

Stable Transductive Learning

Ran El-Yaniv and Dmitry Pechyony*

Computer Science Department
Technion - Israel Institute of Technology
{rani,pechyony}@cs.technion.ac.il

Abstract. We develop a new error bound for transductive learning algorithms. The slack term in the new bound is a function of a relaxed notion of *transductive stability*, which measures the sensitivity of the algorithm to most pairwise exchanges of training and test set points. Our bound is based on a novel concentration inequality for symmetric functions of permutations. We also present a simple sampling technique that can estimate, with high probability, the weak stability of transductive learning algorithms with respect to a given dataset. We demonstrate the usefulness of our estimation technique on a well known transductive learning algorithm.

1 Introduction

Unlike supervised or semi-supervised *inductive* learning models, in *transduction* the learning algorithm is not required to generate a general hypothesis that can predict the label of any unobserved point. It is only required to predict the labels of a given test set of points, provided to the learner before training. At the outset, it may appear that this learning framework should be “easier” in some sense than induction. Since its introduction by Vapnik more than 20 years ago [18], the theory of transductive learning has not advanced much despite the growing attention it has been receiving in the past few years.

We consider Vapnik’s *distribution-free* transductive setting where the learner is given an “individual sample” of $m+u$ unlabeled points in some space and then receives the labels of points in an m -subset that is chosen uniformly at random from the $m+u$ points. The goal of the learner is to label the remaining *test set* of u unlabeled points as accurately as possible. *Our* goal is to identify learning principles and algorithms that will guarantee small as possible error in this game. As shown in [19], error bounds for learning algorithms in this distribution-free setting apply to a more popular *distributional* transductive setting where both the labeled sample of m points and the test set of u points are sampled i.i.d. from some unknown distribution.

Here we present novel transductive error bounds that are based on new notions of *transductive stability*. The *uniform stability* of a transductive algorithm

* Supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

is its worst case sensitivity for an exchange of two points, one from the labeled training set and one from the test set. Our uniform stability result is a rather straightforward adaptation of the results of Bousquet and Elisseeff [4] for inductive learning. Unfortunately, our empirical evaluation of this new bound (that will be presented elsewhere) indicates that it is of little practical merit because the required stability rates, which enable a non-vacuous bound, are not met by useful transductive algorithms.

We, therefore, follow the approach taken by Kutin and Niyogi [12] in induction and define a notion of *weak transductive stability* that requires overall stability ‘almost everywhere’ but still allows the algorithm to be sensitive to some fraction of the possible input exchanges. To utilize this weak transductive stability we develop a novel concentration inequality for symmetric functions of permutations based on Azuma’s martingale bound. We show that for sufficiently stable algorithms, their empirical error is concentrated near their transductive error and the slack term is a function of their weak stability parameters. The resulting error bound is potentially applicable to any transductive algorithm.

To apply our transductive error bound to a specific algorithm, it is necessary to know a bound on the weak stability of the algorithm. To this end, we develop a data-dependent estimation technique based on sampling that provides high probability estimates of the algorithm’s weak stability parameters. We apply this routine on the algorithm of [20].

2 Related Work

The transductive learning framework was proposed by Vapnik [18, 19]. Two transductive settings, distribution-free and distributional, are considered and it is shown that error bounds for the distribution-free setting imply the same bounds in the distributional case. Vapnik also presented general bounds for transductive algorithms in the distribution-free setting. Observing that any hypothesis space is effectively finite in transduction, the Vapnik bounds are similar to VC bounds for finite hypothesis spaces but they are implicit in the sense that tail probabilities are not estimated but are specified in the bound as the outcome of some computational routine. Vapnik bounds can be refined to include prior “beliefs” as noted in [5]. Similar implicit but somewhat tighter bounds were developed in [3]. Explicit general bounds of a similar form as well as PAC-Bayesian bounds for transduction were presented in [5].

Exponential concentration bounds in terms of *uniform stability* were first considered by Bousquet and Elisseeff [4] in the context of induction. Quite a few variations of the inductive stability concept were defined and studied in [4, 12, 15]. It is not clear, however, what is the precise relation between these definitions and the associated error bounds. It is noted in [9, 15] that many important learning algorithms (e.g., SVM) are not stable under any of the stability definitions, including the significantly relaxed notion of weak stability introduced by Kutin and Niyogi [11, 12]. Hush et al. [9] attempted to remedy this by con-

sidering ‘graphical algorithms’ and a new geometrical stability definition, which captures a modified SVM (see also [4]).

Stability was first considered in the context of transductive learning by Belkin et al. [2]. There the authors applied uniform inductive stability notions and results of [4] to a specific graph-based transductive learning algorithm.¹ However, the algorithm considered has the deficiency that it always labels half of the points by ‘-1’ and the other half by ‘+1’.

We present general bounds for transduction based on particularly designed definitions of transductive stability, which we believe are better suited for capturing practical algorithms. Our weak stability bounds have relatively “standard” form of empirical error plus a slack term (unlike most weak stability bounds for induction [12, 15, 16]). Kearns and Ron [10] were the first to develop standard risk bounds based on weak stability. Their bounds are “polynomial”, depending on $1/\delta$, unlike the “exponential” bounds we develop here (depending on $\ln 1/\delta$).

3 Problem Setup and Preliminaries

We consider the following transductive setting [18]. A *full sample* $X_{m+u} = \{x_i\}_{i=1}^{m+u}$ consisting of $m+u$ unlabeled examples in some space \mathcal{X} is given. For each point $x_j \in X_{m+u}$, let $y_j \in \{\pm 1\}$ be its unknown deterministic label. A *training set* S_m consisting of m labeled points is generated as follows. Sample a subset of m points $X_m \subset X_{m+u}$ uniformly at random from all m -subsets of the full sample. For each point $x_i \in X_m$, obtain its uniquely determined label y_i from the teacher. Then, $S_m = (X_m, Y_m) = (z_i = \langle x_i, y_i \rangle)_{i=1}^m$. The set of remaining u (unlabeled) points $X_u = X_{m+u} \setminus X_m$ is called the *test set*. We use the notation I_r^s for the set of (indices) $\{r, \dots, s\}$ (for integers $r < s$). For simplicity we abuse notation, and unless otherwise stated, the indices I_1^m are reserved for training set points and the indices I_{m+1}^{m+u} for test set points.

