpha = \epsilon(m, k^*)$, then for all large enough m, $\alpha < \beta$ which proves the statement of Claim 5. $\square$

From Claims 3, 4 and 5 it follows that $k^* \neq \text{argmin}_{k \in \hat{F}} \|k\|_\infty$, i.o. with probability 0. And since by definition $\hat{k} = \text{argmin}_{k \in \hat{F}} \|k\|_\infty$ then it follows that

$$\|\hat{k}\|_\infty \neq \|k^*\|_\infty \;\; i.o. \tag{B.7}$$

with probability zero but where $\hat{k}$ does not necessarily equal k^*. The latter combined with Claim 3 implies that $\hat{k} \neq \text{argmin}_{k \in F^*_{\epsilon(m,k^*)}} \|k\|_\infty$, i.o. with probability 0. Finally, we have $\hat{k} \notin C$ i.o. with probability 0 since $\|\hat{k}\|_\infty = \|k^*\|_\infty$ hence it follows from the proof of Claim 5 that the event that $\hat{k}$ does not minimize $\|k\|_\infty$ over all $k \in F^* \bigcap C$ and hence over F^* happens infinitely often with probability 0.

So we have proved that any sequence of vectors $\hat{k} = \text{argmin}_{k \in \hat{F}} \|k\|_\infty$ does not minimize $\|k\|_\infty$ over all $k \in F^*$ infinitely often with probability 0. Therefore we conclude that

$$\hat{k} \to k^*, \quad \text{(componentwise) } a.s., \quad m \to \infty$$

(or equivalently, with $n \to \infty$ as the sequence $m(n)$ is increasing) where $k^* = \text{argmin}_{k \in F^*} \|k\|_\infty$, is not necessarily unique, but all of whose components are finite. This proves the first part of the lemma. $\square$

Next, we prove the second part of the lemma which states an upper bound on $L(\hat{c}^*)$. We make use of the same idea as in the proof of Lemma 1 where we start off with sequences of n and then eliminate the dependence on n. We explicitly denote the dependence of $\hat{c}^*$ on n by writing $\hat{c}^*_n$. Let ϕ be any sequence-generating procedure and define the following set of sample size vector sequences: $A_N \equiv \{m(n) : n > N, m(n) \text{ is generated by } \phi\}$. As before, we write $\{\exists m(n) \in A_N : \text{property holds}\}$ to mean there exists a sequence $m(\cdot) \in A_N$ such that there exists an $n > N$ such that the property holds for the point $m(n)$. We have

$$\begin{aligned}
&\mathbf{P}\left(\exists m(n) \in A_N : L(\hat{c}^*_n) > \epsilon(m(n), k^*) \ \right) && \text{(B.8)}\\
&= \mathbf{P}\left(\exists m(n) \in A_N : L(\hat{c}^*_n) > L_{m(n)}(\hat{c}_{k^*}) + \epsilon(m(n), k^*) \ \right) && \text{(B.9)}\\
&= \mathbf{P}\left(\exists m(n) \in A_N : L(\hat{c}^*_n) > \tilde{L}(\hat{c}_{k^*}) \ \right)\\
&\leq \mathbf{P}\left(\exists m(n) \in A_N : L(\hat{c}^*_n) > \tilde{L}(\hat{c}^*_n) \ \right) && \text{(B.10)}\\
&= \mathbf{P}\left(\exists m(n) \in A_N : L(\hat{c}^*_n) > L_{m(n)}(\hat{c}^*_n) + \epsilon(m(n), \hat{k}_n) \ \right)
\end{aligned}$$

where (B.9) follows from (B.3) and (B.10) follows from the definition of $\hat{c}^*$. Now, for any fixed n, based on the randomly drawn sample of size vector $m(n)$ the SRM-chosen classifier $\hat{c}^*_n$ could be any one of $\hat{c}_k$ in a set which is no larger than $\left\{k \in \mathbb{Z}_+^M\right\}$. We therefore have

$$\begin{aligned}
&\mathbf{P}\left(\exists m(n) \in A_N : L(\hat{c}^*_n) > L_{m(n)}(\hat{c}^*_n) + \epsilon(m(n), \hat{k}_n)\right)\\
&\leq \mathbf{P}\left(\exists m(n) \in A_N, \exists k \in \mathbb{Z}_+^M : L(\hat{c}_k) > L_{m(n)}(\hat{c}_k) + \epsilon(m(n), k)\right)\\
&\leq \mathbf{P}\left(\exists m(n) \in A_N, \exists k \in \mathbb{Z}_+^M : \exists 1 \leq j \leq M, L(\hat{c}_{k_j}) > L_{j,m_j(n)}(\hat{c}_{k_j}) + \epsilon(m_j(n), k_j)\right)\\
&\leq \mathbf{P}\left(\exists m(n) \in A_N, \exists 1 \leq j \leq M, \exists k_j \in \mathbb{Z}_+ : L(\hat{c}_{k_j}) > L_{j,m_j(n)}(\hat{c}_{k_j}) + \epsilon(m_j(n), k_j)\right).
\end{aligned}$$

