Bayesian Transduction

Thore Graepel, Ralf Herbrich, and Klaus Obermayer
Department of Computer Science
Technical University of Berlin
Franklinstr. 28/29, 10587 Berlin, Germany
{*graepel2,ralfh,oby*}*@cs.tu-berlin.de*

Abstract

Transduction is an inference principle that takes a training sample and aims at estimating the values of a function at given points contained in the so-called working sample. Hence, transduction is a less ambitious task than induction which aims at inferring a functional dependency on the whole of input space. As a consequence, however, transduction provides a confidence measure on single predictions rather than classifiers, a feature particularly important for risk–sensitive applications. We consider the case of binary classification by linear discriminant functions (perceptrons) in kernel space. From the transductive point of view, the infinite number of perceptrons is boiled down to a finite number of equivalence classes on the working sample each of which corresponds to a polyhedron in parameter space. In the Bayesian spirit the posteriori probability of a labelling of the working sample is determined as the ratio between the volume of the corresponding polyhedron and the volume of version space. Then the maximum posteriori scheme recommends to choose the labelling of maximum volume. We suggest to sample version space by an ergodic billiard in kernel space. Experimental results on real world data indicate that Bayesian Transduction compares favourably to the well-known Support Vector Machine, in particular if the posteriori probability of labellings is used as a confidence measure to exclude test points of low confidence.

1 Introduction

According to Vapnik [9], *when solving a given problem one should avoid solving a more general problem as an intermediate step.* The reasoning behind this principle is that in order to solve the more general task, resources may be wasted or compromises may have to be made, which would not have been necessary for the solution of the problem at hand. A direct application of this common-sense principle reduces the more general problem of inferring a functional dependency on the whole of input space to the problem of estimating the values of a function at given points (working sample), a paradigm referred to as *transductive inference.* More formally, given a probability $P(\mathbf{x}, y)$ on the space of data $\mathcal{X} \times \{-1, +1\}$, a *training sample* $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$ is generated i.i.d. according to P. Additional k data points

$W = \{\mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_{\ell+k}\}$ are drawn: the *working sample*. The goal is to label the objects of the working sample W using a fixed set $\mathcal{H}$ of functions $f : \mathcal{X} \mapsto \{-1, +1\}$ so as to minimise a predefined loss. In contrast, *inductive inference*, aims at choosing a *single* function $f_\ell \in \mathcal{H}$ best suited to capture the dependency expressed by the unknown P. Obviously, if we have a transductive algorithm $\mathcal{A}(S, W)$ that assigns to each working sample W a set of labels given the training sample S, we can define a function $f_S : \mathcal{X} \mapsto \{-1, +1\}$ by $f_S(\mathbf{x}) = \mathcal{A}(S, \{\mathbf{x}\})$ and implicitly get a function as a result of the transduction algorithm. There are two crucial differences, however: i) $\mathcal{A}(S, \{\mathbf{x}\})$ is not restricted to select the same decision function f for each $\mathbf{x}$, ii) a transduction algorithm can give performance guarantees on particular labellings instead of functions. In practical applications this difference may be of great importance. After all, in risk sensitive applications (medical diagnosis, financial and critical control applications) it often matters to know how *confident* a given prediction is. In this case a general confidence measure of the classifier w.r.t. the whole input distribution would not provide the desired warranty at all. Note, that for linear classifiers some guarantee can be obtained by the margin [7] which in Section 4 we will demonstrate to be too coarse a confidence measure. The idea of transduction was put forward in [8], where also first algorithmic ideas can be found. Later [1] suggested an algorithm for transduction based on linear programming and [3] highlighted the need for confidence measures in transduction.

The paper is structured as follows: A Bayesian approach to transduction is formulated in Section 2. In Section 3 the function class of kernel perceptrons is introduced to which the Bayesian transduction scheme is applied. For the estimation of volumes in parameter space we present a kernel billiard as an efficient sampling technique. Finally, we demonstrate experimentally in Section 4 how the confidence measure for labellings helps Bayesian Transduction to achieve low generalisation error at a low rejection rate of test points and thus to outperform Support Vector Machines (SVMs).