The goal of the transductive learning algorithm \mathcal{A} is to utilize both the labeled training points S_m and the unlabeled test points X_u and generate a *soft classification* $\mathcal{A}_{S_m, X_u}(x_i) \in [-1, 1]$ for each (test) point x_i so as to minimize its *transductive error* with respect to some loss function ℓ , $R_u(\mathcal{A}) \triangleq R_u(\mathcal{A}_{S_m, X_u}) \triangleq \frac{1}{u} \sum_{i=m+1}^{m+u} \ell(\mathcal{A}_{S_m, X_u}(x_i), y_i)$. We consider the standard 0/1-loss and margin-loss functions denoted by ℓ and ℓ_γ , respectively.² In applications of the 0/1 loss function we always apply the sign function on the soft classification $\mathcal{A}_{S_m, X_u}(x)$. The *empirical error* of \mathcal{A} is $\hat{R}_m(\mathcal{A}) \triangleq \hat{R}_m(\mathcal{A}_{S_m, X_u}) \triangleq \frac{1}{m} \sum_{i=1}^m \ell(\mathcal{A}_{S_m, X_u}(x_i), y_i)$. When using the margin loss function we denote the training and transductive errors of \mathcal{A} by $\hat{R}_m^\gamma(\mathcal{A})$ and $R_u^\gamma(\mathcal{A})$, respectively.

¹ There is still some disagreement between authors about the definitions of ‘semi-supervised’ and ‘transductive’ learning. The authors of [2] study a transductive setting (according to the terminology presented here) but call it ‘semi-supervised’.

² For a positive real γ , $\ell_\gamma(y_1, y_2) = 0$ if $y_1 y_2 \geq \gamma$ and $\ell_\gamma(y_1, y_2) = \min\{1, 1 - y_1 y_2 / \gamma\}$ otherwise.

Note that in this transductive setting there is no underlying distribution as in (semi-supervised) inductive models.³ Also, training examples are *dependent* due to the sampling without replacement of the training set from the full sample.

We require the following standard definitions and facts about martingales.⁴ Let $\mathbf{B}_1^n \triangleq (B_1, \dots, B_n)$ be a sequence of random variables. The sequence $\mathbf{W}_0^n \triangleq (W_0, W_1, \dots, W_n)$ is called a *martingale* w.r.t. the *underlying* sequence \mathbf{B}_1^n if for any $1 \leq i \leq n$, W_i is a function of \mathbf{B}_1^i and $\mathbf{E}_{B_i} \{W_i | \mathbf{B}_1^{i-1}\} = W_{i-1}$. The sequence of random variables $\mathbf{d}_1^n = (d_1, d_2, \dots, d_n)$, where $d_i \triangleq W_i - W_{i-1}$, is called the *martingale difference sequence* of \mathbf{W}_n . An elementary fact is that $\mathbf{E}_{B_i} \{d_i | \mathbf{B}_1^{i-1}\} = 0$.

Let $f(\mathbf{Z}_1^n) \triangleq f(Z_1, \dots, Z_n)$ be an arbitrary function of n (possibly dependent) random variables. Let $W_0 \triangleq \mathbf{E}_{\mathbf{Z}_1^n} \{f(\mathbf{Z}_1^n)\}$ and $W_i \triangleq \mathbf{E}_{\mathbf{Z}_1^n} \{f(\mathbf{Z}_1^n) | \mathbf{Z}_1^i\}$ for any $1 \leq i \leq n$. An elementary fact is that \mathbf{W}_0^n is a martingale w.r.t. the underlying sequence \mathbf{Z}_n . Thus we can obtain a martingale from any function of (possibly dependent) random variables. This routine of obtaining a martingale from an arbitrary function is called *Doob's martingale process*. Let \mathbf{d}_1^n be the martingale difference sequence of \mathbf{W}_0^n . Then $\sum_{i=1}^n d_i = W_n - W_0 = f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}_1^n} \{f(\mathbf{Z}_1^n)\}$. Consequently, to bound the deviation of $f(\mathbf{Z})$ from its mean it is sufficient to bound the martingale difference sum. A fundamental inequality, providing such a bound, is the Azuma inequality.

Lemma 1 (Azuma,[1]). *Let \mathbf{W}_0^n be a martingale w.r.t. \mathbf{B}_1^n and \mathbf{d}_1^n be its difference sequences. Suppose that for all $i \in I_1^n$, $|d_i| \leq b_i$. Then*

$$\mathbf{P}_{\mathbf{B}_1^n} \{W_n - W_0 > \epsilon\} < \exp\left(-\frac{\epsilon^2}{2 \sum_{i=1}^n b_i^2}\right). \quad (1)$$

4 Uniform Stability Bound

Given a training set S_m and a test set X_u and two indices $i \in I_1^m$ and $j \in I_{m+1}^{m+u}$, let $S_m^{ij} \triangleq S_m \setminus \{z_i\} \cup \{z_j = \langle x_j, y_j \rangle\}$ and $X_u^{ij} \triangleq X_u \setminus \{x_j\} \cup \{x_i\}$ (e.g., S_m^{ij} is S_m with the i th example (from the training set) and j th example (from the test set) exchanged). The following definition of stability is a straightforward adaptation of the uniform stability definition from [4] to our transductive setting.

Definition 1 (Uniform Transductive Stability). *A transductive learning algorithm \mathcal{A} has uniform transductive stability β if for all choices of $S_m \subset S_{m+u}$, for all $i \in I_1^m$, $j \in I_{m+1}^{m+u}$,*

$$\max_{1 \leq k \leq m+u} \left| \mathcal{A}_{S_m, X_u}(x_k) - \mathcal{A}_{S_m^{ij}, X_u^{ij}}(x_k) \right| \leq \beta. \quad (2)$$

³ As discussed earlier, Vapnik also considers a second transductive setting where examples are drawn from some unknown distribution; see Chapter 8 in [19]. Results in the model we study here apply to the other model (Theorem 8.1 in [19]).

⁴ See, e.g., [7], Chapt. 12 and [6] Sec. 9.1 for more details.

Let $\mathbf{Z} \triangleq \mathbf{Z}_1^{m+u} \triangleq (Z_1, \dots, Z_{m+u})$ be a *random permutation vector* where the variable Z_k , $k \in I_1^{m+u}$, is the k th component of a permutation of I_1^{m+u} , chosen uniformly at random. Let \mathbf{Z}^{ij} be a perturbed permutation vector obtained by exchanging Z_i and Z_j in \mathbf{Z} . A function f on permutations of I_1^{m+u} is called (m, u) -*symmetric* permutation function if $f(\mathbf{Z}) = f(Z_1, \dots, Z_{m+u})$ is symmetric on Z_1, \dots, Z_m as well as on Z_{m+1}, \dots, Z_{m+u} .

Let $H_2(n) \triangleq \sum_{i=1}^n \frac{1}{i^2}$ and $K(m, u) \triangleq u^2(H_2(m+u) - H_2(u))$. It can be verified that $K(m, u) < m$. The following lemma is obtained⁵ by a straightforward application of the Azuma inequality to a martingale obtained from $f(\mathbf{Z})$ by Doob's process.