Now, we eliminate n using the same reasoning as in the proof of Lemma 1. We have

$$
\begin{aligned}
&\mathbf{P}\left(\exists m(n) \in A_N,\, \exists 1 \le j \le M,\, \exists k_j \in \mathbb{Z}_+ : L(\hat{c}_{k_j}) > L_{j,m_j(n)}(\hat{c}_{k_j}) + \epsilon(m_j(n), k_j)\right) \\
&\le \mathbf{P}\left(\exists m \in \mathbb{Z}_+^M,\, \min_{1\le i\le M} m_i > T_\phi(N),\, \exists 1 \le j \le M,\, \exists k_j \in \mathbb{Z}_+ : L(\hat{c}_{k_j}) > L_{j,m_j}(\hat{c}_{k_j}) + \epsilon(m_j, k_j)\right) \\
&\le \sum_{j=1}^{M} \mathbf{P}\left(\exists m \in \mathbb{Z}_+^M,\, m_j > T_\phi(N),\, \exists k_j \in \mathbb{Z}_+ : L(\hat{c}_{k_j}) > L_{j,m_j}(\hat{c}_{k_j}) + \epsilon(m_j, k_j)\right) \\
&\le \sum_{j=1}^{M} \mathbf{P}\left(\exists m_j > T_\phi(N),\, \exists k_j \in \mathbb{Z}_+ : L(\hat{c}_{k_j}) > L_{j,m_j}(\hat{c}_{k_j}) + \epsilon(m_j, k_j)\right) \\
&\le \sum_{j=1}^{M} \sum_{k_j=1}^{\infty} \mathbf{P}\left(\exists m_j > T_\phi(N) : L(\hat{c}_{k_j}) > L_{j,m_j}(\hat{c}_{k_j}) + \epsilon(m_j, k_j)\right). \qquad \text{(B.11)}
\end{aligned}
$$

We now make use of the uniform strong law result mentioned under (A.1), just stating it more explicitly. First, let us choose a constant *const* to be the maximum of $\sqrt{6}$ and the constant in (A.1). This means $const\sqrt{\frac{k_j \ln m_j}{m_j}} \ge \sqrt{3}\sqrt{\frac{k_j \ln(em_j)}{m_j}}$, for all $m_j \ge 3$ and henceforth define $\epsilon(m_j, k_j)$ to be $const\sqrt{\frac{k_j \ln m_j}{m_j}}$ with the new *const*. Using the upper bound on the growth function cf. Vapnik [1982] Section 6.9, Devroye et. al. [1996] Theorem 13.3 we have for some absolute constant $\kappa > 0$

$$
\begin{aligned}
&\mathbf{P}\left(L(\hat{c}_{k_j}) > L_{j,m_j}(\hat{c}_{k_j}) + \epsilon(m_j, k_j)\right) \qquad \text{(B.12)} \\
&\le \kappa m_j^{k_j} e^{-m_j \epsilon^2(m_j, k_j)} \\
&= \kappa m_j^{k_j} e^{-m_j (const)^2 k_j \ln m_j / m_j} \\
&\le \kappa m_j^{k_j} e^{-3 m_j k_j \ln(em_j)/m_j} \\
&= \kappa \frac{1}{m_j^{2k_j}} e^{-3k_j} \\
&\le \kappa \frac{1}{m_j^2} e^{-3k_j} \qquad \text{for } k_j \ge 1.
\end{aligned}
$$