2 Bayesian Transductive Classification

For the sake of simplicity let us assume that the data set is *perfect* in the sense that there exists a function f^* in the space $\mathcal{H}$ such that for all $\mathbf{x} \in \mathcal{X} \Rightarrow y = f^*(\mathbf{x})$. Thus the model of how S and W are generated is as follows [8]

1. The training sample $S_{|\mathcal{X}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ and the working sample $W = \{\mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_{\ell+k}\}$ are generated i.i.d. according to some unknown $P_{\mathcal{X}}(\mathbf{x})$.
2. According to a prior $P_{\mathcal{H}}(f)$ on $\mathcal{H}$ a particular f^* is selected and S is complemented to $\{(\mathbf{x}_1, f^*(\mathbf{x}_1)), \ldots, (\mathbf{x}_\ell, f^*(\mathbf{x}_\ell))\}$.

From the knowledge of $\mathcal{H}$ and S, one can define the so–called *version–space* as the set of functions that is consistent with the training sample

$$\mathcal{V}(S) = \{f | f(\mathbf{x}_i) = y_i; \quad (\mathbf{x}_i, y_i) \in S; \quad i = 1, \ldots, \ell\}\,. \tag{1}$$

Moreover, given a *prior* $P_{\mathcal{H}}(f)$ one can derive the posterior probability on labelling $\mathbf{b} \in \{-1, +1\}^k$ of the working sample[1]

$$P(\mathbf{b}|S) = \frac{\int_{\mathcal{V}(S)} \prod_{i=1}^{k} \delta(f(\mathbf{x}_{\ell+i}), b_i)\; dP_{\mathcal{H}}(f)}{\int_{\mathcal{V}(S)}\; dP_{\mathcal{H}}(f)} \qquad \delta(\hat{y}, y) = (1 - \hat{y}y)/2\,. \tag{2}$$

[1] Note, that the number of different labellings $\mathbf{b}$ implementable by $\mathcal{H}$ is bounded above by the value of the growth function $\Pi_{\mathcal{H}}(|W|)$ [8, p. 321].

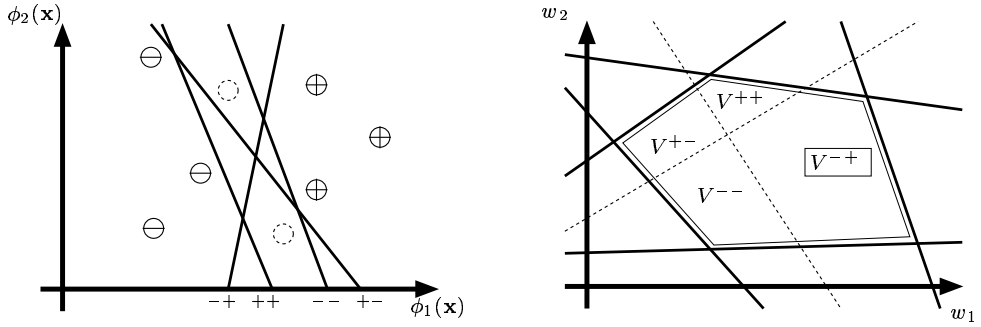

Figure 1: Schematic view of data space (left) and parameter space (right) for a classification toy example. Using the duality given by $\langle \mathbf{w}, \phi(\mathbf{x}) \rangle_{\mathcal{F}} = 0$ data points on the left correspond to hyperplanes on the right, while hyperplanes on the left can be thought of as points on the right.

This rather arcane definition of $P(\mathbf{b}|S)$ measures the volume of functions consistent with the training sample S *and* the labelling $\mathbf{b}$ on the working sample W normalised by the volume of functions consistent with S alone (see Figure 1). Besides the fact that $P_{\mathcal{H}}(f)$ has to be known in advance the integrals are computationally intractable for most function spaces $\mathcal{H}$ and priors $P_{\mathcal{H}}(f)$. However, in the special case of linear decision functions and a uniform prior over weight vectors $\mathbf{w}$ an *ergodic billiard* [6] in version space can be used to estimate the posterior probability of labellings (see Section 3.2).

