PAPER

Polynomial Learnability of Stochastic Rules with Respect to the KL-Divergence and Quadratic Distance

Naoki ABE[†], **Jun-ichi TAKEUCHI**[†], *and* **Manfred K. WARMUTH**[††], ***Nonmembers***

SUMMARY We consider the problem of efficient learning of probabilistic concepts (p-concepts) and more generally stochastic rules in the sense defined by Kearns and Schapire [6] and by Yamanishi [18]. Their models extend the PAC-learning model of Valiant [16] to the learning scenario in which the target concept or function is stochastic rather than deterministic as in Valiant's original model. In this paper, we consider the learnability of stochastic rules with respect to the classic 'Kullback-Leibler divergence' (KL divergence) as well as the quadratic distance as the distance measure between the rules. First, we show that the notion of polynomial time learnability of p-concepts and stochastic rules with fixed range size using the KL divergence is in fact equivalent to the same notion using the quadratic distance, and hence any of the distances considered in [6] and [18]: the quadratic, variation, and Hellinger distances. As a corollary, it follows that a wide range of classes of p-concepts which were shown to be polynomially learnable with respect to the quadratic distance in [6] are also learnable with respect to the KL divergence. The sample and time complexity of algorithms that would be obtained by the above general equivalence, however, are far from optimal. We present a polynomial learning algorithm with reasonable sample and time complexity for the important class of convex linear combinations of stochastic rules. We also develop a simple and versatile technique for obtaining sample complexity bounds for learning classes of stochastic rules with respect to the KL-divergence and quadratic distance, and apply them to produce bounds for the classes of probabilistic finite state acceptors (automata), probabilistic decision lists, and convex linear combinations.

key words: *PAC-learning, KL-divergence, quadratic-distance, stochastic rules, p-concepts*

1. Introduction

We consider the problem of computationally efficient learning of probabilistic concepts (p-concepts) and more generally stochastic rules in the sense defined by Kearns and Schapire [6], and by Yamanishi [18]. These two models extended the PAC-learning model of Valiant [16] to the learning scenario in which the target concept or function is probabilistic, rather than deterministic as in Valiant's original model. We will refer to these models collectively as the probabilistic PAC learning models. In a deterministic model, the target to be learned is a function from a domain X into a range Y.

In a probabilistic model* the target to be learned is a function that *probabilistically maps* elements of X to Y, i.e. a conditional probability distribution over Y given elements of X. We call such a map a *stochastic rule* if $|Y| \geq 2$ and a *p-concept* if $|Y| = 2$. The learning algorithm is given examples (elements of $X \times Y$), drawn according to an arbitrary but fixed distribution over X and probabilistically labeled by the target stochastic rule. After a feasibly small number of examples, it is required to output a hypothesis, which is sufficiently close to the target stochastic rule with high probability.

In a learning model for stochastic rules, the choice of what notion of 'distance' is to be used as a measure of closeness of a hypothesis stochastic rule to the target stochastic rule is an important decision that may affect the notion of learnability defined in the model. Kearns and Schapire used the 'quadratic distance,' whereas Yamanishi considered the Hellinger and variation distances. In this paper, we mainly consider the Kullback-Leibler divergence (KL-divergence)**, and the quadratic distance. Of the four distance measures mentioned above, the KL-divergence and the quadratic distance share the following property: The distance between two stochastic rules c and h can be expressed as the difference between the expected 'loss' associated with h minus the expected loss for c. Specifically, the KL-divergence, d_{KL} can be defined as

$$d^D_{KL}(c,h) = E_D[E_c[L_h(x,y) - L_c(x,y)]],$$

where D is an arbitrary distribution over X, c is the target stochastic rule, and the loss function $L_f : X \times Y \to \mathbf{R}$ denotes the *log loss* of the stochastic rule f, defined as $\ln(1/f(y|x))$. Note that we used E_D and E_c to denote the expectations with respect to D and c, respectively. Similarly, the quadratic distance d_Q is equal to

$$d^D_Q(c,h) = E_D[E_c[Q_h(x,y) - Q_c(x,y)]],$$

Manuscript received January 4, 2000.
Manuscript revised July 26, 2000.

[†]The authors are with the Theory NEC Laboratory, RWCP (Real World Computing Partnership), c/o Computer & Communication Media Resarch, NEC Corporation, Kawasaki-shi, 216–8555 Japan.

[††]The author is with Computer Science Department, University of California at Santa Cruz, CA 95064, USA.

*Non-conditional probability distributions are degenerate stochastic rules with a common range of size one. Thus the PAC learning models for non-conditional probability distributions considered by Angluin and Laird [1] and Abe and Warmuth [3] are essentially special cases of the model studied here.

**Abe and Warmuth also used the KL-divergence in their PAC learning model for (non-conditional) probability distributions [3].

where $Q_f : X \times Y \to \mathbf{R}$ is the *quadratic loss* of f, defined as[†]

$$Q_f(x,y) = (1 - f(y|x))^2 + \sum_{y' \in Y - \{y\}} f(y'|x)^2.$$

The problems of learning stochastic rules with respect to these two measures, therefore, fit in the general framework of Haussler [4], in which the goal of 'learning' is to probably approximately minimize the 'loss' of its hypothesis. This property is particularly desirable since it also suggests a natural strategy for designing an efficient learning algorithm: Find a hypothesis which (approximately) minimizes the empirical loss in the given sample.

The KL-divergence is especially attractive because of its close relationship with the maximum likelihood estimation: It is well known that a hypothesis minimizing the KL-divergence with respect to the *empirical* distribution also maximizes the likelihood of the given sample. The KL-divergence is also known to be a relatively strict notion of distance (both for stochastic rules and for distributions). The KL-divergence bounds from above $\ln 2$ times the Hellinger distance [18], a half times the square of the variation distance [8] and a half times the quadratic distance (see Appendix B). Also the latter three distance measures are known to bound one another within a polynomial (cf. [14], [18]), but there are cases when the three measures are bounded and the KL-divergence is unbounded. This is because the log loss function is unbounded, and in fact the unbounded range of the log loss function tends to make deriving upper bounds on sample complexity harder. For example, Yamanishi [18] leaves sample complexity bounds with respect to the KL-divergence unresolved. The use of log loss, in place of quadratic loss, also tends to make the task of designing efficient learning algorithms challenging. While it is possible in some cases to solve analytically for the hypothesis minimizing the empirical quadratic loss, this is often not possible for the log loss. For example, while Kearns and Schapire [6] show how to efficiently learn the important class of linear combinations of a finite set of p-concepts, taking advantage of the properties of the quadratic distance, it is not immediately clear how one can extend such a result for the KL-divergence.

In this paper, we manage to show a number of positive results with respect to the KL-divergence. To circumvent the problem caused by the unbounded range of the log loss function, we use a technique we call 'ϵ-Bayesian averaging.'[††] ϵ-Bayesian averaging gives us a way of obtaining from an arbitrary stochastic rule f another stochastic rule $f^{(\epsilon)}$ called its *ϵ-Bayesian shift* such that $f^{(\epsilon)}$ is ϵ close to f with respect to the KL-divergence, and the range of its log loss function $L_{f^{(\epsilon)}}$ is polynomially bounded in the relevant parameters of the learning problem. First, we begin by showing that in fact the notion of PAC-learnability with respect to the KL-divergence turns out to be *equivalent* to the notion of PAC-learnability defined with respect to the quadratic distance (and the variation and Hellinger distances). This is true even though the quadratic distance does not bound the KL-divergence within any polynomial: we exhibit (in Sect. 3) a kind of algorithm transformation that shows that any polynomial time learning algorithm with respect to the quadratic distance can be used to obtain one for the KL-divergence. The key idea is to obtain a highly accurate hypothesis with respect to the quadratic distance and then use the ϵ-Bayesian averaging technique to obtain one which is accurate with respect to the KL-divergence. To be more precise, the equivalence result requires that the learning algorithm be able to use an arbitrary *polynomial time evaluable* hypothesis class,[†††] not necessarily equal to the target class.

As corollaries of this result, a number of learnability results with respect to the KL divergence follow immediately from Kearns and Schapire's learnability results with respect to the quadratic distance, including the learnability of the class of linear combinations of a fixed finite set of p-concepts. The sample and time complexity bounds one can obtain via the above general algorithm transformation are, however, far from optimal in most cases. An interesting question, therefore, is whether one can obtain tighter bounds on sample and/or time complexity for learning specific classes of stochastic rules with respect to the KL divergence.

In Sect. 4 we apply the ϵ-Bayesian technique to obtain upper bounds on the sample complexity for robustly[††††] learning classes of stochastic rules with respect to the KL divergence (as well as the quadratic distance). We apply this technique to the following classes: probabilistic automata as acceptors, probabilistic decision lists, and convex linear combinations of a fixed finite set of stochastic rules. Elegant techniques for showing bounds on sample complexity for uniform convergence, using the notion of 'combinatorial dimension' and other related dimensions, were formulated by Pollard [12] and Haussler [4], and were applied

[†]For $Y = \{0,1\}$, if we denote $f(1|x)$ by $f(x)$ and implicitly define $f(0|x)$ as $1 - f(x)$, then $Q_f(x,y)$ equals $2(y - f(x))^2$ as is defined in [6]. Note also that for any stochastic rules c and h, $E_c[Q_h(x,y) - Q_c(x,y)] = \sum_{y \in Y}(h(y|x) - c(y|x))^2$ [6].

[††]A technique similar to the ϵ-Bayesian averaging was used to obtain a sample complexity upper bound for learning (non-conditional) distributions by Abe and Warmuth [3].

[†††]A stochastic rule representation class H is said to be polynomially evaluable, if there exists a polynomial time algorithm that evaluates the value $h(y|x)$ for any $h \in H$ for any input x, y.

[††††]'Robust learning' requires that the algorithm's output hypothesis be within ϵ of the optimal hypothesis in the class of interest, even when the target stochastic rule does not belong to the class.

by Kearns and Schapire [6] to the problem of learning p-concepts. However, the combinatorial dimension of many p-concept classes of interest, including the class of probabilistic acceptors, are hard to compute or too large to be of much use. By using a more direct approach which is based on Pollard's "direct approximation" method [10], we are able to obtain reasonable upper bounds on sample complexity for such cases.

In Sect. 5 we present a polynomial time robust learning algorithm for the class of convex combinations of stochastic rules (linear combinations such that the coefficients sum to one) with a good sample complexity with respect to the KL divergence. The learning algorithm we exhibit is essentially a gradient descent type algorithm which approximately minimizes the empirical log loss of the input sample. The algorithm uses as hypothesis class, not the class of convex combinations of the original p-concepts, but the convex combinations of the ϵ-Bayesian shifts of those p-concepts. The use of ϵ-Bayesian shifts is crucial for the time complexity analysis of the algorithm. With respect to the quadratic distance, it is straightforward to show that the same class is also polynomially learnable.

Our algorithm for learning convex linear combinations with respect to the KL-divergence is similar to the "gradient projection algorithm" of Rosen and its extensions [11], [13] for optimizing a general convex objective function with linear constraints. For these methods, proofs of asymptotic convergence to the global minimum for a general convex objective function are known, but no polynomial bounds on the required number of iterations that applies to our problem. Here we prove fast convergence for our particular objective function for learning linear combinations of stochastic rules. Also related to our work is the more recent work by Helmbold et al [5], carried out subsequently to our original work [2], which has analyzed the 'cumulative losses' of a number of iterative algorithms for learning convex combinations (mixtures), through the optimizing iterations. Their results can be used to derive bounds on the computation time for learning convex combinations of order $O(1/\epsilon^6)$. Here we prove bounds on the computation time of our algorithms of order $O(1/\epsilon^5)$.

2. Preliminaries

Stochastic rules over domain X and range Y are conditional probabilities over Y given elements of X [18]. When the size of Y is equal to two stochastic rules are called *p-concepts*. In our examples we use the discrete domains $X = \Sigma^n$ or $X = \Sigma^{\leq n} = \cup_{0 \leq i \leq n} \Sigma^i$, for some alphabet Σ and some natural number n. In the sequel, we will write X_n for X in such cases, using the subscript to indicate the domain size. When $X_n = \{0,1\}^n$ the stochastic rules are Boolean. If C is a class of representations of stochastic rules, we also let C denote ambiguously the class of stochastic rules represented by C. The *size* of a stochastic rule is a parameter that measures its complexity, usually the number of real valued parameters in the simplest representation for it. We let $C_{n,s}$ denote the subclass of C of size at most s, defined over Σ^n or $\Sigma^{\leq n}$.

All hypothesis classes of stochastic rules used in this paper must be 'polynomially evaluable,' in the sense defined as follows. An evaluation algorithm for a hypothesis class H receives elements $x \in X$, $y \in Y$ and a representation $h \in H$ and outputs the conditional probability $h(y|x)$. Such an algorithm is said to be a 'polynomial evaluation algorithm' if its running time is polynomial in the length of x and y and the size of h, and H is said to be 'polynomially evaluable' if there exists a polynomial evaluation algorithm for H. In our learning scenario, the learning algorithm is given as input an accuracy parameter ϵ, a confidence parameter δ, an upper bound n on the example length, an upper bound s on the size of the target stochastic rule, and the range size $|Y|$. The algorithm is also equipped with an oracle with which to access random examples drawn according to the product $D \cdot c$ of an arbitrary, unknown distribution D over X_n and the target stochastic rule c over Y given X_n. Here, $D \cdot c$ is the distribution over domain $X \times Y$ defined by $D \cdot c\,(x,y) = D(x)c(y|x)$.