Continuing to upper bound (B.11)

$$
\begin{aligned}
&\sum_{j=1}^{M} \sum_{k_j=1}^{\infty} \mathbf{P}\left(\exists m_j > T_\phi(N) : L(\hat{c}_{k_j}) > L_{j,m_j}(\hat{c}_{k_j}) + \epsilon(m_j, k_j)\right) \qquad \text{(B.13)} \\
&\qquad \le \kappa \sum_{j=1}^{M} \sum_{m_j > T_\phi(N)} \sum_{k_j=1}^{\infty} \frac{e^{-3k_j}}{m_j^2} \\
&\qquad \le 2\kappa \sum_{j=1}^{M} \sum_{m_j > T_\phi(N)} \frac{1}{m_j^2} \equiv \sigma_N.
\end{aligned}
$$

Just as (A.4) was shown to be strictly decreasing with N, the same holds here

for σ_N. It follows that

$$\lim_{N\to\infty} \mathbf{P}\left(\exists m(n) \in A_N : L(\hat{c}_n^*) > \epsilon(m(n), k^*)\right) = 0$$

implying that the events $\{L(\hat{c}_n^*) > \epsilon(m(n), k^*)\}$ occur infinitely often with probability 0. The second part of the Lemma is proved. □

C Proof of Corollary 1

The proof of Lemma 2 uses only the result of Lemma 1, namely, the uniform upper bound on the deviations between the empirical and the true loss of well-defined classifiers. From the proof of Lemma 1 it is apparent that the same uniform upper bound holds even if the sample is i.i.d. *only* when conditioned on a pattern class. This follows since the upper bound is the weighted average of the uniform upper bounds on the SLLN deviations of the individual subsamples corresponding to each of the pattern classes.

Also, by premise of the corollary, the components of the vector $m(n)$ all increase with time $n \to \infty$. Thus $\hat{k}_n$ corresponds to a sequence of complexities of chosen classifiers based on an increasing sample size sequence $m(n)$. The proof of Lemma 2 applies also for the setting of Corollary 1. □

D Proof of Lemma 3

Note that for this proof we cannot use Lemma 1 or parts of Lemma 2 since they are conditioned on having a sequence-generating procedure. Our approach here relies on the characteristics of the SRM-selected complexity $\hat{k}_n$ which is shown to be bounded uniformly over n based on Assumption 1. It follows that by the sample-size increment rule of Algorithm SQ the generated sample size sequence $m(n)$ is not only increasing but with a minimum rate of increase as in Definition 2. This establishes that Algorithm SQ is a sequence-generating procedure.

PROOF.

We start with the next claim.

Claim 6 *Consider an increasing sequence $m(n)$ as in Definition 1 For all n there is some constant $0 < \rho < \infty$ such that $\|\hat{k}_n\|_\infty < \rho$.*

Suppose that there does not exist a ρ such that for all n, $\|\hat{k}_n\|_\infty < \rho$. This implies the existence of some $1 \leq i \leq M$ such that for all $\rho > 0$ there exists $N(\rho)$, $\forall n > N(\rho)$, $\hat{k}_{n,i} > \rho$ where $\hat{k}_{n,i}$ denotes the i^{th} component of $\hat{k}_n$. By Assumption 1 there exists k^* with $L(\hat{c}_{k^*}) = 0$. This implies $L_m(\hat{c}_{k^*}) = 0$ which implies $L_{i,m_i}(\hat{c}_{\hat{k}_i^*}) = 0$ where by the same assumption, the complexity $\|k^*\| < \infty$ hence $k_i^* < \infty$, for all $1 \leq i \leq M$. The S-Step of the algorithm minimizes $\tilde{L}(\hat{c}_k)$ over $k \in \mathbb{Z}_+^M$ and $\tilde{L}(\hat{c}_{\hat{k}_n}) \leq \tilde{L}(\hat{c}_k)$ for all $k \in \mathbb{Z}_+^M$ by definition of $\hat{k}_n$. In particular, it is true for a k which equals $\hat{k}_n$ in all but the i^{th} component in which it takes the value k_i^*. So we have

$$\begin{aligned}
&\sum_{j \neq i} p_j \left(L_{j,m_j(n)}(\hat{c}_{\hat{k}_{n,j}}) + \epsilon(m_j(n), \hat{k}_{n,j}) \right) + p_i \left(L_{i,m_i(n)}(\hat{c}_{\hat{k}_{n,i}}) + \epsilon(m_i(n), \hat{k}_{n,i}) \right) \\
&\leq \sum_{j \neq i} p_j \left(L_{j,m_j(n)}(\hat{c}_{\hat{k}_{n,j}}) + \epsilon(m_j(n), \hat{k}_{n,j}) \right) + p_i \left(L_{i,m_i(n)}(\hat{c}_{k_i^*}) + \epsilon(m_i(n), k_i^*) \right)
\end{aligned}$$

true for all n. We have therefore

$$\begin{aligned}
&L_{i,m_i(n)}(\hat{c}_{\hat{k}_{n,i}}) + \epsilon(m_i(n), \hat{k}_{n,i}) \\
&\leq L_{i,m_i(n)}(\hat{c}_{k_i^*}) + \epsilon(m_i(n), k_i^*) = 0 + \epsilon(m_i(n), k_i^*).
\end{aligned} \tag{D.1}$$