Given the posterior probability of the labellings $\mathbf{b}$, in order to arrive at a risk minimal decision w.r.t. the labelling we need to define a loss function $l(\mathbf{b}, \mathbf{b}')$ between labellings and minimise its expectation under the posterior $P(\mathbf{b}|S)$,

$$R(\mathbf{b}; S) = \sum_{\{\mathbf{b}'\}} l(\mathbf{b}, \mathbf{b}') P(\mathbf{b}'|S)\,, \tag{3}$$

where the summation runs over all the 2^k possible labellings $\mathbf{b}$ of the working sample. Let us consider two scenarios:

1. A 0–1–loss on the exact labelling $\mathbf{b}$, i.e. for two labellings $\mathbf{b}$ and $\mathbf{b}'$

$$l^c(\mathbf{b}, \mathbf{b}') = \prod_{i=1}^{k} (1 - \delta(b_i, b_i'))\,. \tag{4}$$

   In this case the maximum posteriori estimator, i.e. the labelling of the highest posterior probability, minimises the risk. This loss is appropriate if the goal is to *exactly* identify a combination of labels, e.g. the combination of handwritten digits defining a postal zip code on an envelope.

2. A 0–1–loss on the single labels b_i, i.e. for two labellings $\mathbf{b}$ and $\mathbf{b}'$

$$l^s(\mathbf{b}, \mathbf{b}') = \sum_{i=1}^{k} (1 - \delta(b_i, b_i'))\,. \tag{5}$$

   Due to the independent treatment of the loss at working sample points the risk (3) is minimised by the labelling of maximum marginal posterior of the labels, i.e.

$$\hat{b}_i = \text{argmax}_{y \in \{-1,+1\}} \left[\sum_{\{\mathbf{b}: b_i = y\}} P(\mathbf{b}|S) = \frac{\int_{\mathcal{V}(S)} \delta(f(\mathbf{x}_{\ell+i}), y)\; dP_{\mathcal{H}}(f)}{\int_{\mathcal{V}(S)}\; dP_{\mathcal{H}}(f)} \right].$$

Thus in the case of the pointwise loss (5) a working sample of $k > 1$ point does not offer any advantages over larger working samples w.r.t. the Bayes-optimal decision. Since this corresponds to the standard classification setting, we will restrict ourselves to working samples of size $k = 1$, i.e. to one working point $\mathbf{x}_{\ell+1}$.

3 Bayesian Transduction by Volume

3.1 The Kernel Perceptron

We consider transductive inference for the class of kernel perceptrons. The decision functions are given by

$$f(\mathbf{x}) = \text{sign}\left(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle_{\mathcal{F}}\right) = \text{sign}\left(\sum_{i=1}^{\ell} \alpha_i k(\mathbf{x}_i, \mathbf{x})\right) \qquad \mathbf{w} \in \mathcal{F}, \boldsymbol{\alpha} \in \mathbb{R}^{\ell}\,, \qquad (6)$$

where the mapping $\phi : \mathcal{X} \mapsto \mathcal{F}$ maps from input space $\mathcal{X}$ to a *feature space* $\mathcal{F}$ completely determined by the inner product function (*kernel*) $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ (see [9, 10]). In the following it is assumed that $\mathcal{F}$ is restricted to the surface of a hypersphere $\|\phi(\mathbf{x})\|_{\mathcal{F}} = 1$. Given a training sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ we can define the version space — the set of all perceptrons compatible with the training data — as in (1) having the additional constraint $\|\mathbf{w}\|_{\mathcal{F}} = 1$ ensuring uniqueness. In order to obtain a prediction on the label b_1 of the working point $\mathbf{x}_{\ell+1}$ we note that $\mathbf{x}_{\ell+1}$ may bisects the volume V of version space into two sub–volumes V^+ and V^-, where the perceptrons in V^+ would classify $\mathbf{x}_{\ell+1}$ as $b_1 = +1$ and those in V^- as $b_1 = -1$. The ratio $p^+ = V^+/V$ is the posterior probability of the labelling $b_1 = +1$ given a uniform prior $P_{\mathcal{F}}(\mathbf{w})$ over $\mathbf{w}$ and the class of kernel perceptrons, accordingly for $b_1 = -1$. Already Vapnik in [8, p. 323] noticed that it is troublesome to estimate sub–volumes of version space. As the solution to this problem we suggest to use a billiard algorithm.

