A Bayesian Approach to Online Learning

Manfred Opper

Neural Computing Research Group, Aston University, Birmingham B4 7ET, UK.

Abstract

Online learning is discussed from the viewpoint of Bayesian statistical inference. By replacing the true posterior distribution with a simpler parametric distribution, one can define an online algorithm by a repetition of two steps: An update of the approximate posterior, when a new example arrives, and an optimal projection into the parametric family. Choosing this family to be Gaussian, we show that the algorithm achieves asymptotic efficiency. An application to learning in single layer neural networks is given.

1 Introduction

Neural networks have the ability to learn from examples. For *batch learning*, a set of training examples is collected and subsequently an algorithm is run on the entire training set to adjust the parameters of the network. On the other hand, for many practical problems, examples arrive sequentially and an instantaneous action is required at each time. In order to save memory and time this action should not depend on the entire set of data which have arrived sofar. This principle is realized in *online* algorithms, where usually only the last example is used for an update of the network's parameters. Obviously, some amount of information about the past examples is discarded in this approach. Surprisingly, recent studies showed that online algorithms can achieve a similar performance as batch algorithms, when the number of data grows large (Biehl & Riegler 1994; Barkai et al 1995; Kim & Sompolinsky 1996).

In order to understand the abilities and limitations of online algorithms, the question of optimal online learning has been raised. For algorithms which are based on a weigthed Hebbian rule, one has sought for optimal weight functions which yield the highest local (i.e. instantaneous) or global reduction of the average generalization error. Within the thermodynamic limit framework of statistical mechanics, this goal has been achieved for highly symmetric distributions of inputs by (Kinouchi & Caticha 1992; Copelli & Caticha 1995) for local optimization and by (Saad & Rattray 1997) for global optimization.

Within this approach, the dynamics of online learning can be described by a few macroscopic order parameters. The results of these studies have shown that in some cases online algorithms can even learn with *the same* asymptotic speed (Biehl et al 1995; Van den Broeck & Reimann 1996) as optimized batch algorithms. Unfortunately, it is not clear how to generalize such approaches to general learning problems outside the thermodynamic limit framework. The reason is twofold. First, very specific assumptions about the probability distributions of network inputs have to made in order to allow for an introduction of order parameters. Second, the optimization also requires some knowledge (e.g. the generalization error) about the unknown teacher rule to be learnt. This information is usually not available in a concrete learning problem. Nevertheless, these statistical mechanics approaches are highly important. Their results can give an idea of what online algorithms can achieve in an idealized scenario and have also motivated further studies (Amari 1996; Opper 1996) like the one presented in this chapter.

In the following, we will discuss in more detail a Bayesian approach to online learning which has been introduced in (Opper 1996) and generalized in (Winther & Solla 1997). In this framework, it is possible to define optimal online learning as an approximation to batch learning, where in each step the loss of information from discarding previous examples is minimized. Such an optimization can be carried out without making assumptions about the distributions of inputs.

The chapter is organized as follows. In section 2, learning from random examples is described within the framework of statistical inference. Section 3 briefly reviews different statistical optimality criteria with a special emphasis on *efficient estimation*. In section 4, online learning is discussed as an approximation to maximum likelihood estimation and a recent approach to efficient online learning is introduced. Sections 5, 6 and 7 give an introduction into Bayesian inference, introduces Bayesian online learning and the explicit form of the algorithm within a Gaussian parametric ansatz. The asymptotic average case performance of the algorithm is calculated in section 8. Section 9 contains the explicit realization of the algorithm for the case of a single layer perceptron. The chapter concludes with an outlook in section 10.