Lemma 2. *Let \mathbf{Z} be a random permutation vector. Let $f(\mathbf{Z})$ be an (m, u) -symmetric permutation function satisfying $|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| \leq \beta$ for all $i \in I_1^m$, $j \in I_{m+1}^{m+u}$. Then*

$$\mathbf{P}_{\mathbf{Z}} \{f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} \geq \epsilon\} \leq \exp\left(-\frac{\epsilon^2}{2\beta^2 K(m, u)}\right). \quad (3)$$

Our first transductive error bound is obtained by applying Lemma 2 to the function $R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A})$ and bounding $\mathbf{E}\{R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A})\}$ using an adaptation of Lemma 7 from [4] to our setting.

Theorem 1. *Let \mathcal{A} be a transductive learning algorithm with transductive uniform stability β . Let $\tilde{\beta} \triangleq \frac{(u-1)\beta}{u\gamma} + \frac{(m-1)\beta}{m\gamma} + \frac{1}{m} + \frac{1}{u}$. Then, for all $\gamma > 0$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the draw of the training/test sets (S_m, X_u) ,*

$$R_u(\mathcal{A}) \leq \hat{R}_m^\gamma(\mathcal{A}) + \beta/\gamma + \tilde{\beta}\sqrt{2K(m, u)\ln(1/\delta)}. \quad (4)$$

The tightness of the bound (4) depends on the transductive uniform stability β of algorithm \mathcal{A} . If $\beta = O(1/m)$ and $u = \Omega(m)$, then the slack terms in (4) amount to $O(\sqrt{\ln(1/\delta)/m/\gamma})$. However, in our experience this stability rate is never met by useful transductive algorithms.

5 Weak Stability Bound

The impractical requirement of the uniform stability concept motivates a weaker notion of stability that we develop here. The following definition is inspired by a definition of Kutin for inductive learning (see Definition 1.7 in [11]).

Definition 2 (Weak Permutation Stability). *Let \mathbf{Z} be a random permutation vector. A function $f(\mathbf{Z})$ has weak permutation stability $(\beta, \beta_1, \delta_1)$ if f has uniform stability β and*

$$\mathbf{P}_{\mathbf{Z}, i \sim I_1^m, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| \leq \beta_1\} \geq 1 - \delta_1, \quad (5)$$

where $i \sim I$ denotes a choice of $i \in I$ uniformly at random.

⁵ All omitted proofs will appear in the full version of the paper.

This weaker notion of stability only requires that $|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})|$ be bounded with respect to most exchanges, allowing for a δ_1 -fraction of outliers. To utilize Definition 2 we develop in Lemma 3 a new concentration inequality for symmetric permutation functions that satisfy the new weak stability property.

Lemma 3. *Let \mathbf{Z} be a random permutation vector and $f(\mathbf{Z})$ be an (m, u) -symmetric permutation function. Suppose that $f(\mathbf{Z})$ has weak permutation stability $(\beta, \beta_1, \delta_1)$. Let $\delta \in (0, 1)$ be given, and for $i \in I_1^m$, let $\theta_i \in (0, 1)$, $\Psi \triangleq \delta_1 \sum_{i=1}^m 1/\theta_i$ and $b_i \triangleq \frac{((1-\theta_i)\beta_1 + \theta_i\beta)}{(m+u-i+1)(1-\Psi)}$. If $\Psi < 1$, then with probability at least $(1-\delta) \cdot (1-\Psi)$ over the choices of \mathbf{Z} ,*

$$f(\mathbf{Z}) \leq \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\} + u \sqrt{2 \sum_{i=1}^m b_i^2 \ln \frac{1}{\delta}} . \quad (6)$$

Note that the confidence level can be made arbitrarily small by selecting appropriate θ_i and δ_1 (thus trading-off β_1).

Proof: Let \mathbf{W}_0^{m+u} be a martingale generated from $f(\mathbf{Z})$ by Doob's process. We derive bounds on the martingale differences d_i , $i \in I_1^{m+u}$, and apply Lemma 1.

Let $\boldsymbol{\pi}_1^{m+u} = \pi_1, \dots, \pi_{m+u}$ be a specific permutation of I_1^{m+u} . In the proof we use the following shortcut: $\mathbf{Z}_1^r = \boldsymbol{\pi}_1^r$ abbreviates the r equalities $Z_1 = \pi_1, \dots, Z_r = \pi_r$. Let θ_i be given. For $r \in I_1^m$, we say that the prefix $\boldsymbol{\pi}_1^r$ of a permutation $\boldsymbol{\pi}_1^{m+u}$ is (r, θ_r) -admissible (w.r.t. a fixed β_1) if it guarantees that

$$\mathbf{P}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}) - f(\mathbf{Z}^{rj})| \leq \beta_1 \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r\} \geq 1 - \theta_r . \quad (7)$$

If the prefix $\boldsymbol{\pi}_1^r$ does not satisfy (7), we say that it is not (r, θ_r) -admissible. Let $\zeta(r, \theta_r)$ be the probability that \mathbf{Z}_1^r is not (r, θ_r) -admissible. Our goal is to bound $\zeta(r, \theta_r)$. For any fixed $1 \leq r \leq m$ we have,

$$\begin{aligned} t(r) &\triangleq \mathbf{P}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}) - f(\mathbf{Z}^{rj})| > \beta_1\} \\ &= \sum_{\substack{\text{all possible} \\ \text{prefixes } \boldsymbol{\pi}_1^r}} \left(\mathbf{P}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}) - f(\mathbf{Z}^{rj})| > \beta_1 \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r\} \cdot \mathbf{P}_{\mathbf{Z}} \{\mathbf{Z}_1^r = \boldsymbol{\pi}_1^r\} \right) \\ &\geq \sum_{\substack{\text{non-} \\ \text{admissible} \\ \text{prefixes } \boldsymbol{\pi}_1^r}} \left(\mathbf{P}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}) - f(\mathbf{Z}^{rj})| > \beta_1 \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r\} \cdot \mathbf{P}_{\mathbf{Z}} \{\mathbf{Z}_1^r = \boldsymbol{\pi}_1^r\} \right) \\ &\geq \theta_r \cdot \sum_{\substack{\text{non-admissible} \\ \text{prefixes } \boldsymbol{\pi}_1^r}} \mathbf{P}_{\mathbf{Z}} \{\mathbf{Z}_1^r = \boldsymbol{\pi}_1^r\} = \theta_r \zeta(r, \theta_r) . \end{aligned} \quad (8)$$

Since $f(\mathbf{Z})$ is (m, u) -permutation symmetric, $t(r) = t$ is constant. Since $f(\mathbf{Z})$ has weak permutation stability $(\beta, \beta_1, \delta_1)$,

$$\delta_1 \geq \mathbf{P}_{\mathbf{Z}, i \sim I_1^m, j \sim I_{m+1}^{m+u}} \{|f(\mathbf{Z}) - f(\mathbf{Z}^{ij})| > \beta_1\} = \sum_{r=1}^m \frac{1}{m} \cdot t(r) = t \geq \theta_r \zeta(r, \theta_r) . \quad (9)$$