3.2 Kernel Billiard for Volume Estimation

The method of playing billiard in version space was first introduced by Rujan [6] for the purpose of estimating its centre of mass and consequently refined and extended to kernel spaces by [4]. For Bayesian Transduction the idea is to bounce the billiard ball in version space and to record how much time it spends in each of the sub–volumes of interest. Under the assumption of ergodicity [2] w.r.t. the uniform measure the accumulated flight times for each sub–volume are proportional to the sub–volume itself.

Since the trajectory is located in $\mathcal{F}$ each position $\mathbf{w}$ and direction vector $\mathbf{v}$ of the ball can be expressed as linear combinations of the $\phi(\mathbf{x}_i)$, i.e.

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i \phi(\mathbf{x}_i) \quad \mathbf{v} = \sum_{i=1}^{\ell} \beta_i \phi(\mathbf{x}_i) \quad \langle \mathbf{w}, \mathbf{v} \rangle_{\mathcal{F}} = \sum_{i,j=1}^{\ell} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j)$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ are real vectors with ℓ components and fully determine the state of the billiard. The algorithm for the determination of the label $\hat{b}_1$ of $\mathbf{x}_{\ell+1}$ proceeds as follows:

1. Initialise the starting position $\mathbf{w}_0$ in $\mathcal{V}(S)$ using any kernel perceptron algorithm that achieves zero training error (e.g., SVM [9]). Set $V^+ = V^- = 0$.

2. Find the closest boundary of $\mathcal{V}(S)$ starting from current $\mathbf{w}$ into direction $\mathbf{v}$, where the flight times τ_j for all points including $\mathbf{x}_{\ell+1}$ are determined using

$$\tau_j \quad = \quad -\frac{\langle \mathbf{w}, \phi(\mathbf{x}_j) \rangle_{\mathcal{F}}}{\langle \mathbf{v}, \phi(\mathbf{x}_j) \rangle_{\mathcal{F}}} \,. \tag{7}$$

The smallest positive flight time $\tau_m = \min_{j:\tau_j>0} \tau_j$ in kernel space corresponds to the closest data point boundary $\phi(\mathbf{x}_m)$ on the hypersphere. Note, that if $\tau_m \to \infty$ we randomly generate a direction $\mathbf{v}$ pointing *towards* version space, i.e. $y\langle \mathbf{v}, \phi(\mathbf{x}) \rangle_{\mathcal{F}} > 0$ assuming the last bounce was at $\phi(\mathbf{x})$.

3. Calculate the ball's new position $\mathbf{w}'$ according to

$$\mathbf{w}' = \frac{\mathbf{w} + \tau_m \mathbf{v}}{\|\mathbf{w} + \tau_m \mathbf{v}\|_{\mathcal{F}}} \,. \tag{8}$$

Calculate the distance $t_i^y = \|\mathbf{w} - \mathbf{w}'\|_{\text{sphere}} = \arccos\left(1 - \|\mathbf{w} - \mathbf{w}'\|_{\mathcal{F}}^2/2\right)$ on the hypersphere and add it to the volume estimate V^y corresponding to the current labelling $y = \text{sign}(\langle \mathbf{w} + \mathbf{w}', \phi(\mathbf{x}_{\ell+1}) \rangle_{\mathcal{F}})$. If the test point $\phi(\mathbf{x}_{\ell+1})$ was hit, i.e. $m = \ell + 1$, keep the old direction vector $\mathbf{v}$. Otherwise update to the reflection direction $\mathbf{v}'$,

$$\mathbf{v}' = \mathbf{v} - 2\langle \mathbf{v}, \phi(\mathbf{x}_m) \rangle_{\mathcal{F}} \; \phi(\mathbf{x}_m) \,. \tag{9}$$

Go back to step 2 unless the stopping criterion (12) is met.