2 Learning and Statistical Inference

The problem of learning in neural networks can be treated within the framework of statistical inference. One assumes that t data $D_t = (y_1, \ldots, y_t)$ are generated independently at random according to a distribution

$$P(D_t|\theta) = \prod_{k=1}^{t} P(y_k|\theta). \tag{2.1}$$

θ is an unknown parameter which has to be estimated from D_t. For a noisy classification problem in a single layer neural net e.g., we set $y = (S, \mathbf{x}) =$ (label, inputs). Here $\mathbf{x} \doteq (x_1, \ldots, x_n)$ is an N dimensional vector of input features and the parameter $\theta \doteq (\theta_1, \ldots, \theta_N)$ is an N dimensional vector of network weights. In this case, a popular model is

$$P(y|\theta) = \phi(S\,\theta \cdot \mathbf{x}) f(\mathbf{x}), \tag{2.2}$$

where $\theta \cdot \mathbf{x} \doteq \sum_{i=1}^{N} \theta_i x_i$ is the inner product of weights and inputs and f is the density of inputs. Usually $\phi(h)$ is a smooth sigmoidal function which increases from zero at $h = -\infty$ to 1 at $h = \infty$. For a regression problem, we can set $y = (z, \mathbf{x}) =$ (function value, inputs). If a Gaussian noise model is assumed, we have

$$P(y|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(z - r(\theta, \mathbf{x}))^2} f(\mathbf{x}), \tag{2.3}$$

where $r(\theta, \mathbf{x})$ is the regression function. In the following, we will usually use the symbol θ^* for the true value of the parameter (the one that stands for the distribution which generates the examples) and $\hat{\theta}$ for an estimate of this parameter.

3 Optimal Learning

Before discussing the problem of optimal online learning, we will briefly review (Vapnik 1982; Schervish 1995) a few optimality criteria which can be used to assess the quality of an estimation procedure. Obviously, there is no *uniformly* optimal estimation strategy. An algorithm which always makes the same prediction for the unknown probability independently of the data, is optimal for one single task (the one where the data come just from the distribution which the algorithm predicts) but will usually perform badly in general. Uniform optimality can be achieved only within special subclasses of estimators. Well known cases are best unbiased density estimators for exponential families of probability densities. Unbiasedness of an estimator means that the estimate averaged over the distribution of the training set gives the true density which generated the data. However, it is not clear at all that one should restrict estimators to unbiased ones. One might prefer an estimator with a small bias and small variance over an unbiased one with a very large variance.

Various optimality criteria are known in statistics. In the *minimax principle* one optimizes the prediction for the worst true density. In the *Bayesian* approach, one can define an average case optimality, where the average is over a prior distribution $p(\theta^*)$ of true parameters θ^*. We will come back to an online approximation to a Bayesian approach later.

For the case of parameter estimation, *efficiency* is another important criterion for probabilistic models which depend smoothly on the parameters. If

parameter estimation is restricted to *unbiased estimators* $\hat{\theta}$ (i.e. if the estimate $\hat{\theta}$ obeys $E_{D_t}\hat{\theta} = \theta^*$), then the famous *Rao-Cramér inequality* limits the speed at which the estimate $\hat{\theta}$ approaches the true parameter θ^* on average. For a single (scalar) parameter it simply reads

$$E_{D_t}\left(\hat{\theta} - \theta^*\right)^2 \geq \frac{1}{t \int dy P(y|\theta^*) \left[\frac{d}{d\theta^*} \ln P(y|\theta^*)\right]^2}. \tag{3.1}$$

E_{D_t} denotes the expectation over datasets. The generalization to an N dimensional vector of parameters is possible. For any real vector $(z_1, \ldots, z_N)$, we have the inequality

$$E_{D_t}\left(\sum_i z_i(\hat{\theta}_i - \theta_i^*)\right)^2 = \sum_{ij} z_i z_j E_{D_t}\left((\hat{\theta}_i - \theta_i^*)(\hat{\theta}_j - \theta_j^*)\right) \geq \frac{1}{t}\sum_{ij} z_i z_j (J^{-1}(\theta^*))_{ij}, \tag{3.2}$$

where

$$J_{ij}(\theta^*) = \int dy P(y|\theta^*) \partial_i \ln P(y|\theta^*) \partial_j \ln P(y|\theta^*)$$