Consequently, $\zeta(r, \theta_r) \leq \delta_1/\theta_r$. Our next goal is to bound d_r for (r, θ_r) -admissible prefixes. Let $l(k)$ be an index l such that $Z_l = k$. If $\boldsymbol{\pi}_1^r = (\boldsymbol{\pi}_1^{r-1}, \pi_r = k)$ is (r, θ_r) -admissible, then

$$\begin{aligned} |d_r| &= |W_r - W_{r-1}| = |\mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z}) \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r\} - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z}) \mid \mathbf{Z}_1^{r-1} = \boldsymbol{\pi}_1^{r-1}\}| \\ &= \left| \mathbf{E}_{\mathbf{Z}} \left\{ f(\mathbf{Z}^{r l(k)}) - f(\mathbf{Z}) \mid \mathbf{Z}_1^{r-1} = \boldsymbol{\pi}_1^{r-1} \right\} \right| \\ &= \left| \mathbf{E}_{\mathbf{Z}, j \sim I_r^{m+u}} \left\{ f(\mathbf{Z}) - f(\mathbf{Z}^{r j}) \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r \right\} \right| \end{aligned} \quad (10)$$

$$\leq \mathbf{E}_{\mathbf{Z}, j \sim I_r^{m+u}} \left\{ |f(\mathbf{Z}^{r j}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r \right\} \quad (11)$$

$$= \mathbf{P}_{j \sim I_r^{m+u}} \{j \in I_r^m\} \cdot \mathbf{E}_{\mathbf{Z}, j \sim I_r^m} \left\{ |f(\mathbf{Z}^{r j}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r \right\} \quad (12)$$

$$+ \mathbf{P}_{j \sim I_r^{m+u}} \{j \in I_{m+1}^{m+u}\} \cdot \mathbf{E}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \left\{ |f(\mathbf{Z}^{r j}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r \right\} \quad (13)$$

$$\leq \frac{u((1-\theta_r)\beta_1 + \theta_r\beta)}{m+u-r+1} \triangleq b_r .$$

The inequality (13) follows because (i) the expectation in (11) is zero since f is (m, u) -permutation symmetric; and (ii) $\boldsymbol{\pi}_1^r$ is (r, θ_r) -admissible, implying that the expectation in (12) is bounded by $(1-\theta_r)\beta_1 + \theta_r\beta$.

A permutation $\boldsymbol{\pi}_1^{m+u}$ is *good* if for all $r \in I_1^m$ its r -prefixes, $\boldsymbol{\pi}_1^r$, are admissible (w.r.t. the corresponding θ_r). Since $\zeta(r, \theta_r) \leq \delta_1/\theta_r$, we have

$$\mathbf{P}_{\mathbf{Z}} \{\mathbf{Z} \text{ not good}\} \leq \sum_{r=1}^m \mathbf{P}_{\mathbf{Z}} \{\mathbf{Z}_1^r \text{ not admissible}\} = \sum_{r=1}^m \zeta(r, \theta_r) \leq \sum_{r=1}^m \frac{\delta_1}{\theta_r} = \Psi . \quad (14)$$

Thus, with probability at least $1 - \Psi$, the random permutation \mathbf{Z} is good, in which case we have $|d_r| \leq b_r$ for all $r \in I_1^m$.

Consider the space \mathcal{G} of all good permutations. Let \mathbf{V}_0^{m+u} be a martingale obtained by Doob's process operated on f and \mathcal{G} . Then, using (13) we bound the martingale difference sequence \mathbf{d}_1^{m+u} of \mathbf{V}_0^{m+u} as follows.

$$\begin{aligned} |d'_i| &\leq \mathbf{P}_{j \sim I_r^{m+u}} \{j \in I_{m+1}^{m+u}\} \times \\ &\quad \mathbf{E}_{\mathbf{Z} \in \mathcal{G}, j \sim I_{m+1}^{m+u}} \left\{ |f(\mathbf{Z}^{r j}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r, \boldsymbol{\pi}_1^r \text{ is admissible} \right\} \quad (15) \\ &\leq \mathbf{P}_{j \sim I_r^{m+u}} \{j \in I_{m+1}^{m+u}\} \times \\ &\quad \frac{\mathbf{E}_{\mathbf{Z}, j \sim I_{m+1}^{m+u}} \left\{ |f(\mathbf{Z}^{r j}) - f(\mathbf{Z})| \mid \mathbf{Z}_1^r = \boldsymbol{\pi}_1^r, \boldsymbol{\pi}_1^r \text{ is admissible} \right\}}{\mathbf{P}_{\mathbf{Z}} \{\mathbf{Z} \in \mathcal{G}\}} \\ &\leq \frac{u((1-\theta_r)\beta_1 + \theta_r\beta)}{(m+u-r+1)(1-\Psi)} \triangleq b_r . \end{aligned} \quad (16)$$

Since $f(\mathbf{Z})$ is (m, u) -permutation symmetric, it follows from (10) that for any $r \in I_{m+1}^{m+u}$, $d'_r = 0$. Therefore, we can apply Lemma 1 to the martingale \mathbf{V}_0^{m+u} . We obtain a bound on the deviation of $V_{m+u} - V_0 = f(\mathbf{Z}) - \mathbf{E}_{\mathbf{Z}} \{f(\mathbf{Z})\}$. Our result (6) is completed by equating the resulting bound to δ and isolating ϵ . \square

It follows from Definition 2 that β_1 depends on δ_1 . Hence, the bound (6) depends on the following parameters: $\delta_1, \theta_i, i \in I_1^m$. It can be shown that if

$\beta_1 = O(1/m)$, $\delta_1 = O(1/m^2)$ and $\theta_i = O(1/m)$ for all $i \in I_1^m$, then the slack term in (6) is $O(\sqrt{\ln(1/\delta)/m})$ and the bound's confidence can be made arbitrarily close to 1.

Our goal now is to derive an error bound for transductive algorithms by utilizing the weak stability notion. To this end, we now define weak transductive stability for algorithms. The following definition, which contains three conditions and six parameters, is somewhat cumbersome but we believe it facilitates tighter bounds than can possibly be achieved using a simpler definition (that only includes condition (18) below); see also the discussion that follows this definition. For a fixed full sample, we abbreviate $\mathcal{A}^{ij}(x, (S_m, X_u)) \triangleq |\mathcal{A}_{S_m, X_u}(x) - \mathcal{A}_{S_m^i, X_u^j}(x)|$.