A learning algorithm is said to *polynomially (PAC-) learn* a class of stochastic rules C *in terms of* a (polynomially evaluable) hypothesis class H with respect to a distance measure d, if for all $\epsilon > 0$, $\delta > 0$, s and n, for all distributions D over X_n, for all target stochastic rules $c \in C_{n,s}$ over $X \times Y$, using the above protocol, the algorithm terminates in time polynomial in $1/\epsilon$, $1/\delta$, s, n, and $|Y|$, and outputs a representation of a stochastic rule $h \in H$ satisfying

$$d(c,h) \leq \epsilon \qquad \text{with probability at least } 1-\delta.$$

A class C of stochastic rule representations is said to be *polynomially (PAC-) learnable*, if C is polynomially learnable in terms of some polynomially evaluable class of stochastic rules.

When an arbitrary target stochastic rule c, not necessarily in the target class C, is allowed, we would like the learning algorithm to output a hypothesis that is "as close as possible" to the target stochastic rule, motivating the following modified definition. The learning algorithm is said to *polynomially and robustly learn* a class of stochastic rules C *with hypothesis class* H if when given the same parameters as above and access to examples drawn according to $D \cdot c$, where *both* D and c are *arbitrary*, the algorithm, using polynomial sample size and polynomial time, outputs a hypothesis $h \in H$ satisfying

$$d(c,h) - \min\{d(c,c') : c' \in C_{n,s}\} \leq \epsilon \qquad \text{with probability at least } 1-\delta.$$

Note that the input s here is an upper bound on the

size of the stochastic rules in C to be considered as candidates for the best possible approximation (of the target stochastic rule), against which the approximate optimality of the output hypothesis is to be measured. A class C of stochastic rule representations is said to be *robustly polynomially learnable*, if there exists an algorithm which polynomially and robustly learns C in terms of some polynomially evaluable class of stochastic rule representations. A class C of stochastic rule representations is said to be *properly* robustly polynomially learnable, if C is robustly polynomially learnable in terms of C itself.

The above model of 'robust learning' [2] has been generalized by the 'agnostic learning model' introduced by Kearns, Schapire and Sellie [7]. In their model, a learning scenario is parameterized by three different classes of stochastic rules; the target class, the touchstone class, and the hypothesis class. Our definition of 'robust learnability of C in terms of H' can be obtained by setting the target class, in their model, to be the set of all possible stochastic rules, the touchstone class to be C, and the hypothesis class to be H.

The distance measures between stochastic rules we consider in this paper are defined below†. In the definitions to follow, c is the target stochastic rule, h is a hypothesis stochastic rule, and D is an arbitrary distribution over X.

$$d^D_{KL}(c,h) = \sum_{x\in X} D(x) \sum_{y\in Y} c(y|x) \ln \frac{c(y|x)}{h(y|x)} \quad \text{and}$$

$$d^D_Q(c,h) = \sum_{x\in X} \sum_{y\in Y} D(x)(c(y|x) - h(y|x))^2. \qquad (1)$$

The definitions of KL-divergence d_{KL} and quadradic distance d_Q given in Introduction in terms of log loss and quadratic loss respectively are equivalent to the above. It can be shown that $(1/2) \cdot (d^D_Q(c,h)) \leq d^D_{KL}(c,h)$ (See Appendix B for the proof) but d^D_{KL} is not bounded from above by any real-valued function of d^D_Q. For example, define a p-concept h, and for each $\alpha > 0$ a p-concept c_α as follows:

$$\forall x \in X\ h(1|x) = 0,\ h(0|x) = 1 \ \text{ and}$$
$$\forall x \in X\ c_\alpha(1|x) = \alpha,\ c_\alpha(0|x) = 1-\alpha.$$

Then, if we let U be the uniform distribution over X, then we have for all $\alpha > 0$, $d^U_{KL}(c_\alpha, h) = \infty$ whereas $d^U_Q(c_\alpha, h) = \alpha^2$.

Recall the definitions of log-loss L_h and quadratic-loss Q_h of a stochastic rule h given in the Introduction. The *empirical* log-loss of hypothesis h in a given sample S of size m, denoted $\widehat{E}_S(L_h)$, is defined as follows:

$$\widehat{E}_S(L_h) = \frac{1}{m} \sum_{(x,y)\in S} L_h(x,y)$$

The *empirical* quadratic-loss of hypothesis h is similarly defined. Note in general that we will let $\widehat{E}_S(f)$ denote the empirical estimation of random variable f in sample S.

As we noted in Introduction, a natural learning strategy A with respect to the KL-divergence is to find a member of the hypothesis class which approximately minimizes the *empirical log-loss* in the input sample. That is,

$$A(S) = \arg\min_{h\in H} \widehat{E}_S(L_h).$$

Provided that the empirical losses converge fast to the true average losses, such a strategy will not only be a polynomial learner but in fact a *robust* learner, since it holds for an arbitrary target stochastic rule p that

$$d^D_{KL}(p,h) - d^D_{KL}(p,c) = E_D[E_p[L_h(x,y)]] - E_D[E_p[L_c(x,y)]].$$

Similarly, the analogous learning strategy with respect to the quadratic distance will be a robust learner.

The main lemma which we use to bound the convergence rate of the empirical log-loss as well as the empirical quadratic-loss for all members of a given hypothesis class is the following fact which follows from Hoeffding's inequality. (See for example [10].)

Lemma 2.1 (Hoeffding): Let $\mathcal{F}$ be a finite class of bounded random variables on a set X, that is for each $f \in \mathcal{F}$, $f : X \to [0, M]$ for some real $M \in R$. Let D be an arbitrary distribution over X. Then we have:

$$m \geq \frac{M^2}{\epsilon^2}\left(\ln \mid \mathcal{F} \mid + \ln\frac{1}{\delta}\right) \Rightarrow$$
$$D^m\{S \in X^m : \exists f \in \mathcal{F}\ |\widehat{E}_S(f) - E_D(f)| > \epsilon\} < \delta.$$

The main difficulty in applying the above lemma to bound the convergence rate for the empirical log-loss is the unbounded range of the log loss function. To circumvent this problem, we introduce the following notion. We say that a class of stochastic rules $H^{(\epsilon)}$ is an ϵ-bounded approximation of another class H, if for each $h \in H_{s,n}$ there is $h^{(\epsilon)} \in H^{(\epsilon)}_{s,n}$ such that for some polynomial q and some real constant C,

$$\forall n \in \mathbf{N}\ \forall x \in X_n\ \forall y \in Y$$
$$L_{h^{(\epsilon)}}(x,y) = \ln\frac{1}{h^{(\epsilon)}(y|x)} \leq q\left(\frac{1}{\epsilon}, n, s, |Y|\right). \quad (2)$$
$$\forall n \in \mathbf{N}\ \forall x \in X_n\ \forall y \in Y$$
$$L_{h^{(\epsilon)}}(x,y) - L_h(x,y) = \ln\frac{h(y|x)}{h^{(\epsilon)}(y|x)} \leq C\epsilon. \quad (3)$$

The following notion of 'ϵ-Bayesian averaging' offers us

†In information theory the KL-divergence is defined with the binary logarithm, but here we define it using the natural logarithm in order to simplify the subsequent calculations. Note that the relationships between the KL-divergence and various distance measures quoted in Introduction and below are with respect to the KL-divergence with the natural logarithm.

a way of obtaining an ϵ-bounded approximation of an *arbitrary* class of stochastic rules. Define for an arbitrary stochastic rule h its ϵ-*Bayesian shift*, written $h^{(\epsilon)}$, as follows.

$$\forall x \in X\ \forall y \in Y\ h^{(\epsilon)}(y|x) = (1-\epsilon)h(y|x) + \frac{\epsilon}{|Y|} \quad (4)$$

Then, it is easy to check that the ϵ-Bayesian shifts satisfy the following properties.

$$\forall x \in X\ \forall y \in Y\ h^{(\epsilon)}(y|x) \geq \frac{\epsilon}{|Y|} \text{ and} \quad (5)$$

$$\forall x \in X\ \forall y \in Y$$

$$\begin{aligned} L_{h^{(\epsilon)}}(x,y) - L_h(x,y) &= \ln \frac{h(y|x)}{h^{(\epsilon)}(y|x)} \\ &= \ln \frac{h(y/x)}{(1-\epsilon)h(y|x) + \epsilon/|Y|} \\ &\leq \ln \frac{1}{1-\epsilon} = 2\epsilon \text{ (when } \epsilon \leq 1/2.) \end{aligned} \quad (6)$$

When one insists that the ϵ-bounded approximation class be contained in the original class, however, the ϵ-Bayesian averaging in its general form does not always work. In deriving sample complexity upper bounds for specific classes of stochastic rule representations in Sect. 4, we will tailor the general idea of 'ϵ-bounded approximation' to each class of interest. For an arbitrary member h of the original class $h^{(\epsilon)}$ will in each case denote a member of the original class for which (2) and (3) holds.

3. Equivalence of Models for Polynomial Learnability of Stochastic Rules

In this section, we prove the equivalence of learning models for the two distance measures d_{KL} and d_Q. This equivalence is shown to hold given that the target stochastic rule belongs to the class of interest, and that the learning algorithm is allowed to use arbitrary polynomially evaluable hypotheses, namely in the non-proper, non-robust learning model.

Theorem 3.1: For any class C of stochastic rule representations, C is learnable with respect to d_{KL} in the non-proper, non-robust learning model, if and only if C is learnable with respect to d_Q.

Proof: It suffices to show that learnability with respect to the quadratic distance, d_Q, implies learnability with respect to the KL divergence, d_{KL}. (The opposite direction is implied by the fact that $d_{KL}(c,h) \geq d_Q(c,h)/2$ (see Appendix B.1)) Suppose a class C is polynomially learnable in terms of H with respect to d_Q. Then for given $\epsilon \leq 1$ and δ, obtain a hypothesis $h \in H$ such that with probability at least $1-\delta$, $d_Q^D(c,h) \leq (\epsilon/4|Y|)^6$, where $c \in C$ is the target stochastic rule and D is the distribution over X according to which the examples are drawn. Now simply output the $(\epsilon/8)$-Bayesian shift $h^{(\epsilon/8)}$ of h. We will show that $d_{KL}^D(c, h^{(\epsilon/8)}) < \epsilon$.

Since $d_Q(h,c) \leq (\epsilon/4|Y|)^6$, by Markov's inequality (as done in Theorem 4, part (iii) of [6]):

$$\sum_{x\,:\,\exists y \in Y\ |h(y|x) - c(y|x)| \geq (\frac{\epsilon}{4|Y|})^2} D(x) \leq \left(\frac{\epsilon}{4|Y|}\right)^2$$

Since $|h^{(\epsilon/8)}(y|x) - c^{(\epsilon/8)}(y|x)| = (1-\epsilon/8)\cdot|h(y|x) - c(y|x)|$ for arbitrary $x \in X$ and $y \in Y$,

$$\sum_{x \in X'} D(x) \leq \left(\frac{\epsilon}{4|Y|}\right)^2, \quad (7)$$

where X' is the following subset of X:

$$X' = \left\{ x \in X\ :\ \exists y \in Y\ |h^{(\epsilon/8)}(y|x) - c^{(\epsilon/8)}(y|x)| \geq \left(\frac{\epsilon}{4|Y|}\right)^2 \right\}. \quad (8)$$

From (5), we have:

$$\forall x \in X\ \forall y \in Y\ h^{(\epsilon/8)}(y|x) \geq \frac{\epsilon}{8|Y|} \quad (9)$$

Using (7) and (9), we can derive the following.

$$\begin{aligned} & d_{KL}^D(c, h^{(\epsilon/8)}) - d_{KL}^D(c, c^{(\epsilon/8)}) \\ &= \sum_{x \in X} D(x) \sum_{y \in Y} c(y|x) \ln \frac{c^{(\epsilon/8)}(y|x)}{h^{(\epsilon/8)}(y|x)} \\ &= \sum_{x \in X'} D(x) \sum_{y \in Y} c(y|x) \ln \frac{c^{(\epsilon/8)}(y|x)}{h^{(\epsilon/8)}(y|x)} \\ &\quad + \sum_{x \in X - X'} D(x) \sum_{y \in Y} c(y|x) \ln \frac{c^{(\epsilon/8)}(y|x)}{h^{(\epsilon/8)}(y|x)} \\ &\leq (\frac{\epsilon}{4|Y|})^2 \ln \frac{8|Y|}{\epsilon} \qquad \text{(from (7) and (9))} \\ &\quad + \sum_{x \in X - X'} D(x) \sum_{y \in Y} c(y|x) \ln \frac{\frac{\epsilon}{8|Y|} + (\frac{\epsilon}{4|Y|})^2}{\frac{\epsilon}{8|Y|}} \\ &\leq \left(\frac{\epsilon}{4|Y|}\right)^2 \ln \frac{8|Y|}{\epsilon} + \ln\left(1 + \frac{\epsilon}{2|Y|}\right) \end{aligned} \quad (10)$$

Now we can bound from above the two terms separately as follows:

$$\left(\frac{\epsilon}{4|Y|}\right)^2 \ln \frac{8|Y|}{\epsilon} \leq \frac{\epsilon^2}{64} \ln \frac{16}{\epsilon} \leq \frac{\epsilon}{8}$$

$$\text{since } \ln \frac{16}{\epsilon} \leq \frac{8}{\epsilon} \text{for } \epsilon \leq 1. \quad (11)$$

$$\ln\left(1 + \frac{\epsilon}{2|Y|}\right) \leq \frac{\epsilon}{4} \quad (12)$$

From (10), (11), and (12), we have

$$d_{KL}^D(c, h^{(\epsilon|Y|/8)}) - d_{KL}^D(c, c^{(\epsilon|Y|/8)}) \leq \frac{3\epsilon}{8} \quad (13)$$