The estimators of the posteriori probability of the labellings are then given by $\hat{p}^+ = V^+/(V^+ + V^-)$ and $\hat{p}^- = V^-/(V^+ + V^-)$. Thus, the algorithm outputs $\hat{b}_1$ with confidence $\hat{c}_1^{\text{trans}}$ according to

$$\hat{b}_1 \quad = \quad \text{argmax}_{y \in \{-1,+1\}} \hat{p}^y \,, \tag{10}$$

$$\hat{c}_1^{\text{trans}} \quad = \quad 2(\max(\hat{p}^+, \hat{p}^-) - 0.5) \,. \tag{11}$$

Note, that the Bayes point [4] is the optimal approximation of (10) by one perceptron. The well known SVM can be viewed as an approximation of the Bayes point by the centre of the largest hypersphere in version space. Thus, treating the real valued output $c_1^{\text{ind}} = |f(\mathbf{x}_{\ell+1})|$ of SVM classifiers as a confidence measure is also an approximation of (11). The consequences will be demonstrated experimentally in the following section.

Disregarding the issue of mixing time [2] we assume for the stopping criterion that the fraction p_i^+ of time t_i^+ spent in volume V^+ on trajectory i of length $(t_i^+ + t_i^-)$ is a random variable having expectation p^+. Hoeffding's inequality [5] bounds the probability of deviation from the expectation p^+ by more than ϵ,

$$\Pr\left\{ \frac{1}{n} \sum_{i=1}^{n} (p_i^+ - p^+) \geq \epsilon \right\} \leq \exp(-2n\epsilon^2) \stackrel{\text{def}}{=} \delta \,. \tag{12}$$

Thus if we want the deviation ϵ from the true posterior to be less than $\epsilon < 0.05$ with probability at least $1 - \delta = 0.99$ we need approximately $n \approx 1000$ bounces.

4 Experimental Results

In our experiments we focused on the confidence $\hat{c}_1^{\text{trans}}$ Bayesian Transduction provides together with the prediction $\hat{b}_1$ of the label. We used the UCI[2] data sets

[2] UCI University of California Irvine: Machine Learning Repository

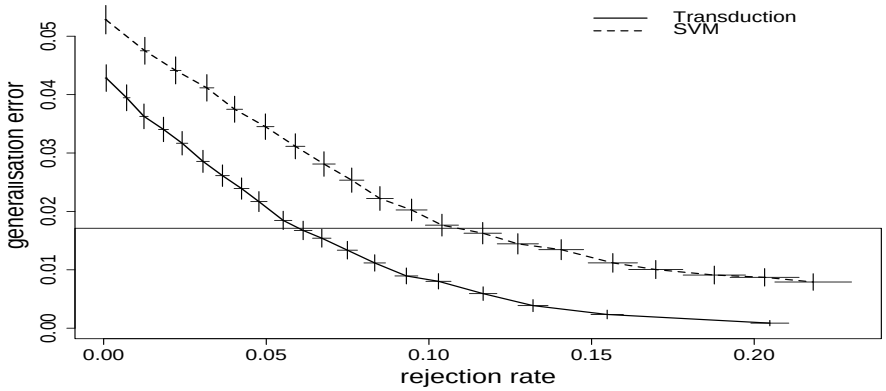

Figure 2: Generalisation error vs. rejection rate for Bayesian Transduction and SVMs for the `thyroid` data set ($\sigma = 3$). The error bars in both directions indicate one standard deviation of the estimated means.