Definition 3 (Weak Transductive Stability). *A transductive learning algorithm \mathcal{A} has weak transductive stability $(\beta, \beta_1, \beta_2, \delta_1^a, \delta_1^b, \delta_2)$ if it has uniform transductive stability β and the following conditions (17) and (18) hold.*

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \mathbf{P}_{x \sim X_{m+u}} \left\{ \mathcal{A}^{ij}(x, (S_m, X_u)) \leq \beta_1 \right\} \geq 1 - \delta_1^a \right\} \geq 1 - \delta_1^b . \quad (17)$$

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \mathcal{A}^{ij}(x_i, (S_m, X_u)) \leq \beta_2 \right\} \geq 1 - \delta_2 . \quad (18)$$

While in (17) we quantify the sensitivity of the algorithm w.r.t. all examples in X_{m+u} , in (18) only the exchanged examples are considered. A number of weak stability definitions for induction is given in [10, 12, 15]. Ignoring the differences between induction and transduction, our condition (17) poses a qualitatively weaker constraint than the ‘weak hypothesis stability’ (Definition 3.5 in [12]), and a stronger constraint than the ‘weak error stability’ (Definition 3.8 in [12]). Our condition (18) is a straightforward adaptation of the ‘cross-validation stability’ (Definition 3.12 in [12]) to our transductive setting.

It should be possible to show, using a technique similar to the one used in the proof of Theorem 3.16 in [12], that (18) implies (17). In this case a simpler weak stability definition may suffice but, using our techniques, the resulting error bound would be looser.

Let $\Delta(i, j, s, t) \triangleq \ell_\gamma(\mathcal{A}_{S_m^i, X_u^j}(x_t), y_t) - \ell_\gamma(\mathcal{A}_{S_m, X_u}(x_s), y_s)$. For the proof of the forthcoming error bound we need the following technical lemma.

Lemma 4. $\mathbf{E}_{(S_m, X_u)} \left\{ R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A}) \right\} = \mathbf{E}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \Delta(i, j, i, i) \right\}$.

Theorem 2. *Let \mathcal{A} be an algorithm with weak transductive classification stability $(\beta, \beta_1, \beta_2, \delta_1^a, \delta_1^b, \delta_2)$. Suppose that $u \geq m$ and $\delta_1^a < \frac{m}{m+u}$.⁶ Let $\gamma > 0$, $\delta \in (0, 1)$ be given and set*

$$\tilde{\beta}_1 \triangleq \frac{u-1}{u} \cdot \frac{\beta_1}{\gamma} + \frac{\delta_1^a(m+u)\beta + [m-1 - \delta_1^a(m+u)]\beta_1}{m\gamma} + \frac{1}{m} + \frac{1}{u} , \quad (19)$$

⁶ The proof for the cases $\delta_1^a > \frac{m}{m+u}$ and $m > u$ is very similar to the proof given below and is omitted.

$$\tilde{\beta} \triangleq \frac{u-1}{u} \cdot \frac{\beta}{\gamma} + \frac{m-1}{m} \cdot \frac{\beta}{\gamma} + \frac{1}{m} + \frac{1}{u} . \quad (20)$$

For any $\theta_i \in (0, 1)$, $i \in I_1^m$, set $\Psi \triangleq \sum_{i=1}^m \frac{\delta_1^b}{\theta_i}$ and $b_i \triangleq \frac{u(1-\theta_i)\tilde{\beta}_1 + \theta_i\tilde{\beta}}{(m+u-i+1)(1-\Psi)}$. If $\Psi < 1$, then with probability at least $(1-\delta) \cdot (1-\Psi)$ over the draw of the training/test sets (S_m, X_u) ,

$$R_u(\mathcal{A}) \leq \hat{R}_m^\gamma(\mathcal{A}) + \left[(1-\delta_2) \frac{\beta_2}{\gamma} + \delta_2 \frac{\beta}{\gamma} \right] + \sqrt{2 \sum_{i=1}^m b_i^2 \ln \frac{1}{\delta}} . \quad (21)$$

Proof: We derive bounds on the weak permutation stability of the function $f(S_m, X_u) \triangleq R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A})$ and its expected value. Then we apply Lemma 3. For $i \in I_1^m$, $j \in I_{m+1}^{m+u}$, we have (by expanding the risk expressions),

$$\begin{aligned} & \left| R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A}) - \left(R_u^\gamma(\mathcal{A}_{S_m^{ij}, X_u^{ij}}) - \hat{R}_m^\gamma(\mathcal{A}_{S_m^{ij}, X_u^{ij}}) \right) \right| \leq \\ & \frac{1}{u} \sum_{\substack{k=m+1, \\ k \neq j}}^{m+u} |\Delta(i, j, k, k)| + \frac{1}{u} |\Delta(i, j, j, i)| + \frac{1}{m} \sum_{\substack{k=1, \\ k \neq i}}^m |\Delta(i, j, k, k)| + \frac{1}{m} |\Delta(i, j, i, j)| . \end{aligned} \quad (22)$$

Since ℓ_γ has Lipschitz constant γ , it follows from (17) that

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \left\{ \mathbf{P}_{k \sim I_1^{m+u}} \{ |\Delta(i, j, k, k)| \leq \beta_1/\gamma \} \geq 1 - \delta_1^a \right\} \geq 1 - \delta_1^b . \quad (23)$$

We say that the example x_k is *bad* if $|\Delta(i, j, k, k)| > \beta_1/\gamma$. According to (23), with probability at least $1 - \delta_1^b$ over the choices of $((S_m, X_u), i, j)$, there are at most $(1 - \delta_1^a)(m+u)$ bad examples. If $u \geq m$, the terms in the second summation in (22) have greater weight (which is $1/m$) than the terms in the first summation (weighted by $1/u$). In the worst case all bad examples appear in the second summation in which case (22) is bounded by (19) with probability at least $1 - \delta_1^b$ over the choices of $((S_m, X_u), i, j)$.

The right hand side of (22) is always bounded by $\tilde{\beta}$. Therefore, the function $f(S_m, X_u)$ has weak permutation stability $(\tilde{\beta}, \tilde{\beta}_1, \delta_1^b)$. By applying Lemma 3 to $f(S_m, X_u)$, we obtain that with probability at least $(1-\delta)(1-\Psi)$,

$$R_u^\gamma(\mathcal{A}) \leq \hat{R}_m^\gamma(\mathcal{A}) + \mathbf{E}_{(S_m, X_u)} \left\{ R_u^\gamma(\mathcal{A}) - \hat{R}_m^\gamma(\mathcal{A}) \right\} + \sqrt{2 \sum_{i=1}^m b_i^2 \ln \frac{1}{\delta}} . \quad (24)$$

Since ℓ_γ has Lipschitz constant γ , it follows from (18) that

$$\mathbf{P}_{(S_m, X_u), i \sim I_1^m, j \sim I_{m+1}^{m+u}} \{ |\Delta(i, j, i, i)| \leq \beta_2/\gamma \} \geq 1 - \delta_2 . \quad (25)$$

Therefore, the right hand side of the equality in Lemma 4 is bounded from above by $\beta_2(1-\delta_2)/\gamma + \beta\delta_2/\gamma$. By substituting this bound to (24) and using the inequality $R_u^\gamma(\mathcal{A}) \geq R_u(\mathcal{A})$, we obtain (21). \square

It follows from Definition 3 that β_1 depends on δ_1^a and δ_1^b , and that β_2 depends on δ_2 . Hence the bound (21) depends on the parameters $\delta_1^a, \delta_1^b, \delta_2, \theta_i, i \in I_1^m$. It is possible to show that if $u = \Omega(m)$, $\delta_1^a = O(1/m + u)$, $\delta_1^b = O(1/m^2)$ and $\beta_1, \beta_2, \delta_2, \theta_i$ are each $O(1/m)$, then the slack term in (21) is $O(\sqrt{\ln(1/\delta)/m/\gamma})$ and the bound's confidence can be made arbitrarily close to 1.