From (13) and since $\epsilon \leq 1$, we finally have:

$$\begin{aligned} & d_{KL}^D(c, h^{(\epsilon/8)}) \quad (14)\\ & = (d_{KL}^D(c, h^{(\epsilon/8)}) - d_{KL}^D(c, c^{(\epsilon/8)})) \\ & \quad + d_{KL}^D(c, c^{(\epsilon/8)}) \\ & \leq \frac{3}{8}\epsilon + \max_{x\in X, y\in Y} \ln \frac{c(y|x)}{c^{(\epsilon/8)}(y|x)} \\ & \leq \frac{3}{8}\epsilon + \ln \frac{1}{1-\frac{\epsilon}{8}} \\ & \leq \frac{3}{8}\epsilon + \frac{\epsilon}{4} \quad (\text{since } \epsilon \leq 1) \\ & < \epsilon. \quad (15) \end{aligned}$$

□

4. Sample Complexity Bounds

As discussed in Introduction, minimizing $d_{KL}^D(c, h)$ and $d_Q^D(c, h)$ is equivalent to minimizing the expected log loss $E_{D\cdot c}(L_h)$ and quadratic loss $E_{D\cdot c}(Q_h)$, respectively. A natural strategy for a learning algorithm is to minimize the empirical loss in a large enough sample and hope that the empirical estimates of the loss functions for the target class in question converge uniformly to their true expectations for a moderate sample size. The sample sizes required for the standard uniform convergence results [4], [12], [17], however, grow polynomially with the size of the range of the random variables. In our case, $L_h = \ln(1/h(y|x))$ is *unbounded*. To address this problem, we make use of the notion of ϵ-bounded approximation: For each hypothesis class $H_{n,s}$ for robust learning and $\epsilon > 0$, we try to find a subclass $H_{n,s}^{(\epsilon)}$ of $H_{n,s}$, which is an ϵ-bounded approximation of $H_{n,s}$. By (3), in order to approximately minimize $E_{D\cdot c}(L_h)$ over $h \in H_{n,s}$, it suffices to minimize this expectation over $h \in H_{n,s}^{(\epsilon)}$. By tailoring the general ϵ-Bayesian averaging idea for the specific class of stochastic rule representations in question, we can show the existence of such $H_{n,s}^{(\epsilon)}$ for many important classes of stochastic rule representations. By taking advantage of the fact that the range of the log loss functions for $H_{n,s}^{(\epsilon)}$ is bounded, we can exhibit a finite class G of stochastic rules of moderate cardinality which finely covers the entire class $H_{n,s}^{(\epsilon)}$. For this finite class, a standard uniform convergence technique can be applied. We show below how this is achieved for probabilistic decision lists, probabilistic acceptors, and (convex) linear combinations[†].

Probabilistic decision lists are stochastic rules over the domain $\{0,1\}^n$. Such a hypothesis h of size s is represented by a list

$$(l_1, r_1(y)), \ldots, (l_s, r_s(y))$$

where the l_i are distinct Boolean literals and the $r_i(y)$ are vectors[††] in $[0,1]^{|Y|}$ such that $\sum_{y\in Y} r_i(y) = 1$. For an assignment x, $h(y|x)$ is defined to be $r_j(y)$, where j is the least index such that l_j is made true by x. Let $DL_{n,s}$ be the class of all such lists of size at most s. The subclass $DL_{n,s}^{(\epsilon)}$ consists of the same hypotheses except that all $r_j(y)$ are at least $\epsilon/|Y|$, and the ϵ-Bayesian-averaged $h^{(\epsilon)}$ is the same as h but the $r_j(y)$ are replaced by $(1-\epsilon)r_j(y) + \epsilon/|Y|$. Note that the range of L_h for any $h \in DL_{n,s}^{(\epsilon)}$ is contained in $[0, \ln(|Y|/\epsilon)]$.

Probabilistic acceptors are nondeterministic finite state automata except that each transition is labeled with a probability so that for all letters a the total probability of all a-transitions leaving each state is one[†††]. Note that for some letters a and pairs of states (t_1, t_2) there might not be any transition from t_1 to t_2 labeled with a, i.e. the corresponding transition probability is zero.

A path of transitions is assigned the product of the probabilities along the path. A word is accepted by the total probability of all accepting paths labeled with that word. We let $h(1|w)$ and $h(0|w)$ respectively denote the acceptance and rejection probabilities of word w by probabilistic acceptor h. We let $h(1|w)$ denote the acceptance probability by probabilistic acceptor h on word w. For some fixed alphabet Σ, let $PA_{n,s}$ be all p-concepts over the domain $\Sigma^{\leq n}$ represented by probabilistic automata with at most s states. Let $PA_{n,s}^{(\epsilon)}$ be the same class except that all transition probabilities are at least $\epsilon/2ns$. For a hypothesis h with $s' \leq s$ states, its ϵ-Bayesian shift $h^{(\epsilon)}$ is represented by the same automaton as h except that each transition probability r in h is replaced by

$$\left(1 - \frac{\epsilon}{2n}\right) r + \frac{\epsilon}{2ns'}.$$

Note that in every $h^{(\epsilon)}$, for each letter and each state, the sum total of the probabilities of transitions going into that state labeled with that letter is at least $\epsilon/2n$ (provided $\epsilon \leq 2n$). Thus, the range of L_h for any $h \in PA_{n,s}^{(\epsilon)}$ is contained in $[0, n\ln(2n/\epsilon)]$, since the smallest probability assignable on a word of length n is at least $(\epsilon/2n)^n$. Hence (2) holds for this definition of $h^{(\epsilon)}$. Also the ratio of any transition probability in h over the same transition probability in $h^{(\epsilon)}$ is at most

$$\frac{1}{1-\epsilon/2n} = 1 + \frac{\epsilon/2n}{(1-\epsilon/2n)} \leq 1 + \frac{\epsilon}{n}, \quad \text{since } \epsilon/n \leq 1.$$

It is easy to see therefore that for all $y \in Y$ and $w \in$

[†]Strictly speaking, the ϵ-bounded approximation $H_{n,s}^{(\epsilon)}$ we obtain for (convex) linear combinations is not a subclass of $H_{n,s}$.

[††]Specifying $r_i(y)$ for all $y \in Y$ is redundant but simplifies the subsequent discussion.

[†††]For simplicity we assume that there is one start state and no initial state distribution.

$\Sigma^{\leq n}$,

$$\ln \frac{h(y|w)}{h^{(\epsilon)}(y|w)} \leq \ln \left(1+\frac{\epsilon}{n}\right)^n \leq \epsilon \left(\ln \left(1+\frac{\epsilon}{n}\right)^{n/\epsilon}\right) \leq \epsilon.$$

Thus (3) holds for this definition of $h^{(\epsilon)}$.

Convex linear combinations of a finite number, say s, of fixed stochastic rules, $p_1, \ldots, p_s$, are the stochastic rules expressible in the form $\sum t_i \cdot p_i$ where the t_i satisfy $t_i \geq 0$ and $\sum_{i+1}^{s} t_i = 1$. Let $CL(p_1, \ldots, p_s)$ denote the set of all convex linear combinations of $p_1, \ldots, p_s$. It is easily verified that the ϵ-Bayesian shift of any member of $CL(p_1, \ldots, p_s)$ belongs to $CL(p_1, \ldots, p_s, u)$, where u is the 'uniform stochastic rule,' namely the unique stochastic rule satisfying $u(y|x) = 1/|Y|$ for all $x \in X$ and $y \in Y$: Given any $p \in CL(p_1, \ldots, p_s)$, say $p = \sum t_i \cdot p_i$, we can obtain its ϵ-Bayesian shift $p^{(\epsilon)}$ by setting $p^{(\epsilon)} = (\sum_{i=1}^{s}(1-\epsilon)t_i \cdot p_i) + \epsilon \cdot u$, which is indeed a member of $CL(p_1, \ldots, p_s, u)$. Clearly, $p^{(\epsilon)}$ so defined satisfies (2) and (3), by the properties (5) and (6) of ϵ-Bayesian shifts. For brevity, we will let $CL^{(\epsilon)}(p_1, \ldots, p_s)$ denote the class of ϵ-Bayesian shifts obtained in this manner from members of $CL(p_1, \ldots, p_s)$.

For the log loss functions for these bounded classes $DL_{n,s}^{(\epsilon)}$, $PA_{n,s}^{(\epsilon)}$, and $CL^{(\epsilon)}(p_1, \ldots, p_s)$, we can obtain upper bounds on the sample complexity required for uniform convergence. In the case of quadratic loss for any stochastic rule h the random variable $Q_h = (1 - h(y|x))^2$ has range $[0, 1]$. Thus we can apply uniform convergence methods directly to the classes $DL_{n,s}$, $PA_{n,s}$, and $CL(p_1, \ldots, p_s)$, instead of to the corresponding bounded classes.

Below we give proofs of sample complexity bounds based on the idea of 'direct approximation' by Pollard [10], rather than the 'dimension based' approaches [4], [12]. The method can be readily adapted to hypothesis classes other than DL, PA and CL.

Theorem 4.1: There is an algorithm that properly robustly learns the class of probabilistic decision lists with respect to d_{KL} requiring sample complexity

$$O\left(\frac{\log^2(|Y|/\epsilon)}{\epsilon^2}\left(s \log n + s|Y| \log\left(\frac{1}{\epsilon} \log \frac{|Y|}{\epsilon}\right) + \log \frac{1}{\delta}\right)\right),$$

and

$$O\left(\frac{1}{\epsilon^2}\left(s \log n + s|Y| \log \frac{1}{\epsilon} + \log \frac{1}{\delta}\right)\right)$$

with respect to d_Q.

Proof: For the log loss we construct a small class G of stochastic rules by rounding the probability parameters in the representations for hypotheses in $DL_{n,s}^{(\epsilon)}$: From an arbitrary stochastic rule h in $DL_{n,s}^{(\epsilon)}$, we obtain the corresponding member g of G by first rounding down each parameter $(r_j(y))$ to the next power of $(1 - \epsilon/2)$ and then normalizing so that $\sum_{y \in Y} r_j(y) = 1$. By the way they are constructed, for each $h \in DL_{n,s}^{(\epsilon)}$ the corresponding $g \in G$ has the following property:

$$\forall x \in \{0,1\}^n \; y \in \{0,1\} : \quad L_g(x,y) - L_h(x,y) = \ln \frac{h(y|x)}{g(y|x)} \leq \ln \frac{1}{1-\epsilon/2} \leq \epsilon. \tag{16}$$

Since normalization can increase the value of any probability parameter by at most a factor of $1/(1-\epsilon/2)$, we also have

$$\forall x \in \{0,1\}^n \; y \in \{0,1\} : \quad L_h(x,y) - L_g(x,y) = \ln \frac{g(y|x)}{h(y|x)} \leq \ln \frac{1}{1-\epsilon/2} \leq \epsilon. \tag{17}$$

From (16) and (17) it follows that the log-loss for g and h differ by at most ϵ on *any* data point (x, y), namely,

$$|E_D(L_g) - E_D(L_h)| \leq \epsilon,$$

and that for an arbitrary sample S,

$$|\widehat{E}_S(L_g) - \widehat{E}_S(L_h)| \leq \epsilon.$$

Hence it follows that the uniform convergence for the class of random variables $\mathcal{G} = \{L_g : g \in G\}$ implies the same for $\mathcal{H} = \{L_h : h \in H\}$. More precisely, suppose that we have

$$\forall g \in G \; |\widehat{E}_S(L_g) - E_D(L_g)| \leq \epsilon,$$

then by summing the above three inequalities, we will also have

$$\forall h \in DL_{n,s}^{(\epsilon)} \; |\widehat{E}_S(L_h) - E_D(L_h)| \leq 3\epsilon.$$

Now, a sample complexity bound for uniform convergence for the finite class $\mathcal{G}$ can be shown using Hoeffding's inequality (See Sect. 2): Sample size $(M^2/\epsilon^2) \cdot (\ln |\mathcal{G}| + \ln(1/\delta))$ suffices for the uniform convergence of any finite class $\mathcal{G}$ of random variables with range $[0, M]$.

In this case $M = \ln(2|Y|/\epsilon)$ since the smallest conditional probability for any hypothesis in G exceeds

$$\frac{\epsilon}{|Y|}\left(1 - \frac{\epsilon}{2}\right) \geq \frac{\epsilon}{2|Y|}.$$

To get a bound on $|\mathcal{G}| = |G|$, first observe that there are at most $(2n)^s$ ways of setting the literals of a decision list of size s. For a particular setting of the s literals there are $|Y|s$ probability parameters consisting of powers of $1 - \epsilon/2$ larger than $\epsilon/2|Y|$. For each parameter, at most $(2/\epsilon) \cdot \ln(2|Y|/\epsilon)$ powers need to be considered and thus $|G| \leq (2n)^s((2/\epsilon)\ln(2|Y|/\epsilon))^{|Y|s}$, leading to the following sample complexity bound for robust learning with $DL_{n,s}$:

$$O\left(\frac{\log^2(|Y|/\epsilon)}{\epsilon^2}\left(s \log n + s|Y| \log\left(\frac{1}{\epsilon} \log \frac{|Y|}{\epsilon}\right) + \log \frac{1}{\delta}\right)\right)$$

with respect to d_{KL}. (The foregoing argument is actually for obtaining accuracy 3ϵ, but this does not matter since for the moment we are only concerned with the order of the sample complexity.)