`thyroid` and `heart` because they are medical applications for which the confidence of single predictions is particularly important. Also a high rejection rate due to a too conservative confidence measure often means that a human expert needs to be consulted, which may incur considerable costs. We trained a Support Vector Machine using RBF kernels $k(\mathbf{x}, \mathbf{x}') = \exp(\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2)$ on 100 different training samples obtained by random 60%:40% splits of the whole data set. We calculated the margin c_1^{ind} of each test point as a confidence measure of SVM classifiers. For comparison we determined the labels $\hat{b}_1$ and resulting confidences $\hat{c}_1^{\text{trans}}$ using the Bayesian Transduction algorithm (see Section 3). To check the reliability of the confidence measures we calculated the rejection rates and generalisation errors of the non–rejected points as functions of a confidence threshold θ. Since the rejection for the Bayesian Transduction was in both cases higher than for SVMs at the same level θ we determined $\theta_{\max}$ which achieves the same rejection rate for the SVM confidence measures as Bayesian Transduction achieves at $\theta = 1$ (`thyroid`: $\theta_{\max} = 2.15$, `heart`: $\theta_{\max} = 1.54$). Thus we could compare both algorithms in a generalisation error versus rejection rate plot. The results for the two data sets are depicted in Figures 2 and 3. In the `thyroid` example Figure 2 one can see that $\hat{c}_1^{\text{trans}}$ is indeed an appropriate indicator of confidence: At a rejection rate of approximately 20% the generalisation error approaches zero at minimal variance. Moreover, for any desired generalisation error Bayesian Transduction needs to reject significantly less examples of the test set as compared to SVM classifiers, e.g. 4% less at 2.3% generalisation error. The results of the `heart` data set show even more pronounced characteristics w.r.t. to the rejection rate. Note, that the confidence measure of transduction cannot capture the effects of noise in the data which leads to a generalisation error of 16.4% even at maximal rejection $\theta = 1$ corresponding to the Bayes error under the given function class.

5 Conclusions and Future Work

In this paper we reviewed the general idea of transduction and introduced Bayesian Transduction. The required volume estimates for kernel perceptrons in version space are performed by an ergodic billiard in kernel space. Most importantly, transduction

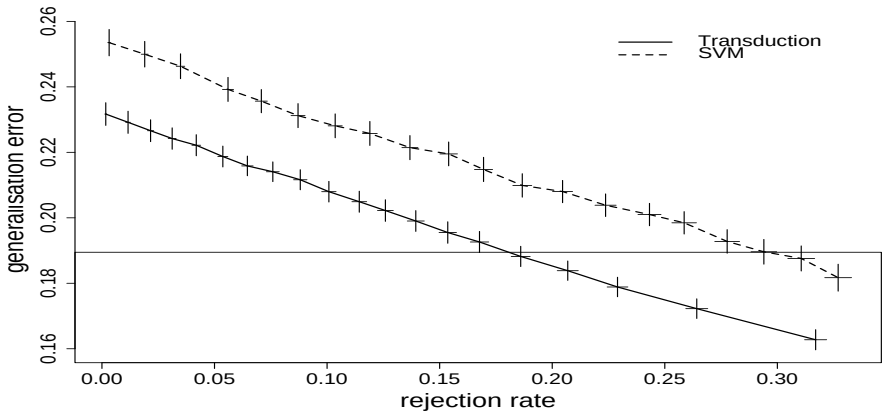

Figure 3: Results for the `heart` data set ($\sigma = 10$). For details see Figure 2.

not only determines the label of a given point but also returns a confidence measure of the classification in the form of the posterior probability of the label under the model. Using this confidence measure to reject test examples then lead to improved generalisation error over SVMs. As a generalisation, the billiard algorithm can be extended to the case of non-zero training error by allowing the ball to penetrate walls, a property that is captured by adding a constant λ to the diagonal of the kernel matrix [4]. Further research will aim at the consideration of unbalanced data sets and of data-dependent priors over parameters.

Acknowledgements

This project was funded by Technical University Berlin via FIP 13/41.

References

[1] K. Bennett. *Advances in Kernel Methods — Support Vector Learning*, chapter 19, Combining Suuport Vector and Mathematical Programming Methods for Classification, pages 307–326. MIT Press, 1998.

[2] I. Cornfeld, S. Fomin, and Y. Sinai. *Ergodic Theory*. Springer Verlag, 1982.

[3] A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of Uncertainty in AI*, pages 148–155, Madison, Wisconsin, 1998.

[4] R. Herbrich, T. Graepel, and C. Campbell. Bayesian learning in reproducing kernel hilbert spaces – the usefulness of the bayes point. Technical report, Technical University Berlin, 1999. http://stat.cs.tu-berlin.de/~ralfh/publications.html.

[5] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[6] P. Rujàn. Playing billiard in version space. *Neural Computation*, 9:99–122, 1997.

[7] J. Shawe-Taylor. Confidence estimates of classification accuracy on new examples. Technical report, Royal Holloway, University of London, 1996. NC2–TR–1996–054.

[8] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, 1982.

[9] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[10] G. Whaba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, 1990.