6 High Confidence Stability Estimation

In this section we describe a routine that can generate useful upper bounds on the weak stability parameters (Definition 3) of transductive algorithms. The routine generates these estimates with arbitrarily high probability and is based on a sampling-based quantile estimation technique. Given a particular learning algorithm, our stability estimation routine relies on an ‘‘oracle’’ that bounds the sensitivity of the transductive algorithm with respect to a small change in the input. We present such an oracle for a familiar practical algorithm. In Sec. 6.1 we describe the quantile estimation method, which is similar to the one presented in [14]; in Sec. 6.2 we present the bounding algorithm, and in Sec. 6.3 we consider a known transductive algorithm and present a few numerical examples of the application of these methods.

6.1 Quantile Estimation

Consider a very large set Ω of N numbers. Define the q -quantile of Ω to be the $\lceil qN \rceil$ -th smallest element of Ω (i.e., it is the $\lceil qN \rceil$ -th element in an increasing order sorted list of all elements in Ω). Our goal is to bound the q -quantile x_q from above as tightly as possible, with high confidence, by sampling a ‘‘small’’ number $k \ll N$ of elements. For any $\epsilon \in (0, 1)$ we generate a bound β such that $\mathbf{P}\{x_q \leq \beta\} \geq 1 - \epsilon$. The idea is to sample $k = k(q, \epsilon)$ elements from Ω uniformly at random, compute their exact $(\bar{q} \triangleq q + \frac{1-q}{2})$ -quantile $x_{\bar{q}}$, and output $\beta \triangleq x_{\bar{q}}$. Denote by $\mathbf{quantile}(q, \epsilon, \Omega)$ the resulting routine whose output is $\beta = x_{\bar{q}}$.

Lemma 5. *For any $q, \epsilon \in (0, 1)$. If $k = k(q, \epsilon) = \frac{2 \ln(1/\epsilon)}{(1-q)^2}$, then*

$$\mathbf{P}\{x_q \leq \mathbf{quantile}(q, \epsilon, \Omega)\} \geq 1 - \epsilon . \quad (26)$$

Proof: For $i \in I_1^k$ let X_i be the indicator random variable obtaining 1 if the i th drawn element (from Ω) is smaller than x_q , and 0 otherwise. Set $Q = \frac{1}{k} \sum_{i=1}^k X_i$. Clearly, $\mathbf{E}Q \leq q$. By Hoeffding's inequality and using the definition of \bar{q} , we get

$$\begin{aligned} \mathbf{P}\{Q > \bar{q}\} &= \mathbf{P}\left\{Q - q > \frac{1-q}{2}\right\} \\ &\leq \mathbf{P}\left\{Q - \mathbf{E}Q > \frac{1-q}{2}\right\} \leq \exp\left(-\frac{k(1-q)^2}{2}\right) . \end{aligned} \quad (27)$$

Therefore, with ‘‘high probability’’ the number kQ of sample points that are smaller than x_q is smaller than $k\bar{q}$. Hence, at least $(1-\bar{q})k$ points in the sample are larger than x_q . $\mathbf{quantile}$ returns the smallest of them. Equating the right hand side of (27) to ϵ and solving for k yields the stated sample size. \square

6.2 Stability Estimation Algorithm

Let \mathcal{A} be a transductive learning algorithm. We assume that some (rough) bound on \mathcal{A} 's uniform stability β is known. If no tight bound is known, we take the maximal default value, which is 2, as can be seen in Definition 1. Our goal is to find useful bounds for the weak stability parameters of Definition 3. Let the values of δ_1^a , δ_1^b and δ_2 be given. We aim at finding upper bounds on β_1 and β_2 .

Definition 4 (The diff Oracle). Consider a fixed labeled training set $S_m = (X_m, Y_m)$ given to the learning algorithm. Let $\mathbf{diff}(\tilde{X}_m, \tilde{X}_u, i, j, r | S_m)$ be an “oracle” function defined for any possible partition $(\tilde{X}_m, \tilde{X}_u)$ of the full sample and indices $i \in I_1^m$, $j \in I_{m+1}^{m+u}$ and $r \in I_1^{m+u}$. \mathbf{diff} provides an upper bound on

$$\left| \mathcal{A}_{\tilde{S}_m, \tilde{X}_u}(x_r) - \mathcal{A}_{\tilde{S}_m^{ij}, \tilde{X}_u^{ij}}(x_r) \right|, \quad (28)$$

where \tilde{S}_m is any possible labeling of \tilde{X}_m that “agrees” with S_m on points in $X_m \cap \tilde{X}_m$. Note that here we assume that I_1^m is the set indices of points in \tilde{X}_m (and indices in X_m are not specified and can be arbitrary indices in I_1^{m+u}).

We assume that we have an access to a useful $\mathbf{diff}(\tilde{X}_m, \tilde{X}_u, i, j, r | S_m)$ function that provides a tight upper bound on (28). We now describe our stability estimation algorithm that applies \mathbf{diff} .

Let K be the set of all possible quadruples $(\tilde{X}_m, \tilde{X}_u, i, j)$ as in Definition 4. Define $\Omega_1 = \{\omega(t) : t \in K\}$, $\omega(t) = \omega(\tilde{X}_m, \tilde{X}_u, i, j)$ is a $(1 - \delta_1^a)$ -quantile of the set $\Phi = \left\{ \mathbf{diff}(\tilde{X}_m, \tilde{X}_u, i, j, r | S_m), r = 1, \dots, m + u \right\}$. It is not hard to see that for any $\epsilon \in (0, 1)$, with probability at least $1 - \epsilon$ (over random choices made by the `quantile` routine), $\mathbf{quantile}(1 - \delta_1^b, \epsilon, \Omega_1)$ is an upper bound on the weak stability parameter β_1 of Definition 3. Likewise, let $\Omega_2 = \{\omega(t) : t \in K\}$, but now $\omega(t) = \omega(\tilde{X}_m, \tilde{X}_u, i, j) = \mathbf{diff}(\tilde{X}_m, \tilde{X}_u, i, j, i)$. It is not hard to see that for any ϵ , with probability at least $1 - \epsilon$, $\mathbf{quantile}(1 - \delta_2, \epsilon, \Omega_2)$ is an upper bound on the weak stability parameter β_2 of Definition 3.