is the *Fisher Information* matrix. Partial derivatives are with respect to the components of θ^*. If we take nonnegative numbers for the z_i, we can interprete the left hand side of (3.2) as a squared weighted average of the individual error components $\hat{\theta}_i - \theta_i^*$. Estimators which fulfill these relations with an *equality*, are called *efficient*. Since the proof of the inequalities requires unbiasedness, it is not clear at all why efficiency is important. Biased estimators may in some cases violate (3.2) and achieve a better performance. However, this is not true asymptotically. It can be shown that when the number of data grows large, then for almost all true parameters θ^*, no estimator can beat the Rao-Cramér inequality. As has been proved e.g. by (LeCam 1953) (under smoothness assumptions), the Lebesgue measure of the set of all parameters θ^* for which we can have *superefficiency*, i.e. a violation of (3.2), goes to zero asymptotically. Hence, one reasonable requirement for a good algorithm is *asymptotic efficiency* which by (3.2) means

$$E_{D_t}(\hat{\theta}_i - \theta_i^*)(\hat{\theta}_j - \theta_j^*) = \frac{1}{t}(J^{-1}(\theta^*))_{ij}, \tag{3.3}$$

in the limit $t \to \infty$.

4 Online Learning

Often, learning algorithms for estimating the unknown parameter θ^* are based on the principle of *Maximum Likelihood* (ML). It states that we should choose

a parameter θ which maximizes the likelihood $P(D_t|\theta)$ of the observed data. Under weak assumptions, ML estimators are asymptotically efficient. As a learning algorithm, one can use e.g. a gradient descent algorithm and iterate

$$\theta_i' - \theta_i = \eta \; \partial_i \sum_{k=1}^{t} \ln P(y_k|\theta) = -\eta \; \partial_i \sum_{k=1}^{t} E_T(y_k|\theta) \tag{4.1}$$

until convergence is achieved. Here, $E_T(y_k|\theta)$ defines the training energy of the examples to be minimzed by the algorithm. When a new example y_{t+1} is received, the ML procedure requires that the learner has to update her estimate for θ using *all previous* data. Hence D_t has to be stored in a memory. The goal of online learning is to calculate a new estimate $\hat{\theta}(t+1)$ which is only based on the new data point y_{t+1}, the old estimate $\hat{\theta}(t)$ (and possibly a set of other auxiliary quantities which have to be updated at each time step, but are much smaller in number than the entire set of previous training data). A popular idea is to use a procedure similar to (4.1), but to replace the training energy of all examples $\sum_{k=1}^{t} E_T(y_k|\theta)$ by the training energy of the most recent one. Hence, we get

$$\begin{aligned} \hat{\theta}_i(t+1) - \hat{\theta}_i(t) &= \eta(t) \; \partial_i \ln P(y_{t+1}|\hat{\theta}(t)) \\ &= -\eta(t) \; \partial_i E_T(y_{t+1}|\hat{\theta}(t)). \end{aligned} \tag{4.2}$$

The choice of the learning rate $\eta(t)$ is important. If the algorithm should converge asymptotically, η must be decreased during learning. A schedule $\eta \propto 1/t$ yields the fastest rate of convergence, but the prefactor must be chosen with care, in order to avoid that the algorithm gets stuck away from the optimal parameter. Another choice is an adaptive η (Barkai et al 1995), which depends on the performance of the algorithm and can be used in the case of temporal changes of the distribution.

A recent modification of (4.2) has been introduced by Amari (Amari 1996; Amari 1997), who replaces the scalar learning rate $\eta(t)$ by a tensor. This idea may be derived from the fact that the online training energy contains only information about the last example, and the change of the estimate of the distribution due to a change $\Delta\theta$ of the parameter should not be too large. The new idea is to define a measure for distances $||\Delta\theta||$ in the parameter space which reflects distances between probability distributions and is invariant against transformations of the parameters. A simple Euklidian distance will not satisfy this condition. One can be guided by the principle that the distance between two distributions should reflect how well they can be distinguished by an estimation based on random data. This can be achieved by defining the metric in parameter space by

$$||d\theta||^2 \propto \sum_{ij} d\theta_i J_{ij}(\theta) d\theta_j. \tag{4.3}$$