We can also apply scaling to the whole class $DL_{n,s}$ and construct a finite class G such that uniform convergence of the quadratic loss for hypotheses in G assures uniform convergence of the quadratic loss for hypotheses in $DL_{n,s}$: G consists of all hypotheses which can be obtained from some member h of $DL_{n,s}$ in the following manner. Given h, we first round up each parameter of h to the next non-zero *multiple* of $\sqrt{\epsilon}/2$, and then re-normalize them by deviding by a constant factor so that $\sum_{y\in Y} r(y)_j = 1$ for each j (this factor is at least 1). Thus each parameter in h changes by at most $\sqrt{\epsilon}/2$. By this procedure we obtain a hypothesis $g \in G$ whose quadratic loss differs from the loss of the original h (using Eq. (1)) by at most $2(\sqrt{\epsilon}/2)^2 < \epsilon$. Similarly to the case of log loss, triangular inequalities can be applied to show that uniform convergence for the class of random variables $\{Q_g : g \in G\}$ will imply the same for $\{Q_h : h \in DL_{n,s}\}$. Finally we apply the bound $\frac{M^2}{\epsilon^2}(\log|G| + \log\frac{1}{\delta})$ for uniform convergence derived from Hoeffding's inequality with $M = 1$ and $|G| = (2n)^s(2/\sqrt{\epsilon})^{s|Y|}$ to obtain the sample complexity bound of $O((1/\epsilon^2)\cdot(s\log n + s|Y|\log(1/\epsilon) + \log(1/\delta)))$ with respect to d_Q. □

Theorem 4.2: There is an algorithm that robustly and properly learns the class of probabilistic acceptors requiring sample complexity

$$O\left(\frac{n^2\log^2(n/\epsilon)}{\epsilon^2}\left(s^2|\Sigma|\log\left(\frac{n}{\epsilon}\log\frac{ns}{\epsilon}\right)+\log\frac{1}{\delta}\right)\right)$$

with respect to d_{KL}, and

$$O\left(\frac{1}{\epsilon^2}\left(s^2|\Sigma|\log\left(\frac{\epsilon}{n}\log\frac{ns}{\epsilon}\right)+\log\frac{1}{\delta}\right)\right)$$

with respect to d_Q.

Proof: For the log loss we again construct a small class G of hypotheses by rounding the probability parameters in the representations for hypotheses in the ϵ-Bayesian averaged class $PA^{(\epsilon)}_{n,s}$: From an arbitrary hypothesis h in $PA^{(\epsilon)}_{n,s}$, we obtain the corresponding member g of G by first rounding off all parameters of h to the next power of $(1-\epsilon/2n)$ and then normalizing so that for each state and letter the outgoing probabilities sum to one. (Note that all parameters in the p-concepts in G are larger than $(\epsilon/2ns)(1-(\epsilon/2n)) \geq \epsilon/4ns$.) Again (16) and (17) can be shown to hold (except this time for all $x \in \Sigma^{\leq n}$), since

$$\ln\frac{h(y|x)}{g(y|x)} \leq n\ln\frac{1}{1-\epsilon/2n} \leq n\ln\left(1+\frac{\epsilon}{n}\right) \leq \epsilon, \quad (18)$$

and

$$\ln\frac{g(y|x)}{h(y|x)} \leq \epsilon. \quad (19)$$

We can bound the cardinality of G from above by $((2n/\epsilon)\log(4ns/\epsilon))^{s^2|\Sigma|}$. Also the probability assigned by any member of G on any string in $\Sigma^{\leq n}$ is at least $(\epsilon/2n)^n$. So we can apply the bound $(M^2/\epsilon^2)\cdot(\log|G| + \log(1/\delta))$ with $M = n\ln(2n/\epsilon)$ and $|G| = ((2n/\epsilon)\cdot\log(4ns/\epsilon))^{s^2|\Sigma|}$, leading to a sample complexity bound of order $O((n^2\log^2(n/\epsilon)/\epsilon^2)\cdot(s^2|\Sigma|\log((n/\epsilon)\log(ns/\epsilon)) + \log(1/\delta)))$. A similar bound can be proven for $PA_{n,s}$ and the quadratic loss using the same scaling method: We can show that the following holds for any x, y, using the same definition for g,

$$\begin{aligned}|h(y|x) - g(y|x)| &\leq \left(1-\left(1-\frac{\epsilon}{2n}\right)^n\right)\\ &\leq \left(1-\left(1-\frac{\epsilon}{2}\right)\right) \leq \frac{\epsilon}{2}.\end{aligned}$$

Now using the Hoeffding's inequality with $M = 1$ and $|G| = ((n/\epsilon)\cdot\log(ns/\epsilon))^{s^2|\Sigma|}$ we obtain the bound

$$O\left(\frac{1}{\epsilon^2}\left(s^2|\Sigma|\log\left(\frac{n}{\epsilon}\log\left(\frac{ns}{\epsilon}\right)\right)+\log\frac{1}{\delta}\right)\right).$$

□

Theorem 4.3: There is an algorithm that robustly learns the class of convex linear combinations of s stochastic rules requiring sample complexity

$$O\left(\frac{\log^2(|Y|/\epsilon)}{\epsilon^2}\left(s\log\left(\frac{1}{\epsilon}\log\frac{1}{\epsilon}\right)+\log\frac{1}{\delta}\right)\right)$$

with respect to d_{KL}, and

$$O\left(\frac{1}{\epsilon^2}\left(s\log\frac{1}{\epsilon}+\log\frac{1}{\delta}\right)\right)$$

with respect to d_Q.

Proof: As we explained in the introductory part of this section, we use $CL^{(\epsilon)}(p_1,\ldots,p_s)$ as hypotheses, so this is strictly speaking a non-proper learning algorithm for $CL(p_1,\ldots,p_s)$.

Similarly to the other cases, we can obtain a finite subclass G of $CL^{(\epsilon)}(p_1,\ldots,p_s)$ whose uniform convergence of the associated log loss functions implies the same for those of the entire class: Given any member of $CL^{(\epsilon)}(p_1,\ldots,p_s)$, we round off each parameter t_i to the next power of $(1-\frac{\epsilon}{2})$ and then re-normalize them so that they sum to one. (Note that all parameters of the resulting p-concept are at least $\epsilon/2|Y|$.)

The cardinality of G is clearly bounded above by $((2/\epsilon)\log(2|Y|/\epsilon))^s$, and for this definition of G, we can show again that (16) holds, except this time for all x in the domain X. Once again, we apply the bound due tue Hoeffding's inequality with $M = \log(2|Y|/\epsilon)$ and $|G| = ((2/\epsilon)\log(2|Y|/\epsilon))^s$, and obtain a sample complexity bound of order $O((\log^2(|Y|/\epsilon)/\epsilon^2)\cdot(s\log((1/\epsilon)\cdot$

$\log(|Y|/\epsilon)) + \log(1/\delta)))$.

For the quadratic loss, we can again round up all the parameters to the next non-zero multiple of $\sqrt{\epsilon}/2$ and then renormalize. Now the cardinality of G is at most $(2/\sqrt{\epsilon})^s$ and Hoeffdings inequality with $M = 1$ leads to sample size bound†, of $O((1/\epsilon^2) \cdot (s \log(1/\epsilon) + \log(1/\delta)))$. □

5. Efficient Learning of Convex Combinations of Probabilistic Concepts

The learning algorithms that were implicit in the proofs of the last section all have polynomial sample complexity but their running time is exponential. In this section we give efficient algorithms for learning one of the classes of stochastic rules considered: convex combinations of stochastic rules. It is rather straightforward to do this for the quadratic loss (as we will see in the proof of Theorem 5.1). The proof for learning with respect to the Kullback-Leibler divergence, in contrast, is rather involved (Theorem 5.2).

Given a finite set of stochastic rules, $q_i(y|x)$ $(i = 1, \ldots, s)$, let $CL(q_1, \ldots, q_s)$ denote the class of *convex* linear combinations of these stochastic rules, i.e.

$$CL(q_1, \ldots, q_s) = \left\{ \sum_{i=1}^{s} t_i q_i(y|x) : t_i \geq 0, \sum_{i=1}^{s} t_i = 1 \right\}. \tag{20}$$

The condition $\sum_{i=1}^{s} t_i = 1$ in the above definition is the *convexity* condition. We assume that each member x of the domain X is of length at most n, and that each p-concept q_i is polynomially evaluable.

Theorem 5.1: There exists an algorithm which polynomially and robustly learns the class of convex linear combinations of s arbitrary, polynomially evaluable stochastic rules (evaluable in $q(n)$ time) with respect to d_Q. The algorithm requires sample complexity $O((1/\epsilon^2) \cdot (s \log(1/\epsilon) + \log(1/\delta)))$, and running time $O((s^4 + ms) \cdot q(n))$.

Proof of Theorem 5.1: Let $F(\vec{u})$ denote the empirical quadratic loss of the input sample, which is to be minimized, i.e.,

$$F(\vec{u}) = \sum_{t=1}^{m} Q_{\sum_{i=1}^{s} u_i q_i}(y_t|x_t)/m.$$

Let Z denote the set of $\vec{t} \in \mathbf{R}^s$ satisfying the convexity condition, and V the subset of Z satisfying the non-negativity conditions, i.e. $t_i \geq 0, i = 1, \ldots, s$. We will show that we can approximately solve the minimization problem of $F(\vec{u})$ with the restriction that $\vec{u} \in V$ in $O(ms^3)$ time. It is a quadratic programming problem with linear and convex constraints: $\forall i$, $u_i \geq 0$ and $\sum_i u_i = 1$. First, we modify $F(\vec{u})$ so that it necessarily attains minimum at a unique point by introducing an additional term, and obtain $F_\epsilon(\vec{u})$.

$$F_\epsilon(\vec{u}) = \sum_{t=1}^{m} Q_{\sum_{i=1}^{s} u_i q_i}(y_t|x_t)/m + \frac{\epsilon}{2} \sum_{i=1}^{s} u_i^2$$

Note that a point that minimizes $F_\epsilon(\vec{u})$ will approximately minimize $F(\vec{u})$ within ϵ, since $0 \leq \sum_{i=1}^{s} u_i^2 \leq 1$ holds for any $u \in V$. Now let $\{\vec{e}_i\}$ $(i = 1, \ldots, s)$ be the standard unit vectors of $\mathbf{R}^s$. For each $\vec{e}_i$ define a new vector by $\vec{f}_i = \vec{e}_i - \vec{g}$, where $\vec{g} = \sum_{i=1}^{s} \vec{e}_i/s$. Then $\{\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_{s-1}\}$ forms a basis of the linear space $\{\vec{x} | \sum_{i=1}^{s} x_i = 0\}$. Let $\vec{h}(\vec{x}) = \vec{g} + \sum_{i=1}^{s-1} x_i \vec{f}_i$, then $\sum_{i=1}^{s} h_i(\vec{x}) = 1$ holds for any $\vec{x} \in \mathbf{R}^{s-1}$. Conversely, any $\vec{v}$ with $\sum_{i=1}^{s} v_i = 1$ can be represented as $\vec{v} = \vec{h}(\vec{x})$ for some $\vec{x}$. Hence we can think of the minimization problem for $F_\epsilon(\vec{u})$ with respect to $(s-1)$-dimensional vector $\vec{x}$ as the variables. Define $T(\vec{x}) = F_\epsilon(\vec{h}(\vec{x}))$, then our problem is to minimize $T(\vec{x})$ with the restriction that $h_i(\vec{x}) \geq 0$. Since T is quadratic in $\vec{x}$ and positive (non-negative) definite (recall the definition of quadratic loss), we can write $T(\vec{x})$ as $T(\vec{x}) = \vec{x} \cdot H \cdot \vec{x}/2 + \vec{a} \cdot \vec{x} + C'$, where H is a positive definite constant symmetric matrix and $\vec{a}$ and C' are constants (which depend on the sample.) We then diagonalize H as follows. Let λ_i $(i = 1, \ldots, s-1)$ denote the eigen values of H, and let $\vec{\mu}_i$ be the unit eigen vector associated with λ_i, where we insist that $\vec{\mu}_i \cdot \vec{\mu}_j = 0$ $(i \neq j)$, even for $\lambda_i = \lambda_j$ (multiple eigenvalue). Note that, by the modification that gave F_ϵ from F, $\lambda_i > 0$ holds for all i. Now introduce new coordinates y_i's by setting $\vec{x} = \sum_{i=1}^{s-1} y_i \vec{\mu}_i$, and define $U(\vec{y}) = T(\sum_{i=1}^{s-1} y_i \vec{\mu}_i)$. Then, we have $U(\vec{y}) = \sum_{i=1}^{s} \lambda_i y_i^2/2 + \sum_{i=1}^{s-1} y_i \vec{a} \cdot \vec{\mu}_i + C'$, and by quadratic completion, $U(\vec{y}) = \sum_{i=1}^{s} \lambda_i (y_i - y_i^*)^2/2 + C$ for some constants y_i^* and C. Finally, put $z_i = y_i/\sqrt{\lambda_i}$ $(z_i^* = y_i^*/\sqrt{\lambda_i})$ so that

$$U'(\vec{z}) = U(\vec{y}) = \sum_{i=1}^{s} (z_i - z_i^*)^2/2 + C. \tag{21}$$

As $\vec{u}$ varies over V, $\vec{z}$ varies over $\bar{V}$, where $\bar{V}$ is the intersection of $s-1$ halfspaces corresponding to the non-negativity conditions for $h_i(\vec{x})$, say $\bar{V}_i = \{\vec{z} | \vec{b}_i \cdot \vec{z} \geq c_i\}, i = 1, \ldots, s-1$. We also let B_i denote the hyperplane associated with $\bar{V}_i$, i.e. $B_i = \{\vec{z} | \vec{b}_i \cdot \vec{z} = c_i\}$.

Now if $\vec{z}^* = (z_1^*, \ldots, z_s^*)^T \in \bar{V}$ holds, then $F_\epsilon(\vec{u}) = U'(\vec{z})$ attains minimum when $\vec{z} = \vec{z}^*$. If, on the other hand, $\vec{z}^* \notin \bar{V}$, assume that $\vec{z}^* \notin \bar{V}_i$ $(i \in I)$ and $\vec{z}^* \in \bar{V}_i$ $(i \notin I)$. Then, we see from (21) that the minimum of $U(\vec{z})$ is attained at the point in $\bar{V}$ that is closest to $\vec{z}^*$ with respect to the Euclidean distance. In order to search for that point, the following algorithm, FindNearest, given in Fig. 1 can be employed.