Thus, our weak stability estimation algorithm simply applies `quantile` twice with appropriate parameters. To actually draw the samples, `quantile` utilizes the `diff` function. Let v be the time complexity of computing `diff` oracle. By Lemma 5 the number of samples that should be drawn, in order to obtain with probability at least $1 - \epsilon$ the bound on q -quantile, is $O(\ln(1/\epsilon)/(1 - q)^2)$. It can be verified that the complexity of our stability estimation algorithm is $O(\ln(1/\epsilon)(m + u)v / \min\{(\delta_1^b)^2, (\delta_2)^2\})$. As discussed after Theorem 2, δ_1^b should be $O(1/m^2)$ to ensure that the bound (21) has arbitrarily high confidence. This constraint entails a time complexity of $\Omega(m^4(m + u))$. Hence currently our ability to use the stability estimation routine in conjunction with the transductive error bound is limited to very small values of m .

6.3 Stability Estimation Examples

In this section we consider the transductive learning algorithm of Zhou et al. [20] and demonstrate a data-dependent estimation of its weak stability parameters

using our method. While currently there is no comprehensive empirical comparison between all available transductive algorithms, this algorithm appears to be among the more promising ones [8]. We chose this algorithm, denoted by **CM** (stands for ‘Consistency Method’; see [8]), because we could easily develop a useful **diff** “oracle” for it. We were also able to efficiently implement **diff** “oracle” for the algorithm of Zhu et al. [21], which will be presented elsewhere.

We start with the brief description of the **CM** algorithm. Let W be a symmetric $(m+u) \times (m+u)$ affinity matrix of the full sample X_{m+u} . We assume that $W_{ii} = 0$. In this paper we use RBF kernels, parameterized by σ , to construct W . Let D be a diagonal matrix, whose (i, i) -element is the sum of the i th row in W . A normalized Laplacian of W is $L = D^{-1/2}WD^{-1/2}$. Let α be a parameter in $(0, 1)$. Let Y be an $(m+u) \times 1$ vector of available full sample labels, where the entries corresponding to training examples are ± 1 and entries of unlabeled examples are 0. We assume w.l.o.g. that the first m entries in Y correspond to the m labeled training examples. Let $P = (I - \alpha L)^{-1}$. The **CM** algorithm produces soft-classification $F = P \cdot Y$. In other words, if p_{ij} is the (i, j) th entry of P and f_i is the i th entry of F , the point x_i receives the soft-classification

$$f_i = \sum_{j=1}^m p_{ij} y_j . \quad (29)$$

To obtain useful bounds on the (weak) stability of **CM** we require the following benign technical modifications of **CM** that would not change the *hard* classification it generates over test set examples.

1. We prevent over-fitting to the training set by setting $p_{ii} = 0$.
2. To enable a comparison between stability values corresponding to different settings of the parameters α and σ , we ensure that the dynamic range of f_i is normalized w.r.t. different values of α and σ . That is, instead of using (29) for prediction we use

$$f_i = \frac{\sum_{j=1}^m p_{ij} y_j}{\sum_{j=1}^m p_{ij}} . \quad (30)$$

The first modification prevents possible over-fitting to the training set since for any $i \in I_1^{m+u}$, in the original **CM** the value of p_{ii} is much larger than any of the other p_{ij} , $j \neq i$, and therefore, the soft classification of the training example x_i is almost completely determined by its given label y_i . Hence by (29), when x_i is exchanged with some test set example x_j , the soft classification change of x_i will probably be large. Therefore, the stability condition (18) cannot be satisfied with small values of β_2 . By setting $p_{ii} = 0$ we prevent this problem and only affect the soft and hard classification of training examples (and keep the soft classifications of test points intact). The second modification clearly changes the dynamic range of all soft classifications but does not alter any hard classification.

To use our stability estimation algorithm one should provide an implementation of **diff**. We show that for the **CM** algorithm $\text{diff}(\tilde{X}_m, \tilde{X}_u, i, j, r | S_m)$ can be effectively implemented as follows. For notational convenience we assume

here (and also in Definition 4) that examples in \tilde{X}_m have indices in I_1^m . Let $\tau(r) = \sum_{k=1, k \neq i}^m p_{rk}$ and $\tau_y(r) = \sum_{k=1, k \neq i}^m p_{rk} y_k$. It follows from (30) that

$$\left| \mathcal{A}_{\tilde{S}_m, \tilde{X}_u}(x_r) - \mathcal{A}_{\tilde{S}_m^{ij}, \tilde{X}_u^{ij}}(x_r) \right| = \left| \frac{(p_{rj} - p_{ri}) \cdot \sum_{k=1, k \neq i, x_k \notin X_m}^m p_{rk} y_k + T}{(\tau(r) + p_{ri})(\tau(r) + p_{rj})} \right|, \quad (31)$$

where $T \triangleq (p_{rj} - p_{ri}) \cdot \sum_{k=1, k \neq i, x_k \in X_m}^m p_{rk} y_k + \tau(r) \cdot (p_{ri} y_i - p_{rj} y_j) + p_{ri} p_{rj} (y_i - y_j)$.

To implement $\text{diff}(\tilde{X}_m, \tilde{X}_u, i, j, r | S_m)$ we should upper bound (31). Suppose first that the values of y_i and y_j are known. Then, T is constant and the only unknowns in (31) are the y_k 's in the first summation. Observe that (31) is maximal when all values of these y_k 's are -1 or all of them are $+1$. Hence by taking the maximum over these possibilities we obtain an upper bound on (31). If y_i (or y_j) is unknown then, similarly, for each of its possible assignments we compute (31) and take the maximum. In the worst case, when both y_i and y_j are unknown, we compute the maximum of (31) over the eight possible assignments for these two variables and the y_k 's in the first summation. It can be verified that the time complexity of the above diff oracle is $O(m)$.