Assuming that the probability distribution of efficient estimators is Gaussian (at large t) with a covariance given by (3.3), the probability density that a

point close to the true value θ will be the estimate for θ, depends only on the distance between the two points. Based on such ideas, S. Amari (Amari 1985) has developped a beautiful differential geometric approach to statistical inference. In the context of online learning, he proposed the so called *natural gradient* algorithm (Amari 1996; Amari 1997), where the update is defined by a minimization of the training energy under the condition that $||\Delta\theta||^2$ is kept fixed. Solving the constrained variational problem for small $\Delta\theta$ yields

$$\theta_i(t+1) - \theta_i(t) = \gamma_t \sum_j (J^{-1}(\theta(t))_{ij} \partial_j \ln P(y_{t+1}|\theta(t)). \tag{4.4}$$

The differential operator $\sum_j (J^{-1}(\theta(t))_{ij} \partial_j$ is termed natural gradient. For the choice $\gamma_t = \frac{1}{t}$, one can show that the online algorithm yields *asymptotically efficient* estimation (Amari 1996; Opper 1996).

5 The Bayesian Approach

In the Bayesian approach to statistical inference, the degrees of prior belief or plausibility of parameters are expressed within probability distributions, the so called prior distributions (or priors) $p(\theta)$. Once this idea is accepted, subsequent inferences can be based on *Bayes rule* of probability. Formally, we may think that data are generated by a two step process: First, the true parameter θ is drawn at random from the prior distribution $p(\theta)$. Second, the data are drawn at random from $P(D_t|\theta)$. Bayes rule yields the conditional probability density (*posterior*) of the unknown parameter θ, given the data:

$$p(\theta|D_t) = \frac{p(\theta)P(D_t|\theta)}{\int d\theta' p(\theta')P(D_t|\theta')}. \tag{5.1}$$

The posterior density (5.1) can be used to calculate an estimate for the unknown parameter. The simplest case is the so called MAP (maximum a posteriori) value $\hat{\theta} = \arg\max \ln p(\theta|D_t)$, i.e. the most probable parameter value. Another choice is the posterior mean of the parameter. Using the full posterior, it is possible to go beyond a simple parameter estimation and to define a *Bayes optimal prediction* for the unknown probability distribution. Optimality is here understood in an average sense, both over random drawings of the data and random drawings of true parameters θ according to $p(\theta)$. It is not hard to show that the optimal distribution $\hat{P}(y|D_t)$ which minimizes the expected quadratic deviation (the symbol $E_{y|\theta}$ stands for expectation with respect to $P(y|\theta)$)

$$\int d\theta \; p(\theta) E_{D_t} E_{y|\theta} \, [\hat{P}(y|D) - P(y|\theta)]^2.$$

is given by a mixture of all possible distributions in the considered family, weighted by their posterior probabilities

$$\hat{P}(y|D) = \int d\theta \; P(y|\theta) \; p(\theta|D). \tag{5.2}$$

This so called *predictive distribution* also minimizes a second important functional, the averaged relative entropy

$$\int d\theta \; p(\theta) E_D E_{y|\theta}[\log \frac{P(y|\theta)}{\hat{P}(y|D)}] \tag{5.3}$$

where

$$E_{y|\theta}\left[\log \frac{P(y|\theta)}{\hat{P}(y|D_t)}\right] = \int dy P(y|\theta) \log \frac{P(y|\theta)}{\hat{P}(y|D_t)}$$

is an important dissimilarity measure between distributions, the *relative entropy* or *Kullback-Leibler divergence* $D_{KL}(P(\cdot|\theta)||\hat{P})$.

From the optimality on average, we see immediately that *no* estimator can beat a Bayes procedure for *all* true parameters θ^*. An estimator that would be uniformly better than a Bayes procedure would also be better on average. Moreover, it can be shown that for special choices of $p(\theta)$, Bayes procedures can also be minimax.