But by the premise, $\hat{k}_{n,i}$ is increasing with n hence there exists some N' such that $\hat{k}_{n,i} \geq k_i^*$ and hence $L_{i,m_i(n)}(\hat{c}_{\hat{k}_{n,i}}) = 0$, for all $n > N'$ where we used Assumption 1. Combining with (D.1), for all $n > N'$ we have $\epsilon(m_i(n), \hat{k}_{n,i}) \leq \epsilon(m_i(n), k_i^*)$ which implies $\hat{k}_{n,i} \leq k_i^*$. This contradicts the premise of having $\hat{k}_{n,i}$ increasing forever and hence proves the claim. $\square$

It follows that for all n, $\hat{k}_n$ is bounded by a finite constant independent of n. So for a sequence generated by the GQ criterion, $p_j \frac{\epsilon(m_j(n), \hat{k}_{n,j})}{m_j(n)}$ are bounded by $p_j \frac{\epsilon(m_j(n), \tilde{k}_j)}{m_j(n)}$, for some finite $\tilde{k}_j$, $1 \leq j \leq M$, respectively. It can be shown by simple analysis of the function $\epsilon(m, k)$ that for a fixed k the quantity $\frac{\partial^2 \epsilon(m_j, k_j)}{\partial m_j^2} / \frac{\partial^2 \epsilon(m_i, k_i)}{\partial m_i^2}$ converges to a constant dependent on k_i and k_j with increasing m_i, m_j. Hence even for the worst case $\hat{k}_n$, which still must be bounded by the above claim, it follows that the adaptation step of Procedure GQ, which always increases one of the sub-samples, amounts to increments of Δm_i and Δm_j that are no farther apart than a constant multiple of each other for all n, for any pair $1 \leq i, j \leq M$.

Hence for a sequence $m(n)$ generated by Algorithm SQ the following is satisfied: it is increasing in the sense of Definition 1, namely, for all $N > 0$ there exists a $T_\phi(N)$ such that for all $n > N$ every component $m_j(n) > T_\phi(N)$, $1 \leq j \leq M$. Furthermore, its rate of increase is bounded from below, namely, there

exists a $const > 0$ such that for all $N, N' > 0$ satisfying $T_\phi(N') = T_\phi(N) + 1$, then $|N' - N| \leq const$. It follows that Algorithm SQ is a sequence-generating procedure according to Definition 2. $\square$

E Proof of Theorem 1

The classifier $\hat{c}_n^*$ is chosen according to (13) based on a sample of size vector $m(n)$ generated by Algorithm SQ which is a sequence-generating procedure (see Lemma 3). Then from Corollary 1

$$L(\hat{c}_n^*) > const\ \epsilon(m(n), k^*), \quad i.o.$$

with probability 0 and furthermore since $\Delta = 1$ then from Lemma 5 it follows that $\|m(n) - m^*(n)\|_{l_1^M} > 1$ infinitely often with probability 0 where $m^*(n) = \text{argmin}_{m:\|m\|=\overline{m}(n)} \epsilon(m, k^*)$. $\square$

References

Angluin D., (1988), Queries and Concept Learning, *Machine Learning*, Vol 2, p. 319-342.

Barron A. R., (1994), Approximation and Estimation Bounds for Artificial Neural Networks. *Machine Learning*, vol. 14, pp. 115-133.

Anthony M., Bartlett P. L., (1999), "Neural Network Learning:Theoretical Foundations", Cambridge University Press, UK.

Bartlett P. L., Boucheron S., Lugosi G., (2002) Model Selection and Error Estimation, *Machine Learning*, Vol. 48(1-3), p. 85-113.

A. Blumer, A. Ehrenfeucht, D. Haussler, M. Warmuth. (1989). Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the ACM*. Vol. 36. No. 4. pp. 929-965.

Buescher K. L., Kumar P. R. (1996). Learning by Canonical Smooth Estimation, Part I: Simultaneous Estimation, *IEEE Trans. on Automatic Control*, Vol. 41 n. 4, p.545.