We now show two numerical examples of stability estimations for the **CM** algorithm with respect to two UCI datasets. These results were obtained by implementing the modified **CM** algorithm and the stability estimation routine applied with the above implementation of diff . For each ‘‘experiment’’ we ran the modified **CM** algorithm with 21 different hyper-parameter settings for α and σ , each resulting in a different application of the algorithm.⁷

We considered two UCI datasets, **musk** and **mush**. From each dataset we generated 30 random full samples X_{m+u} each consisting of 400 points. We divided each full sample instance to equally sized training and test sets uniformly at random. The high confidence (95%) estimation of stability parameter β_1 (see Definition 3) w.r.t. $\delta_1^a = \delta_1^b = 0.1$, and the corresponding empirical and true risks are shown in Fig. 1. The graphs for the β_2 parameter are qualitatively similar and are omitted here. Indices in the x -axis correspond to the 21 applications of **CM** and are sorted in increasing order of true risk. Each stability and error value depicted is an average over the 30 random full samples. We also depict a high confidence (95%) true stability estimates, obtained *in hindsight* by using the unknown labels in the computation of diff . The uniform stability graphs correspond to *lower bounds* obtained by taking the maximal soft classification change encountered while estimating the true weak stability.

It is evident that the (true) weak stability is often significantly lower than the (lower bound on) the uniform stability. In cases where the weak and uniform stabilities are similar, the **CM** algorithm performs poorly. The estimated weak stability behaves qualitatively the same as the true weak stability. When the uniform stability obtains lower values the algorithm performs very poorly. This may indicate that a good uniform stability is correlated with degenerated behavior (similar phenomenon was observed in [2]). In contrast, we see that very

⁷ We naively took $\alpha \in \{0.01, 0.5, 0.99\}$ and $\sigma \in \{0.1, 0.2, 0.3, 0.4, 0.5, 1, 2\}$ and these were our first and only choices.

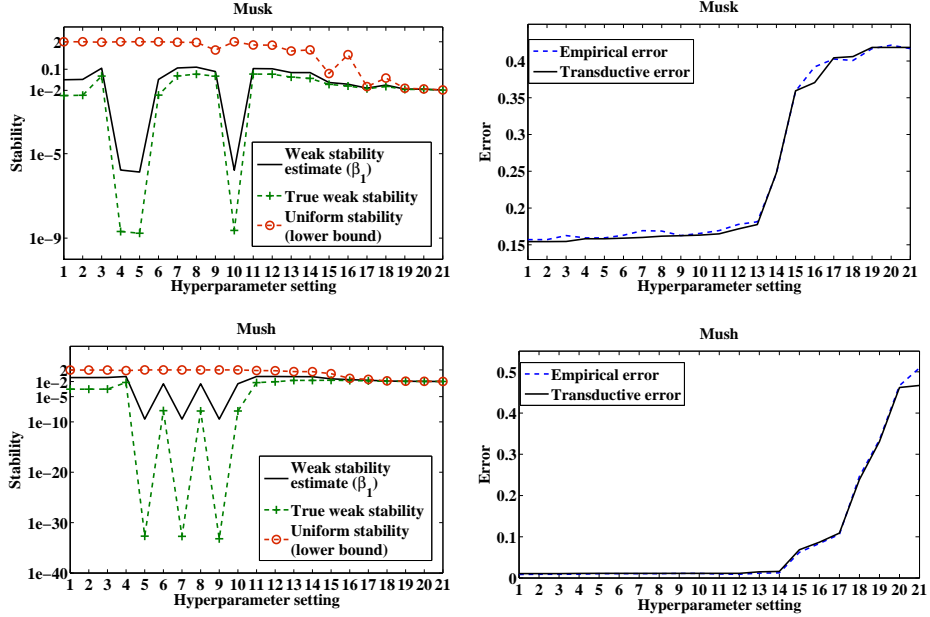


Fig. 1. Stability estimates (left) and the corresponding empirical/true errors (right) for musk and mush datasets.

good weak stability can coincide with very high performance. Finally, we note that these graphs do not demonstrate that good weak stability is proportional to low discrepancy between the empirical and true errors.

7 Concluding Remarks

This paper has presented new error bounds for transductive learning. The bounds are based on novel definitions of uniform and weak transductive stability. We have also shown that weak transductive stability can be bounded with high confidence in a data-dependent manner and demonstrated the application of this estimation routine on a known transductive algorithm. As far as we know this is the first attempt to generate truly data-dependent high confidence stability estimates based on all available information including the labeled samples.

We note that similar risk bounds based on weak stability can be obtained for induction. However, the adaptation of Definition 3 to induction (see also inductive definitions of weak stability in [10, 12, 15]) depends on the probability space of training sets, which is unknown in general. This prevents the estimation of weak stability using our method.

As discussed, to derive stability bounds with sufficient confidence our stability estimation routine is required to run in $\Omega(m^4(m+u))$ time, which precluded, at this stage, an empirical evaluation of our bounds. In future work we will attempt to overcome this obstacle by tightening our bound, perhaps using the

techniques from [13, 17]. A second direction would be to develop a more suitable weak stability definition. We also plan to consider other known transductive algorithms and develop for them a suitable implementation of the `diff` oracle.

References

1. K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19:357–367, 1967.
2. M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, pages 624–638, 2004.
3. A. Blum and J. Langford. PAC-MDL Bounds. In *COLT*, pages 344–357, 2003.
4. O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.
5. P. Derbeko, R. El-Yaniv, and R. Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *Journal of Artificial Intelligence Research*, 22:117–142, 2004.
6. L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer Verlag, New York, 1996.
7. G.R. Grimmett and D.R. Stirzaker. *Probability and Random Processes*. Oxford Science Publications, 1995. Second edition.
8. T.M. Huang and V. Kecman. Performance comparisons of semi-supervised learning algorithms. In *ICML Workshop “Learning with Partially Classified Training Data”*, pages 45–49, 2005.
9. D. Hush, C. Scovel, and I. Steinwart. Stability of unstable learning algorithms. Technical Report LA-UR-03-4845, Los Alamos National Laboratory, 2003.
10. M. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.
11. S. Kutin. Extensions to McDiarmid’s inequality when differences are bounded with high probability. Technical Report TR-2002-04, University of Chicago, 2002.
12. S. Kutin and P. Niyogi. Almost-everywhere algorithmic stability and generalization error. In *UAI*, pages 275–282, 2002.
13. M. Ledoux. *The concentration of measure phenomenon*. Number 89 in Mathematical Surveys and Monographs. American Mathematical Society, 2001.
14. G.S. Manku, S. Rajagopalan, and B.G. Lindsay. Approximate medians and other quantiles in one pass and with limited memory. In *SIGMOD*, volume 28, pages 426–435, 1998.
15. S. Mukherjee, P. Niyogi, T. Poggio, and R. Rifkin. Statistical learning: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. Technical Report AI Memo 2002–024, MIT, 2004.
16. A. Rakhlin, S. Mukherjee, and T. Poggio. Stability results in learning theory. *Analysis and Applications*, 3(4):397–419, 2005.
17. M. Talagrand. *Majorizing measures: the generic chaining*. Springer Verlag, 2005.
18. V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer Verlag, New York, 1982.
19. V. N. Vapnik. *Statistical Learning Theory*. Wiley Interscience, New York, 1998.
20. D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2003.
21. X. Zhu, Z. Ghahramani, and J.D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.