When $\vec{z}^* \in \bar{V}$, FindNearest outputs $\vec{z}^*$, otherwise

†Using the dimension-based approach, a comparable sample complexity bound has also been obtained for learning this class with respect to the quadratic loss [6].

Algorithm 'FindNearest'

Input:
$(z_1^*, \ldots, z_s^*)$ – optimal point ignoring constraints
$(\vec{b}_1, c_1), \ldots, (\vec{b}_{s-1}, c_{s-1})$ – boundaries
1. /* Initialization */
$\vec{\xi} := \vec{z}^*$, $I := \emptyset$
2. Check which boundaries are violated by $\vec{\xi}$, i.e.
$I' := \{i : \vec{b}_i \cdot \vec{\xi} < c_i,\ i \notin I\}$
3 If $I' = \emptyset$, then output ξ and halt.
Otherwise let $I := I \cup I'$ and continue.
4. Update $\vec{\xi}$ by projecting it onto the intersection of the violated boundaries, i.e. $\bigcap_{i \in I} B_i$
Goto 2.

Fig. 1 The algorithm 'FindNearest.'

it projects $\vec{z}^*$ onto $\bigcap_{i \in I} B_i$ (the intersection of those boundaries violated by $\vec{z}^*$). By the Pythagorean theorem, $\bar{U}(\vec{z})$ attains minimum at the point in $\bigcap_{i \in I} B_i \cap \bar{V}$ which is closest to the projected point ξ. When FindNearest goes back to step 2, the situation is the same as before, except the dimension of the search space is reduced. FindNearest runs until ξ belongs to $\bar{V}$ or the dimension of the search space is 0. Since the computation time for projection is $O(s^3)$, and the number of iterations is at most s, the computation time of FindNearest is $O(s^4)$.

Since the construction of the function $U(\vec{y})$ is computable in $O(ms)$ steps, and orthogonalization of U is computable in $O(s^3)$ steps, the computational time required for the entire procedure described in this proof is $O(ms + s^4)$. The sample comlexity bound follows from Theorem 4.3 in Sect. 4. □

Theorem 5.2: There exists an algorithm which polynomially and robustly learns the class of convex linear combinations of s arbitrary, polynomially evaluable stochastic rules (in $q(n)$ time) with respect to d_{KL}. The algorithm requires sample complexity

$$O\left(\frac{\log^2(|Y|/\epsilon)}{\epsilon^2}\left(s|Y|\log\left(\frac{1}{\epsilon}\log\frac{|Y|}{\epsilon}\right)+\log\frac{1}{\delta}\right)\right),$$

and running time $O(|Y|^2 s^4 \cdot (1/\epsilon)^3 \cdot m \cdot q(n))$.

Proof of Theorem 5.2: Let Z denote the set of $\vec{t} \in \mathbf{R}^s$ satisfying the convexity condition, and V the subset of Z satisfying the non-negativity conditions, i.e. $t_i \geq 0, i = 1, \ldots, s$. Let b_i denote the boundary hyper-plane given by $t_i = 0$. We will think of V as our hypothesis space in which to perform our search for the best hypothesis. For a given accuracy parameter $\epsilon > 0$, each member $\vec{t}$ of V represents the p-concept $p(\vec{t}) = \sum_{i=1}^{s} t_i p_i$, where each p_i is the $(\epsilon/4)$-Bayesian shift of q_i, i.e. p_i is defined by $p_i(y|x) = (1 - \epsilon/4)q_i(y|x) + \epsilon/4|Y|$. We will use $\vec{p}(y|x)$ as a shorthand for $(p_1(y|x), p_2(y|x), \ldots, p_s(y|x))^T$.

Now we will present our learning algorithm, 'Descent' (See Fig. 2). The algorithm Descent is given

Algorithm 'Descent'

Input:
ϵ (accuracy parameter)
δ (confidence parameter)
$q_1, \ldots, q_s$. (s p-concept representations)
1. /* Initialization */
Draw m random examples (m is a sample complexity upper bound.) and let $S = (\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_m, y_m\rangle)$ denote the sample thus obtained.
For each $i = 1, \ldots, s$ let p_i denote the $\frac{\epsilon}{4}$-Bayesian shift of q_i, namely $p_i = q_i^{(\epsilon/4)}$.
Define initial vector $\vec{t}$ as
$\vec{t} := (\frac{1}{s}, \ldots, \frac{1}{s})^T$
$\Delta_1 := \frac{\epsilon}{8s}$ /* Δ_1 is the slack at boundaries */
$\Delta_2 := \frac{\epsilon^2}{64s|Y|^2}$ /* Δ_2 is the increment size */
2. Let $F(\vec{t})$ be the empirical log loss of S as a function of $\vec{t}$, i.e.
$F(\vec{t}) = -\frac{1}{m}\sum_{i=1}^{m} \log p(\vec{t})(y_i|x_i)$
3. /* $CurrB$ is the set of boundaries that $\vec{t}$ is Δ_1-close to */
$CurrB := \{i : t_i \leq \Delta_1\}$
4. /* Compute the gradient of F at $\vec{t}$ projected onto the convex region V */
$\nabla F(\vec{t}) = Project(-[\frac{\partial F}{\partial t_1}(\vec{t}), \frac{\partial F}{\partial t_2}(\vec{t}), \ldots, \frac{\partial F}{\partial t_s}(\vec{t})], V)$
6. $Slope := |\nabla F(\vec{t})|$
If $Slope < \frac{\epsilon}{\sqrt{2}}$
Then Output Round-off($\vec{t}$)
End If
7. /* Increment the current position */
/* Compute tentative update ignoring boundaries */
$\vec{w} := \vec{t} + \Delta_2 \cdot \nabla F(\vec{t})$
/* Take projection of $\nabla F(\vec{t})$ onto those boundaries */
/* to which (i) $\vec{t}$ is already Δ_1-close and (ii) $\vec{w}$ comes even closer */
$ProjB := \{i \in CurrB : w_i \leq t_i\}$
$\vec{h} := \Delta_2 \cdot BProject(\nabla F(\vec{t}), ProjB)$
$\vec{t} := \vec{t} + \vec{h}$
Goto 4.

Fig. 2 The algorithm 'Descent.'

access to an oracle, which upon each call returns an example $x \in X$ randomly drawn according to an unknown distribution D, and labeled probabilistically according to the target p-concept. Descent obtains a random sample $S = (\langle x_1, y_1\rangle, \ldots, \langle x_m, y_m\rangle)$ by making m calls to the oracle. It then calculates, for each $\langle x_j, y_j\rangle$ and each p_i, the conditional probability $p_i(y_j|x_j)$ in polynomial time, and obtains the matrix $[p_i(y_j|x_j)]_{i,j}$ which completely specifies the objective empirical log-loss function to be minimized, namely:

$$F(\vec{u}) = -\frac{1}{m}\sum_{j=1}^{m} \log\left(\sum_{i=1}^{s} t_i p_i(y_j|x_j)\right)$$

Descent uses as subroutine '$Project(\vec{d}, S)$' which returns the vector obtained by projecting vector $\vec{d}$ onto the sub-space S. The projection procedure is well-known in the literature and consists of a small fixed number of matrix multiplications, and hence is computable in $O(s^3)$ time. (c.f. [15] p. 116.) Descent also uses as subroutine '$BProject(\vec{d}, Indices)$' (which

stands for 'boundary project') which simply zeros out those components v_i for which $i \in Indices$. Finally, when outputting the final value of $\vec{t}$, Descent rounds off $\vec{t}$ to the nearest power of $(1-\epsilon/2)$ (which is denoted by Round-off($\vec{t}$) in the algorithm description) and renormalizes so that they sum to one.

We prove a series of lemmas concerning this algorithm, which together serve to show Theorem 5.2. First, we prove the following technical lemma, which states that the output of Decent belongs to G – the bounded, finite subclass which was defined earlier.

Lemma 5.1: Let t_i denote the i-th component of vector $\vec{t}$. Then, at any point during the execution of Decent, $t_i \geq \Delta_1/2$ holds for an arbitrary i, $1 \leq i \leq s$.

Proof of Lemma 5.1: First note that the first time t_i is incremented, $t_i \geq \Delta_1/2$ clearly holds. Now, assume inductively that $t_i \geq \Delta_1/2$ holds after the k-th increment of t_i. When $\vec{t}$ is within distance Δ_1 from the boundary b_i, we have $h_i = 0$ (where $\vec{h}$ is the amount by which $\vec{t}$ is incremented) and hence $t_i \geq \Delta_1/2$ must hold after the $k+1$-th increment as well. Next, suppose that $t_i > \Delta_1$ after the k-th increment. Now since

$$F(\vec{t}) = -\frac{1}{m}\sum_{k=1}^{m}\log\left(\sum_{i=1}^{s} t_i p_i(y_k|x_k)\right)$$

we have

$$\frac{\partial F}{\partial t_i} = -\frac{1}{m}\sum_{k=1}^{m}\frac{p_i(y_k|x_k)}{p(\vec{t})(y_k|x_k)}. \tag{22}$$

Hence, it follows that

$$|\nabla F((\vec{t}))_i| \leq \frac{4|Y|}{\epsilon}.$$

Now since $\vec{h}$ is obtained by projecting $\nabla F(\vec{t})$ onto a subspace of Z and multiplying it by Δ_2, we must have

$$|h_i| \leq \frac{4|Y|\Delta_2}{\epsilon} = \frac{\epsilon}{16|Y|s} \leq \frac{\Delta_1}{4}.$$

It then follows that

$$t_i \geq \Delta_1 - \frac{\Delta_1}{4} > \frac{\Delta_1}{2}.$$

End of Proof of Lemma 5.1

Next, we will prove that the objective function we are approximately minimizing, the empirical log-loss function, is convex and attains global minimum at either one point (when the sample is of reasonable complexity), or at one section of V. (See the statement of Lemma 5.2 for details.) Here it is important that we use $CL(p_1, p_2, \ldots, p_s)$ as the hypothesis space, and not the original class, or at least that the stochastic rules p_1, p_2, $\ldots$,p_s, are bounded away from zero. We will in fact show that the algorithm outputs a member of the hypothesis space $H = CL(p_1, p_2, \ldots, p_s)$ which approximately minimizes the empirical log-loss within accuracy $\frac{\epsilon}{2}$. Since the minimum empirical loss of any hypothesis in $CL(p_1, p_2, \ldots, p_s)$ is at most $\epsilon/4$ greater than that in $CL(q_1, q_2, \ldots, q_s)$ (given that $\epsilon \leq 4$), this implies that the algorithm's output approximately minimizes the empirical loss within $3\epsilon/4$ in $CL(q_1, q_2, \ldots, q_s)$. Whenever the sample size exceeds a certain bound (of order specified in the statement of Theorem 5.2) such a hypothesis minimizes the true expected log loss within ϵ with probability $1-\delta$.

Lemma 5.2: Let

$$S = (\langle x_1, y_1\rangle, \ldots, \langle x_m, y_m\rangle) \in (X \times Y)^m$$

be an arbitrary sample, and let $F : V \to \mathbf{R}$ denote its empirical log-loss function, namely

$$F(\vec{t}) = \widehat{E}_S(L_{p(\vec{t})}) = -\frac{1}{m}\sum_{k=1}^{m}\log\left(\sum_{i}^{s} t_i p_i(y_k|x_k)\right).$$

(a) If the rank of the set of vectors $\vec{p}(S) = \{\vec{p}(y_i|x_i) : i = 1, \ldots, m\}$ is s, then the function F is strictly convex and has at most one stationary point in V, and if one exists it is a minimal point.

(b) If the rank of $\vec{p}(S)$ is $r < s$, then the function F is convex and there exists an $(s-r)$-dimensional section W of V such that $W = \{\vec{v} \in V | F(\vec{v}) = \min_{\vec{t}\in V} F(\vec{t})\}$.

Proof of Lemma 5.2: Let H denote the Hessian matrix for F, namely the matrix having $h_{ij} = \partial^2 F/\partial t_i \partial t_j$ as the (i,j)-component, and $H_q(\vec{z})$ the quadratic form having H as its coefficient matrix, that is, $H_q(\vec{z}) = \sum_{i,j} h_{i,j} z_i z_j$. In order to show that F is convex, it suffices to show that $H_q(\vec{z})$ is positive definite, i.e. $H_q(\vec{z}) \geq 0$ and the equality holds only when $\vec{z} = \vec{0}$. From (22), we have

$$\frac{\partial^2 F}{\partial t_i \partial t_j} = \frac{1}{m}\sum_{k=1}^{m}\frac{p_i(y_k|x_k)p_j(y_k|x_k)}{(p(\vec{t})(y_k|x_k))^2}.$$

If we let $p_{ik} = p_i(y_k|x_k)$ and $p_k = p(\vec{t})(y_k|x_k)$ then, $h_{ij} = (1/m)\sum_{k=1}^{m} p_{ik}p_{jk}/p_k^2$. Hence,

$$\begin{aligned}\sum_{i,j=1}^{s} h_{ij} z_i z_j &= \frac{1}{m}\sum_{k=1}^{m}\sum_{i,j=1}^{s}\frac{z_i p_{ik} z_j p_{jk}}{p_k^2} \\ &= \frac{1}{m}\sum_{k=1}^{m}\left(\sum_{i=1}^{s}\frac{z_i p_{ik}}{p_k}\right)^2 \\ &\geq 0 \end{aligned} \tag{23}$$

The inequality of (23) is strict unless

$$\sum_{i=1}^{s}\frac{z_i p_i(y_k|x_k)}{p_k} = \frac{\vec{z}\cdot\vec{p}(y_k|x_k)}{p(\vec{t})(y_k|x_k)} = 0$$

for each $k = 1, \ldots, m$, i.e. $\vec{z}$ is orthogonal to all $\vec{p}(y_k|x_k)$, $k = 1, \ldots, m$. This is impossible if the rank of

$\vec{p}(S)$ is s, and hence in this case h_q is positive definite, and F is strictly convex. If on the other hand the rank of $\vec{p}(S)$ is $r < s$, then define a subspace U of Z by $U = \{\vec{z} \in Z | \forall k = 1, \ldots, m, (\vec{z} - \vec{g}) \cdot \vec{p}(y_k|x_k) = 0\}$ where $\vec{g}$ denotes the barycenter of V (namely $\vec{g} = (1/s, \ldots, 1/s)$), and define $U^{\perp}$ to be the orthogonal space of U with respect to Z, i.e. $U^{\perp} = \{\vec{z} \in Z | \forall \vec{u} \in U\ (\vec{z} - \vec{g}) \perp (\vec{u} - \vec{g})\}$. (Note that U is an $(s-r)$-dimensional space and $U^{\perp}$ r-dimensional.) Then the quadratic form of the Hessian matrix for F restricted to $U^{\perp}$ is clearly positive definite.