A prior $p(\theta)$ which is well adapted to a problem may act as a regularizer, which can prevent an algorithm from overfitting when the number of examples is small. Some regularization methods for neural networks, e.g. weight decay, can be interpreted in a Bayesian way. On the other hand, when the number of examples is large, the influence of the prior distribution becomes weak. The posterior is sharply peaked at its maximum $\hat{\theta}$ and a Gaussian approximation for its shape

$$p(\theta|D_t) \simeq \exp[-\frac{t}{2} \sum_{ij} (\theta_i - \hat{\theta}_i) \hat{J}_{ij} (\theta_j - \hat{\theta}_j)] \tag{5.4}$$

becomes asymptotically exact. Here $\hat{J}_{ij} = -\partial_i \partial_j \frac{1}{t} \sum_{\mu=1}^{t} \ln P(y_\mu|\hat{\theta})$. In nice parametric cases, consistency and asymptotic efficiency of Bayes predictors can be proved.

6 Online Update of the Posterior

In order to construct an online algorithm within the Bayesian frameweork, we have to find out how the posterior distribution changes when a new datapoint y_{t+1} is observed. It can be easily shown that the new posterior corresponding to the new dataset D_{t+1} is given in terms of the old posterior and the likelihood of the new example by

$$p(\theta|D_{t+1}) = \frac{P(y_{t+1}|\theta) p(\theta|D_t)}{\int d\theta P(y_{t+1}|\theta) p(\theta|D_t)}. \tag{6.1}$$

(6.1) does not have the form of an online algorithm, because it requires the knowledge of the *entire old dataset* D_t. The basic idea to turn this into an online algorithm is to replace the true posterior $p(\theta|D)$ by a simpler parametric distribution $p(\theta|par)$, where *par* is a small set of parameters, which is

able to capture a major part of the information about the previous data and which has to be updated at each step. Hence, the Bayes online algorithm will be based on a repetition of two basic steps:

- Update: Use the old approximative posterior $p(\theta|par(t))$ to perform an update of the form (6.1)

$$p(\theta|y_{t+1}, par(t)) = \frac{P(y_{t+1}|\theta)p(\theta|par(t))}{\int d\theta P(y_{t+1}|\theta)p(\theta|par(t))}. \tag{6.2}$$

- Project: The new posterior $p(\theta|y_{t+1}, par(t))$ will usually not belong to the parametric family $p(\theta|par)$. Hence, in the next step, it need to be projected into this family in order to obtain $p(\theta|par(t+1))$. The parameter $par(t+1)$ must be chosen such that $p(\theta|par(t+1))$ is as close as possible to $p(\theta|y_{t+1}, par(t))$. It is a not clear a priori, which measure of dissimilarity between distributions should be used. Different choices may lead to different algorithms. I have chosen the KL-divergence

$$D_{KL}\left(p(\cdot|y_{t+1}, par(t))||p(\cdot|par)\right) = \tag{6.3}$$
$$\int d\theta\ p(\theta|y_{t+1}, par(t)) \ln \frac{p(\theta|y_{t+1}, par(t))}{p(\theta|par)},$$

  which is nonsymmetric in its arguments. Minimizing (6.3) can be thought of as minimizing the loss of information in the projection step. For the important case, where the parametric family is an exponential family, i.e. if the densities are of the form

$$p(\theta|par) \propto \exp[-\sum_k \alpha_k f_k(\theta)], \tag{6.4}$$

  it is easy to see, that minimizing (6.3) is equivalent to adjusting the parameters α_k such that the moments $E_\theta f_k(\theta)$ match for both distributions $p(\theta|par)$ and $p(\theta|y_{t+1}, par(t))$. This is also equivalent to finding the distribution $p(\theta|par)$ which maximizes the entropy under the constraints that these moments are given. Two cases of exponential families have sofar been studied for Bayes online learning: The case of a Gaussian family of distributions for learning of continuous parameters was discussed in (Opper 1996). A family of product distributions for binary random variables was chosen by (Winther & Solla 1997) for learning in the Ising perceptron. In the next section I will discuss the Gaussian case.