Cohn D., (1996), Neural Network Exploitation Using Optimal Experiment Design. *Neural Networks*, vol. 9, No. 6, (p.1071-1083).

Cohn D., Atlas L., Ladner R. (1994), Improving Generalization with Active Learning. *Machine Learning*, Vol 15, p.201-221.

Dixon L. C. (1972). "Nonlinear Optimization", English Universities Press, London.

Devroye L., Gyorfi L. Lugosi G. (1996). "A Probabilistic Theory of Pattern Recognition", Springer Verlag.

Duda R. O., Hart P. E., Stork D. (2001). "Pattern Classification", Second Ed., John Wiley & Sons, New York.

Freund Y., Schapire R. E., (1995), A decision-theoretic generalization of on-line learning and an application to boosting, *Journal Comp. Sys. Sci.*, Vol. 55(1), p.119-139.

Fukunaga K. (1972). "Introduction to Statistical Pattern Recognition", Academic Press, New York.

Geman S., Bienestock E., Doursat R. (1992). Neural Networks and the bias/variance dilemma. *Neural Computation*, Vol. 4, p.1-58.

Haussler D., (1992), Decision theoretic generalizations of the PAC model for neural net and other learning applications, *Inform. Comput.*, vol. 100 no. 1, pp. 78-150.

Japkowicz N. (2000), *Proc. AAAI'2000 Workshop on Learning from Imbalanced Data Sets*, Technical Report WS-00-05, AAAI Press.

Kendall M. G., Stuart A. (1994). "The Advanced Theory of Statistics", Third Ed., Griffin, London.

Kulkarni S. R., Mitter S. K., Tsitsiklis J. N., (1993). Active Learning Using Arbitrary Valued Queries. *Machine Learning*, Vol 11, p.23-35.

Kultchinskii V., (2001), Rademacher Penalties and Structural Risk Minimization, *IEEE Trans. on Info. Theory*, 47(5): pp.1902-1914.

Kushner J. H., Clark, D. S., (1978), "Stochastic Approximation Methods for Constrained and Unconstrained Systems", Applied Mathematical Sciences #26, Springer-Verlag, N.Y.

Linhart H., Zucchini W., (1986), "Model Selection". Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, N.Y.

Lugosi G., Nobel A., (1999), Adaptive Model Selection Using Empirical Complexities. Annals of Statistics vol. 27(6), 1830-1864.

Lugosi G., Zeger K., (1996), Concept Learning Using Complexity Regularization. *IEEE Trans. on Info. Theory*, Vol. 42, p.48-54.

Meir R., (1994), Bias, Variance and the Combination of Least-Squares Estimators, in "Advances in Neural Information Processing systems 7", Eds. Tesauro G., Touretzky D. and Leen T, MIT Press.

Niyogi P., (1995). Free to Choose: Investigating the Sample Complexity of Active Learning of Real Valued Functions. *Proceedings of 12th International Conference on Machine Learning*. p.405-412, Morgan Kaufmann.

Ratsaby J., Meir R., Maiorov V., (1996), Towards Robust Model Selection using Estimation and Approximation Error Bounds, *Proc.* 9^{th} *Annual Conference on Computational Learning Theory*, p.57, ACM, New York N.Y..

Ratsaby J., (1998), Incremental Learning with Sample Queries, *IEEE Trans. on PAMI*, Vol. 20, No. 8, pp.883-888.

Ripley, B. D., (1996), "Pattern Recognition and Neural Networks". Cambridge University Press.

Rivest R. L., Eisenberg B., (1990), On the sample complexity of pac-learning using random and chosen examples. *Proceedings of the 1990 Workshop on Computational Learning Theory*, p. 154-162, Morgan Kaufmann, San Maeto,

CA.

Shawe-Taylor J., Bartlett P., Williamson R.C., Anthony M., (1998) Structural Risk Minimization over Data-Dependent Hierarchies, *IEEE Trans. Inf. Theory*, 44 (5), 1926-1940.

Valiant L. G., (1984), A Theory of the learnable, *Comm. ACM* 27:11, p. 1134-1142.

Vapnik V.N., (1982), "Estimation of Dependences Based on Empirical Data", Springer-Verlag, Berlin.

Vapnik V. N and Chervonenkis A. Ya. (1981). Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theoret. Probl. and Its Appl.*, Vol. 26, 3, p.532-553.