First note that since any vector $\vec{z}$ in U satisfies $(\vec{z} - \vec{g}) \cdot \vec{p}(y_k|x_k) = 0$ for all $k = 1, \ldots, m$, we see from (22) that the directional derivative of F along $(\vec{z} - \vec{g})$ is zero. Thus, if we consider the projection from V to the subspace $U^{\perp}$, written $\pi(\vec{x}, U^{\perp})$, then for any two vectors $\vec{x}$ and $\vec{y}$ in V satisfying $\pi(\vec{x}, U^{\perp}) = \pi(\vec{y}, U^{\perp})$, we must have $F(\vec{x}) = F(\vec{y})$. Thus if we define $F^* : \pi(V, U^{\perp}) \to \mathbf{R}$ by $F^*(\pi(\vec{x}, U^{\perp})) = F(\vec{x})$, then F^* is a well-defined function. Now since the Hessian matrix for F restricted to $U^{\perp}$ is positive definite, the Hessian matrix of F^* must also be positive definite, and hence F^* is strictly convex. Now suppose for contradiction that F^* has two stationary points. Since the Hessian matrix is positive definite, they are both minimal points. If we restrict the domain of F^* to the line segment connecting the two points, then by the *convexity* of $\pi(V, U^{\perp})$ (which follows easily by the convexity of V) the segment lies within $\pi(V, U^{\perp})$. F^* is therefore continuous in $\pi(V, U^{\perp})$ because a convex combination of bounded stochastic rules is necessarily bounded. Hence, F^* must take a (conditional) maximal point between the two points, contradicting the fact that its Hessian matrix is positive definite. Thus, F^* has at most one stationary point, and if one exists, say $\vec{u}_{min}$, then it must be a minimal point. Now if we let $W = \pi(\cdot, U^{\perp})^{-1}(\vec{u}_{min})$, W is the desired section of V at which F attains a constant, global minimum, proving (b). When the rank r equals s, $U^{\perp}$ is the entire Z, so W is a single point, which proves (a).

End of Proof of Lemma 5.2

Next, we bound from below the decrease in the value of F in any single update of $\vec{t}$ during an execution of 'Descent' as follows.

Lemma 5.3: Let $\vec{c}$ be the current position (current value of $\vec{t}$) and $\vec{n}$ be the next position, at an arbitrary iteration during an execution of the algorithm 'Descent.' Define $G(\vec{t})$ by

$$G(\vec{t}) = F(\vec{t}) - \left(min_{\vec{x} \in V} F(\vec{x}) + \frac{\epsilon}{4} \right)$$

Then we have:

$$G(\vec{c}) - G(\vec{n}) \geq \frac{\epsilon^2}{128 s |Y|^2} G(\vec{c})^2$$

Proof of Lemma 5.3: We prove this via a number of sublemmas.

Sublemma 5.1: Let $p_1, \ldots, p_s$ be stochastic rules which are bounded from below by $\epsilon/4|Y| > 0$, that is, for each $i \leq s$, for all $(x, y) \in X \times Y$, $p_i(y|x) \geq \epsilon/4|Y|$, and let $F(\vec{t})$ denote the empirical log-loss function in sample S of a convex linear combination $p(\vec{t})$. Then the following bound on each second partial derivative holds.

$$(\forall i, j \leq s)\ \frac{\partial F}{\partial t_i \partial t_j} \leq \left(\frac{4|Y|}{\epsilon} \right)^2$$

Proof of Sublemma 5.1: Recall that we can write $\frac{\partial^2 F}{\partial t_i \partial t_j}$ as follows.

$$\frac{\partial^2 F}{\partial t_i \partial t_j} = \frac{1}{m} \sum_{k=1}^{m} \frac{p_i(y_k|x_k) p_j(y_k|x_k)}{(p(\vec{t})(y_k|x_k))^2}$$

The numerator of each summand is clearly bounded from above by unity, and the denominator of each summand is bounded from below by $(\epsilon/4|Y|)^2$ by assumption. Hence, the above quantity is bounded from above by $(4|Y|/\epsilon)^2$.

End of Proof of Sublemma 5.1

Sublemma 5.2: Let $\vec{t}$ be the current position at any given step in an execution of the algorithm 'Descent.' Let $G(\vec{t})$ be as defined in the statement of Lemma 5.3. Then, assuming that $\epsilon \leq 1$, the following bound on the Euclidean norm of $Project\,(BProject(\nabla F(\vec{t}), ProjB), V)$ holds. (We will let B denote the intersection of the 'possibly violated' boundaries, namely $b_i, i \in ProjB$, and write $\pi_{B,V}(\nabla F(\vec{t}))$ for $Project\,(BProject(\nabla F(\vec{t}), ProjB), V)$, in the sequel. Also, we will write in general $||\vec{v}||$ for the Euclidean norm of any vector $\vec{v}$.)

$$||\pi_{B,V}(\nabla F(\vec{t}))|| \geq \frac{G(\vec{t})}{\sqrt{2}}$$

Proof of Sublemma 5.2: There are two cases. First, assume that *ProjB* is empty, namely the gradient directs inside all boundaries and no projection need be taken. In this case, $\pi_{B,V}(\nabla F(\vec{t}))$ equals $\nabla F(\vec{t})$. Let $\vec{v} \in V$ be a point at which F attains global minimum. (When F attains global minimum at a whole section, any point in that section will do.) By the *convexity* of V, the line segment between $\vec{t}$ and $\vec{v}$ is inside V. By the *convexity* of F *in* V, it follows then that the norm of the directional derivative of F at $\vec{t}$ in the direction of $\vec{v}$ is at least $G(\vec{t})/\sqrt{2}$, since the diameter of V is $\sqrt{2}$. Thus the length of the gradient is at least this quantity also.

Next, assume that *ProjB* is non-empty, namely the current position $\vec{t}$ lies within Δ_1 of a number of boundary hyper-planes. We will first argue, assuming that $\vec{t}$ in fact lies (exactly) on the intersection of these boundary hyper-planes (as in the following claim), and later extend the argument to the case in which $\vec{t}$ may be off by (at most) Δ_1.

Claim 5.1: Let $\vec{t} \in V$ be the current position which is on the current boundary B. Let $\vec{e}_\nabla$ be the unit vector in the direction of the gradient (at $\vec{t}$), $\vec{e}_{min}$ be the unit vector in the direction towards a global minimum of F (from $\vec{t}$), and $\vec{e}_\pi$ be the unit vector in the direction of the projection of $\vec{e}_\nabla$ onto B. (When B is dimensionless, namely when it is a single point, let $\vec{e}_\pi$ be the zero vector.) Then, the norm of the directional derivative of F along $\vec{e}_\pi$ is at least as large as that along $\vec{e}_{min}$, i.e.,

$$\nabla F(\vec{t}) \cdot \vec{e}_\pi \geq \nabla F(\vec{t}) \cdot \vec{e}_{min}.$$

We defer the proof of the claim to the appendix.

Given Claim 5.1, the same argument as the first case shows that the norm of $\pi_{B,V}(\nabla F(\vec{t}))$, which is nothing but $\nabla F(\vec{t}) \cdot \vec{e}_\pi$, is at least $G(\vec{t})/\sqrt{2}$. Next we will show how to in effect remove the assumption that $\vec{t}$ lies in B and show in general for any $\vec{t}$ that lies within Δ_1 of B that $\nabla F(\vec{t}) \cdot \vec{e}_\pi \geq \nabla F(\vec{t}) \cdot \vec{e}_{min} - \epsilon/4$. Note that $\epsilon/4$ introduced here is accounted for by $\epsilon/4$ in the definition of $G(\vec{t}) = F(\vec{t}) - (\min_{\vec{t} \in R} F(\vec{t}) + \epsilon/4)$, when applying the argument here to prove Sublemma 5.2. Recall that *CurrB* is set in block 7 of 'Descent' to be the set of indices i such that $t_i \leq \Delta_1$. Recall that *ProjB* consists of just those boundaries b_i in *CurrB* such that the next increment will bring $\vec{t}$ even closer to b_i. Let $\vec{t'} \in V$ be a point at which F attains global minimum. There are two cases. Suppose that $\vec{t'}$ is more than Δ_1 away from any of the boundaries in *ProjB*, i.e. $t'_i > \Delta_1$, $\forall i \in ProjB$. Then, the boundaries in *ProjB* can be moved by at most Δ_1, so that $\vec{t}$ lie in the modified intersection B. In this case $\vec{t'}$ still lies within the modified region, and hence the vector $\vec{e}_{min}$ (unit vector directed from $\vec{t}$ towards $\vec{t'}$) still points within. Thus, the foregoing claim applies, with the modified boundaries playing the role of *ProjB*. Next, suppose that $\vec{t'}$ is within Δ_1 of some of the boundaries in *ProjB*. In this case, we can show that there exists another point $\vec{t''}$ which is at least Δ_1 away from all the boundaries in *ProjB*, and such that its log-loss $L_{p(\vec{t''})}$ exceeds $L_{p(\vec{t'})}$ by at most $\epsilon/4$ on any $(x, y) \in X \times Y$ (assuming $\epsilon \leq 1$). We use a trick similar to the ϵ-Bayesian averaging once again. Let $p(\vec{t'})$ be the p-concept corresponding to $\vec{t'}$. Then, define $\vec{t''}$ as follows.

$$\forall i \leq s,\ t''_i = \left(1 - \frac{\epsilon}{8}\right) t'_i + \frac{\epsilon}{8}\left(\frac{1}{s}\right) \tag{24}$$

Then note the following.

$$\begin{aligned}
&\forall i,\ t''_i \geq \frac{\epsilon}{8s} = \Delta_1 \\
&\forall x \in X\ \forall y \in Y\ \ L_{p(\vec{t''})(y|x)} - L_{p(\vec{t'})(y|x)} \\
&\quad \leq \frac{\epsilon/8}{1 - \epsilon/8} < \frac{\epsilon}{4}, \qquad \text{assuming } \epsilon \leq 4.
\end{aligned}$$

In other words, $\vec{t''}$ is more than Δ_1 away from all boundaries in B such that its corresponding p-concept $p(\vec{t''})$ is at most $\epsilon/4$ worse than $p(\vec{t'})$ on any examples. We can argue in the same manner as in the case that $\vec{t'}$ was Δ_1 away from all boundaries, by using the point $\vec{t''}$ in place of $\vec{t'}$, yielding the conclusion that $\nabla F(\vec{t}) \cdot \vec{e}_\pi \geq \nabla F(\vec{t}) \cdot \vec{e}_{min} - \epsilon/4$.

End of Proof of Sublemma 5.2

We now use Taylor's Theorem for multi-variable functions to derive a lower bound on the improvement in the value of F at each iteration.

Sublemma 5.3 (Taylor): Let $F(\vec{x})$ be a twice differentiable function from an open set U in $\mathbf{R}^s$ into $\mathbf{R}$. Then for an arbitrary $\vec{x} \in U$ and arbitrary $\vec{x} + \vec{h} \in U$, the following holds for some real number θ, $0 < \theta < 1$.

$$\begin{aligned}
&F(\vec{x} + \vec{h}) \\
&\quad = F(\vec{x}) + \nabla F(\vec{x}) \cdot \vec{h} + \frac{1}{2}\vec{h}^T \cdot \nabla^2 F(\vec{x} + \theta\vec{h}) \cdot \vec{h}
\end{aligned}$$

Recall that the increment $\vec{h}$ is given by $\vec{h} = \Delta_2 \pi_{B,V}(\nabla F(\vec{t}))$, where $\Delta_2 = \epsilon^2/64s|Y|^2$. We thus have the following from Sublemma 5.3 for some real θ.

$$\begin{aligned}
&F(\vec{x}) - F(\vec{x} + \vec{h}) \\
&\quad \geq \nabla F(\vec{x}) \cdot \Delta_2 \pi_{B,V}(\nabla F(\vec{x})) \\
&\qquad -\frac{s}{2}||\vec{h}||^2 \max_{ij} |\nabla^2 F(\vec{x} + \theta\vec{h})_{ij}| \\
&\quad = \Delta_2 ||\pi_{B,V}(\nabla F(\vec{x}))||^2 - \frac{s||\vec{h}||^2}{2}\frac{16|Y|^2}{\epsilon^2} \\
&\qquad (\text{since } \vec{y} \cdot \pi_U(\vec{y}) = \pi_U(\vec{y}) \cdot \pi_U(\vec{y}) \\
&\qquad\ \text{holds in general for any subspace } U) \\
&\quad = ||\pi_{B,V}(\nabla F(\vec{x}))||^2 \left(\Delta_2 - \frac{8|Y|^2 s\Delta_2^2}{\epsilon^2}\right) \\
&\quad \geq \frac{1}{2}\Delta_2 ||\pi_{B,V}(\nabla F(\vec{x}))||^2, \left(\text{since}\Delta_2 = \frac{\epsilon^2}{64|Y|^2 s}\right) \\
&\quad = \frac{\epsilon^2}{128s|Y|^2}||\pi_{B,V}(\nabla F(\vec{x}))||^2
\end{aligned}$$

End of Proof of Lemma 5.3

We are now ready to bound the number of iterations i needed for 'Descent' to achieve a desired accuracy.

Lemma 5.4: Let $\vec{t_0}$ be the initial value of $\vec{t}$ and $\vec{t_f}$ be the final value of $\vec{t}$ in an execution of the algorithm 'Descent.' Then the number of iterations (the number of times block 7 is entered) during this time, f, is at most $512s|Y|^2/\epsilon^3$.

Proof of Lemma 5.4: We use the following sublemma.

Sublemma 5.4: Let $\{g_i\}$ be a sequence of non-negative numbers satisfying the following recurrence, where $c > 0$ is a positive constant.

$$g_{i+1} \leq g_i - cg_i^2 \tag{25}$$

Define $\{f_i\}$ to be the sequence given by the following explicit formula.

$$f_i = \frac{1}{c(i + \frac{1}{g_0 c})}$$

Then, for all $i \in \mathbf{N}$, we have

$$g_i \leq f_i$$

Proof of Sublemma 5.4: We prove by induction that if $g_i \leq f_i$ then $g_{i+1} \leq f_{i+1}$. Since $f_0 = g_0$, it suffices to show the inductive step. Assume for contradiction that both of the following hold:

$$g_i \leq f_i \tag{26}$$
$$g_{i+1} > f_{i+1} \tag{27}$$

First note that the following identity holds of $\{f_i\}$ defined above.

$$\begin{aligned} f_i - f_{i+1} &= \frac{1}{c(i + \frac{1}{g_0 c})} - \frac{1}{c(i+1 + \frac{1}{g_0 c})} \\ &= \frac{c}{c(i + \frac{1}{g_0 c})c(i+1 + \frac{1}{g_0 c})} \\ &= c f_i f_{i+1} \end{aligned} \tag{28}$$

Now, by the property (25) of $\{g_i\}$,

$$\begin{aligned} g_{i+1} &\leq g_i - c g_i^2 \\ &\leq g_i - c g_i g_{i+1} \quad (\text{since } 0 \leq g_{i+1} \leq g_i) \\ &= g_i(1 - c g_{i+1}) \\ &\leq f_i(1 - c g_{i+1}) \\ &\quad (\text{from the inductive hypothesis (26)}) \\ &< f_i(1 - c f_{i+1}) \quad (\text{from the assumption (27)}) \\ &= f_i - c f_i f_{i+1} \\ &= f_{i+1} \qquad (\text{by (28)}) \end{aligned} \tag{29}$$

which contradicts our assumption (27). Hence, we conclude that if $g_i \leq f_i$ then we must also have $g_{i+1} \leq f_{i+1}$.

End of Proof of Sublemma 5.4

Now let $f_0 = \alpha$ and set $d = \lceil (1/c) \cdot (1/\beta - 1/\alpha) \rceil$ for some non-negative β. Then,

$$\begin{aligned} d &\geq \frac{1}{c}\left(\frac{1}{\beta} - \frac{1}{\alpha}\right) \\ d &\geq \frac{1}{c\beta} - \frac{1}{\alpha c} \\ c\left(d + \frac{1}{\alpha c}\right) &\geq \frac{1}{\beta} \\ \frac{1}{c(d + \frac{1}{\alpha c})} &\leq \beta \end{aligned} \tag{30}$$

Thus, from Sublemma 5.4, $f_d \leq \beta$, and hence $d = \lceil (1/\beta - 1/\alpha)/c \rceil$ bounds from above the number of iterations needed to improve the value of F from α down to β.

Now let $\vec{t}_0, \vec{t}_1, \ldots, \vec{t}_f$ be the positions (values of $\vec{t}$) taken in succession during the course of an execution of 'Descent.' Then by Lemma 5.3, we have for all $i < f$,

$$G(\vec{t}_i) - G(\vec{t}_{i+1}) \geq \frac{\epsilon^2}{128 s |Y|^2} G(\vec{t}_i)^2$$

Thus, we can bound $G(\vec{t}_i)$ by the sequence f_i for appropriate values of c and g_0. In particular, since we are using $CL(q_1^{(\epsilon)}, \ldots, q_s^{(\epsilon)})$ as the hypothesis space, the initial value of $G(\vec{t})$ is at most $\log(8/\epsilon)$, and the final desired value of $G(\vec{t})$ is $\epsilon/4$. The constant corresponding to c above is $\epsilon^2/128 s|Y|^2$. Therefore by plugging in these values into $d = \lceil (1/\beta - 1/\alpha)/c \rceil$,, we see that the number of iterations f required for the algorithm 'Descent' is bounded from above as follows.

$$\begin{aligned} f &\leq \left\lceil \frac{128 s |Y|^2}{\epsilon^2} \left(\frac{4}{\epsilon} - \frac{1}{\log \frac{8}{\epsilon}} \right) \right\rceil \\ &\leq \left\lceil \frac{512 s |Y|^2}{\epsilon^3} \right\rceil \end{aligned}$$

End of Proof of Lemma 5.4

From the above, we conclude that the number of iterations required for Descent to converge to a hypothesis which is ϵ close to one minimizing the empirical log loss is $O(s|Y|^2/\epsilon^3)$. The amount of computation required at each iteration is $O(s^3 + mq(n))$, due mostly to the projection operation which involves matrix multiplications ($O(s^3)$), and the calculation of the gradient at each position ($O(mq(n))$), leading to the upper bound on the overall running time of order $O((s^4|Y|^2/\epsilon^3) \cdot mq(n))$.

Q.E.D

6. Concluding Remarks

We conclude by listing some open problems inspired by this research. First, does the equivalence between PAC learnability with respect to the quadratic distance and the KL-divergence also hold for the *robust* case? Second, can we improve significantly the sample complexity bounds we give? Third, can we improve the bound we give on the running time of the learning algorithm for convex linear combinations? Fourth, are there simple on-line learning algorithms similar to the ones given in [9] for learning convex linear combinations with good total loss bounds?

Acknowledgements

We thank David Haussler for many enlightening conversations and for teaching us many techniques. In particular, an idea similar to the ϵ-Bayesian averaging was first suggested to us by him. We also thank Dr. Kenji Yamanishi of NEC Computer & Communication Media Research for fruitful discussions. Finally, we thank

the anonymous reviewers for numerous invaluable comments and corrections which greatly helped improve the paper.

References

[1] D. Angluin and P.D. Laird, "Learning from noisy examples," Machine Learning, vol.2, no.4, pp.343–370, Kluwer Academic Publishers, Boston, 1988.
[2] N. Abe, J. Takeuchi, and M.K. Warmuth, "Polynomial learnability of probabilistic concepts with respect to the Kullback-Leibler divergence," Proc. 1991 Workshop on Computational Learning Theory. Morgan Kaufmann, San Mateo, California, Aug. 1991.
[3] N. Abe and M.K. Warmuth, "On the computational complexity of approximating probability distributions by probabilistic automata," Machine Learning, vol.9, no.2/3, pp.205–260, 1992.
[4] D. Haussler, "Decision theoretic generalizations of the PAC model for neural net and other learning applications," Information and Computation, vol.100, no.1, Sept. 1992.
[5] D.P. Helmbold, R.E. Schapire, Y. Singer, and M.K. Warmuth, "A comparison of new and old algorithms for a mixture estimation problem," Machine Learning, vol.27, 1997.
[6] M. Kearns and R. Schapire, "Efficient distribution-free learning of probabilistic concepts," J. Computer and System Sciences, vol.48, no.3, June 1994, A special issue on 31st IEEE Conference on Foundations of Computer Science.
[7] M. Kearns, R. Schapire, and L. Sellie, "Toward efficient agnostic learning," Proc. Fifth Annual ACM Workshop on Computational Learning theory, Aug. 1992.
[8] S. Kullback, "A lower bound for discrimination in terms of variation," IEEE Trans. Inf. Theory, vol.13, no.1, pp.126–127, 1967.
[9] N. Littlestone, P. Long, and M. Warmuth, "On-line learning of linear functions and iterative solution of linear systems," Proc. 24th ACM Symposium on the Theory of Computation, 1990.
[10] D. Pollard, Convergence of Stochastic Processes, Springer-Verlag, 1984.
[11] E. Polak, Computational methods in optimization, Academic Press, 1971.
[12] D. Pollard, Empirical Processes: Theory and Applications, Institute of Mathematical Statistics and American Statistical Association, 1990.
[13] J.B. Rosen, "The gradient projection method for nonlinear programming. part i. linear constraints," J. S.I.A.M., vol.8, no.1, pp.181–217, 1960.
[14] R.E. Schapire, The design and analysis of efficient learning algorithms, Ph.D. Thesis, M.I.T., 1990.
[15] G. Strang, Linear Algebra and its Applications, Academic Press, 1976.
[16] L.G. Valiant, "A theory of the learnable," Commun. ACM, vol.27, pp.1134–1142, 1984.
[17] V.N. Vapnik, "Inductive principles of the search for empirical dependences (methods based on weak convergence of probability measures)," Proc. 2nd Workshop on Computational Learning Theory, Morgan Kaufmann, 1989.
[18] K. Yamanishi, "A learning criterion for stochastic rules," Machine Learning, vol.9, no.2/3, pp.165–203, 1992.

Appendix A: Proof of Claim 5.1

Proof of Claim 5.1: Since the convex subspace Z is an $s-1$ dimensional hyper-plane we will introduce a new $s-1$-dimensional coordinate system to denote the elements of Z. In particular, given an orthonormal basis of $\mathbf{R}^s$, one can rotate any $s-1$ of the s basis vectors to get a basis for Z, as shown below. Let $\vec{e_s} \in Z$ be the vector $(1/\sqrt{s}, \ldots, 1/\sqrt{s})^T$. We can find an orthonormal basis of $\mathbf{R}^s$ $\{\vec{e_j} : j = 1, \ldots, s\}$ that includes $\vec{e_s}$ by a standard technique. (See for example [15], pp. 129). Using $\{\vec{e_1}, \ldots, \vec{e_s}\}$ as the new basis, an arbitrary vector $\vec{t}$ in $\mathbf{R}^s$ can now be written as $\vec{u} = (u_1, .., u_s)^T$ where u_i are the new coordinates, namely,

$$\vec{t} = A \cdot \vec{u}, \text{ where } A = [\vec{e_1}\vec{e_2} \ldots .\vec{e_s}].$$

Since A is orthonormal, $A^{-1} = A^T$, and hence we also have

$$\vec{u} = A^T \cdot \vec{t} \qquad (\text{A}\cdot 1)$$

If we let g denote the linear transformation given by A, we can use $\vec{u}$ to denote the elements of $g(Z)$. Note that the s-th row of A^T is $\vec{e_s}^T = (1/\sqrt{s}, \ldots, 1/\sqrt{s})$. Hence, for an arbitrary $\vec{t} \in Z$ (i.e. one satisfying the convex condition $\sum_{i=1}^{s} t_i = 1$), the s-th compoment of $\vec{u} = A^T \cdot \vec{t}$ equals $(1/\sqrt{s}) \cdot (\sum_{i=1}^{s} t_i) = 1/\sqrt{s}$, and is constant. We will thus ignore the s-th component of $\vec{u}$ and the last component of A and just use the first $s-1$ components, in what follows. We will let V^* denote the $s-1$-dimensional sub-space of V obtained by ignoring the last (s-th) component of V, and let $\vec{u}^*$, $\vec{e}^*_{\nabla}$, $\vec{e}^*_{min}$, and $\vec{e}^*_{\pi}$ denote respectively the vectors obtained by dropping the s-th component of $\vec{u}$, $\vec{e}_{\nabla}$, $\vec{e}_{min}$, and $\vec{e}_{\pi}$, *all in the new coordinate.*

We have in effect transformed the original s dimensional search space with the convex constraint into an $s-1$ dimensional space which already embeds such a constraint. The only constraints on the region we have left, therefore, are the non-negativity conditions on the components t_i of $\vec{t}$. These constraints translate to the following s boundary conditions, where we let A_i denote the i-th row of A.

$$\forall 1 \le i \le s \ : \ A_i \cdot \vec{u} \ge 0 \qquad (\text{A}\cdot 2)$$

The s boundary hyper-planes defined by $A_i \cdot \vec{u} = 0$ correspond to the boundaries b_i in the new coordinates, and as hyper-planes of V^*, these boundaries will be denoted b^*_i. Now, without loss of generality, let $ProjB = \{1, \ldots, q\}$, $q \le s-1$. Let B^* denote the intersection of the q boundaries, b^*_i, $i \in ProjB$. Then note that the subspace of $g(Z)$ that is orthogonal to B^* is a q-dimensional space. (We will call this subspace the orthogonal space of B^*, and denote it by $B^{*\perp}$ in the sequel.) The q normal vectors, $N = \{\vec{\nu_i} : i = 1, \ldots, q\}$,

of the boundaries b_i^*, $i = 1, \ldots, q$, span the orthogonal space $B^{*\perp}$. In terms of the matrix A which defines the transformation from $\vec{u}$ to $\vec{t}$, each $\vec{\nu_i}$ is a row of A, except the s-th component is dropped. That is, $\vec{\nu_i} = (A_{j,1}, \ldots, A_{j,s-1})^T$ for some j. Since the columns of A are orthonormal, so are its rows. As the $\vec{\nu_i}$ are constructed by dropping the last component $1/\sqrt{s}$ of the rows of A, which are s orthonormal vectors, we can evaluate the inner product $\vec{\nu_i} \cdot \vec{\nu_j}$ for arbitrary i, j as follows.

$$\forall i, j \le q,\ i \ne j,\ \vec{\nu_i} \cdot \vec{\nu_j} = -1/s \qquad (\text{A}\cdot 3)$$

$$\forall i \le q,\ \vec{\nu_i} \cdot \vec{\nu_i} = 1 - 1/sn \qquad (\text{A}\cdot 4)$$

It is known that there exists a 'dual basis' $M = \{\vec{\mu_i} : i = 1, \ldots, q\}$, distinct from N that also spans $B^{*\perp}$. That is, M is a basis for $B^{*\perp}$ satisfying the following condition

$$\forall i\ \forall j \le q,\ \vec{\nu_i} \cdot \vec{\mu_j} = \delta_{ij} \qquad (\text{A}\cdot 5)$$

where we used δ_{ij} to denote 'Kronecker's delta,' defined by

$$\delta_{ij} = 0 \quad \text{if} \quad i \ne j \quad \text{and} \quad \delta_{ii} = 1$$

In particular, we can express M in terms of N as follows.

$$\forall i \le q\ \ \vec{\mu_i} = \vec{\nu_i} + \frac{1}{s-q} \sum_{j=1}^{q} \vec{\nu_j}$$

We can verify that (A·5) indeed holds, by using (A·3) and (A·4). First, for all $i, j \le q$ such that $i \ne j$,

$$\begin{aligned}
&\vec{\nu_i} \cdot \vec{\mu_j} \\
&\quad = \vec{\nu_i} \cdot \left(\vec{\nu_j} + \frac{1}{s-q} \sum_{k=1}^{q} \vec{\nu_k} \right) \\
&\quad = (\vec{\nu_i} \cdot \vec{\nu_j}) + \frac{1}{s-q} \sum_{k=1}^{q} (\vec{\nu_i} \cdot \vec{\nu_k}) \\
&\quad = \left(-\frac{1}{s} \right) + \frac{1}{s-q} \left(1 - \frac{q}{s} \right) \quad \text{(by (A·3) and (A·4))} \\
&\quad = 0
\end{aligned}$$

Next, for any $i \le n$, we have:

$$\begin{aligned}
\vec{\nu_i} \cdot \vec{\mu_i} &= \vec{\nu_i} \cdot \left(\vec{\nu_i} + \frac{1}{s-q} \sum_{k=1}^{q} \vec{\nu_k} \right) \\
&= \left(1 - \frac{1}{s} \right) + \frac{1}{s-q} \left(1 - \frac{q}{s} \right) \\
&= 1
\end{aligned}$$

Another important property of M, which we will make use of later, is that $\vec{\mu_i} \cdot \vec{\mu_j} > 0$ for all i and j, as demonstrated below.

$$\begin{aligned}
\vec{\mu_i} \cdot \vec{\mu_j} &= \left(\vec{\nu_i} + \frac{1}{s-q} \sum_{k=1}^{q} \vec{\nu_k} \right) \left(\vec{\nu_j} + \frac{1}{s-q} \sum_{k=1}^{q} \vec{\nu_k} \right) \\
&= \vec{\nu_i} \cdot \vec{\nu_j} + \frac{1}{s-q} \sum_{k=1}^{q} \vec{\nu_k} \cdot \vec{\nu_j} \\
&\quad + \frac{1}{s-q} \sum_{k=1}^{q} \vec{\nu_i} \cdot \vec{\nu_k} \\
&\quad + \frac{1}{(s-q)^2} \left(\sum_{k=1}^{q} \vec{\nu_k} \right) \cdot \left(\sum_{k=1}^{q} \vec{\nu_k} \right) \\
&= \delta_{ij} - \frac{1}{s} + \frac{2(s-q)}{(s-q)s} + \frac{q(s-q)}{(s-q)^2 s} \\
&= \delta_{ij} - \frac{1}{s} + \frac{2}{s} + \frac{q}{(s-q)s} \\
&= \delta_{ij} + \frac{1}{s-q} > 0, \qquad \text{since } s - q \ge 2.
\end{aligned} \qquad (\text{A}\cdot 6)$$

Now recall that what we wish to show is $\nabla F(\vec{u}^*) \cdot \vec{e}_\pi^* \ge \nabla F(\vec{u}^*) \cdot \vec{e}_{min}^*$. This is equivalent to the following:

$$\vec{e}_\nabla^* \cdot \vec{e}_\pi^* \ge \vec{e}_\nabla^* \cdot \vec{e}_{min}^* \qquad (\text{A}\cdot 7)$$

Since N spans the orthogonal space of B^*, an arbitrary vector $\vec{x} \in \mathbf{R}^n$ can be written in the following form:

$$\vec{x} = \sum_{i=1}^{q} c_i \vec{\mu_i} + \lambda \vec{\beta}$$

where $\lambda\vec{\beta}$ is the projection of $\vec{x}$ onto B^*, $\vec{\beta}$ is a unit vector and $\lambda \ge 0$. Furthermore, the direction of $\vec{x}$ 'violates' the boundary b_i^* (i.e. $\vec{\nu_i} \cdot \vec{x} < 0$) if and only if the corresponding coefficient c_i is negative. This can be seen by taking the inner product of $\vec{x}$ with the normal vector $\vec{\nu_j}$ of an arbitrary boundary hyper-plane.

$$\begin{aligned}
\vec{\nu_j} \cdot \vec{x} &= \sum_{i=1}^{q} \vec{\nu_j} \cdot (c_i \vec{\mu_i}) + \lambda \vec{\nu_j} \cdot \vec{\beta} \\
&= c_j + \lambda (\vec{\nu_j} \cdot \vec{\beta}) \qquad \text{(by (A·5))} \\
&= c_j, \qquad \text{(since } \vec{\nu_j} \text{ and } \vec{\beta} \text{ are orthogonal).}
\end{aligned}$$

Thus we see that $\vec{x}$ violates b_j^* if and only if $c_j < 0$.

Now let us write both $\vec{e}_\nabla^*, \vec{e}_{min}^*$ in the above form

$$\vec{e}_\nabla^* = \sum_{i=1}^{q} (-\alpha_i) \vec{\mu_i} + \lambda_1 \vec{e}_\pi^*$$

$$\vec{e}_{min}^* = \sum_{i=1}^{q} \beta_i \vec{\mu_i} + \lambda_2 \vec{f_\pi}$$

where $\vec{e}_\pi^*$ and $\vec{f_\pi}$ are unit vectors in B^* (except they are zero vectors when $q = s - 1$ and B^* is a single point). In the above, all the α_i and β_i are non-negative because the vector $\vec{e}_\nabla^*$ violates all boundaries in B^*, and $\vec{e}_{min}^*$, which is assumed to direct towards an interior point

from $\vec{u}^*$, violates none of them. Also note that since both $\vec{e}^*_\nabla$ and $\vec{e}^*_{min}$ are unit vectors, their projections are of at most unit length, and hence $\lambda_1 \leq 1$ and $\lambda_2 \leq 1$.

With these properties at hand, we are now ready to verify (A·7), i.e. to show that $\vec{e}^*_\nabla \cdot \vec{e}^*_{min} \leq \vec{e}^*_\nabla \cdot \vec{e}^*_\pi$. First, when $q = s-1$, $\vec{e}^*_\nabla = \sum_{i=1}^q (-\alpha_i)\vec{\mu}_i$ and $\vec{e}^*_{min} = \sum_{i=1}^q \beta_i \vec{\mu}_i$, and thus it follows immediately that $\vec{e}^*_\nabla \cdot \vec{e}^*_{min} \leq 0$, and $\vec{e}^*_\nabla \cdot \vec{e}^*_\pi = 0$, and that the claim holds.

When $q < s-1$, we first bound $\vec{e}^*_\nabla \cdot \vec{e}^*_{min}$ from above as follows.

$$\begin{aligned}
&\vec{e}^*_\nabla \cdot \vec{e}^*_{min} \\
&= \left(\sum_{i=1}^q (-\alpha_i)\vec{\mu}_i + \lambda_1 \vec{e}^*_\pi\right) \cdot \left(\sum_{i=1}^q \beta_i \vec{\mu}_i + \lambda_2 \vec{f}_\pi\right) \\
&= \left(-\sum_{i,j=1}^n \alpha_i \beta_j \vec{\mu}_i \cdot \vec{\mu}_j\right) + \lambda_1 \lambda_2 \vec{e}^*_\pi \cdot \vec{f}_\pi \\
&\leq \lambda_1 \lambda_2, \qquad \text{from (A·6)}
\end{aligned}$$

In deriving the second line from the first line, we used the fact that the $\vec{\mu}_i$ are orthogonal to B^* and hence $\vec{e}^*_\pi \cdot \vec{\mu}_i$ and $\vec{\mu}_i \cdot \vec{f}_\pi$ are zero. In deriving the last line, we used the fact that any product $\vec{\mu}_i \cdot \vec{\mu}_j$ is positive (A·6). Next, $\vec{e}^*_\nabla \cdot \vec{e}^*_\pi$ can be calculated as follows.

$$\begin{aligned}
\vec{e}^*_\nabla \cdot \vec{e}^*_\pi &= \sum_{i=1}^q (-\alpha_i)\vec{\mu}_i \cdot \vec{e}^*_\pi + \lambda_1 \vec{e}^*_\pi \cdot \vec{e}^*_\pi \\
&= \lambda_1
\end{aligned}$$

Here we used the fact that $\vec{\mu}_i$ and $\vec{e}^*_\pi$ are orthogonal. Recalling that $0 \leq \lambda_1, \lambda_2 \leq 1$, we conclude that $\vec{e}^*_\nabla \cdot \vec{e}^*_\pi = \lambda_1 \geq \lambda_1 \lambda_2 \geq \vec{e}^*_\nabla \cdot \vec{e}^*_{min}$, and hence, we have shown that $\vec{e}^*_\nabla \cdot \vec{e}^*_{min} \leq \vec{e}^*_\nabla \cdot \vec{e}^*_\pi$.

Appendix B: Inequality between KL Divergence and Quadratic Distance

Proposition Appendix B.1: The following inequality holds for arbitrary stochastic rules, p and q.

$$2d^D_{KL}(p,q) \geq d^D_Q(p,q).$$

Proof: Define the (extended) KL divergence and the (extended) quadratic ditance between $p \in [0,\infty)^s$ and $q \in (0,\infty)^s$ as $d_{KL}(p,q) = \sum_{i=1}^s p_i \log(p_i/q_i)$ and $d_Q(p,q) = \sum_{i=1}^s (p_i - q_i)^2$. (Recall the convention $0 \log 0 = 0$.) Note that the restrictions of these extended definitions to $V \times (V \cap (0,\infty)^s)$ give the ordinary KL divergence and the quadradic distance, where V is the s-dimensional probability convex, i.e. $V = \{p : \sum_{i=1}^s p_i = 1,\ \forall i,\ p_i \geq 0\}$ (c.f. the definition of V in Sect. 5).

Applying Taylor's theorem to $d_{KL}(p,q)$ as a function of p around the point $p = q$, we obtain

$$\begin{aligned}
&d_{KL}(p,q) \\
&= \sum_i \left.\frac{\partial d_{KL}(p,q)}{\partial p_i}\right|_{p=q} (p_i - q_i) \\
&\quad + \frac{1}{2}\sum_{i,j} \left.\frac{\partial^2 d_{KL}(p,q)}{\partial p_i \partial p_j}\right|_{p=\tilde{p}} (p_i - q_i)(p_j - q_j) \\
&= \sum_i (p_i - q_i) + \frac{1}{2}\sum_i \frac{(p_i - q_i)^2}{\tilde{p}_i},
\end{aligned}$$

where $\tilde{p} = \alpha p + (1-\alpha)q$ $(\alpha \in [0,1])$.

Noting that $\sum_i (p_i - q_i) = 0$ and $\tilde{p}_i \leq 1$ hold for $p \in V$ and $q \in V \cap (0,\infty)^s$, we have

$$\begin{aligned}
&\forall p \in V,\ \forall q \in V \cap (0,\infty)^s, \\
&\quad d_{KL}(p,q) \geq \frac{1}{2}\sum_i (p_i - q_i)^2 = \frac{1}{2} d_Q(p,q),
\end{aligned}$$

which clearly implies

$$2d^D_{KL}(p,q) \geq d^D_Q(p,q). \qquad \square$$

Naoki Abe received the B.S. and M.S. degrees from Massachusetts Institute of Technology in 1984, and the Ph. D. degree from the University of Pennsylvania in 1989, all in Computer Science. After holding a post doctoral researcher position at the University of California, Santa Cruz, he joined NEC Corporation in 1990, where he is currently research manager in Computer & Communication Media Research. He is also visiting associate professor in the department of computational intelligence and systems science of Tokyo Institute of Technology. His research interests include theories and applications of machine learning to various domains, including natural language processing, data mining, and internet information filtering.

Jun-ichi Takeuchi was born in Tokyo, Japan, in 1964. He received the B.Sc. degree in physics and the Dr.Eng. degree in mathematical engineering from University of Tokyo in 1989 and 1996, respectively. He joined NEC Corporation in 1989, where he is currently an Assistant Manager in Computer & Communication Media Research. From 1996 to 1997 he was a Visiting Research Scholar at Department of Statistics, Yale University. His research interest includes mathematical statistics, computational learning theory, machine learning, information theory and data mining.

Manfred K. Warmuth is a professor in the Computer Science department at the University of California at Santa Cruz. He received his Ph.D. in Computer Science in 1981 from the University of Colorado at Boulder. He is an editor of the journal of *Machine Learning* and is involved with the organization of the annual ACM Conference on Computational Learning Theory. His current research interests include machine learning, on-line learning, game theory, neural networks, and statistics.