7 Gaussian Ansatz

If we use a general multivariate Gaussian distribution for $p(\theta|par)$, then $par = (mean, covariance) = (\hat{\theta}_i, C_{ij})$. Matching the moments results in

$$\begin{aligned}\hat{\theta}_i(t+1) &= \frac{\int d\theta\ \theta_i P(y_{t+1}|\theta)p(\theta|par(t))}{\int d\theta P(y_{t+1}|\theta)p(\theta|par(t))} \\ C_{ij}(t+1) &= \frac{\int d\theta\ \theta_i\ \theta_j P(y_{t+1}|\theta)p(\theta|par(t))}{\int d\theta P(y_{t+1}|\theta)p(\theta|par(t))} - \hat{\theta}_i(t+1)\hat{\theta}_j(t+1).\end{aligned}$$

Using a simple property of centered Gaussian random variables z, namely the fact that for well behaved functions f, we have $E(zf(z)) = E(f'(z))\cdot E(z^2)$, we can get the explicit update:

$$\begin{aligned}\hat{\theta}(t+1) &= \hat{\theta}(t) + \sum_j C_{ij}(t) \times \qquad (7.1)\\ &\quad \times\partial_j \ln E_u[P(y_{t+1}|\hat{\theta}(t)+u)] \\ &\text{and} \\ C_{ij}(t+1) &= C_{ij}(t) + \sum_{kl} C_{ik}(t)C_{lj}(t) \times \qquad (7.2)\\ &\quad \times\partial_k\partial_l \ln E_u[P(y_{t+1}|\hat{\theta}(t)+u)].\end{aligned}$$

Here the expectation $\int d\theta P(y_{t+1}|\theta)p(\theta|par(t))$ is written as $E_u[P(y_{t+1}|\hat{\theta}(t)+u)]$ where u is a zero mean Gaussian random vector with covariance $C(t)$.

It is interesting that the Bayesian approach combined with the Gaussian approximation to the posterior has led to an update for the posterior mean (which for this approximation equals the MAP value) which looks like a gradient descent with a tensorial learning rate. This learning rate need not to be determined from the outside by some given schedule. In the Bayes approach it is automatically adjusted by the data!

For smooth models, the exact Gaussian asymptotics (5.4) of the posterior suggests that the approximation should not be bad when the number of examples grows large. In the next section, we will see that this is actually the case.

8 Asymptotic Performance

In order to study the large time behaviour of the algorithm (7.1), (7.2), we first need the asmptotic form of the covariance matrix C. We define $V_{kl} \doteq \partial_k\partial_l \ln E_u P(y_{t+1}|\theta+u)$ and assume that for large times, the

temporal changes of the matrix C are small, so that we can introduce continuous times and replace (7.2) by the matrix differential equation

$$\frac{dC}{dt} = CVC \tag{8.1}$$

which is solved by

$$\frac{dC^{-1}}{dt} = -V.$$

Integrating yields

$$C^{-1}(t) - C^{-1}(t_0) = -\int_{t_0}^{t} V(t')\, dt'. \tag{8.2}$$

To proceed, we will make the assumption that the data are generated independently at random from a distribution $Q(y)$, which we allow to be also outside of the family $P(y|\theta)$, in order to treat the case of a misspecified model. We now assume that the online dynamics is close to an attractive fixed point θ^*, which corresponds to a local minimum of $-\int dy\, Q(y) \ln P(y|\theta)$ and satisfies

$$\int dy\, Q(y) \partial_i \ln P(y|\theta^*) = 0. \tag{8.3}$$

Dividing (8.2) by t and taking the limit $t \to \infty$, we get

$$\begin{aligned} \lim_{t\to\infty} \frac{(C^{-1}(t))_{ij}}{t} &= \lim_{t\to\infty} \frac{-\int_{t_0}^{t} \partial_i \partial_j \ln P(y|\theta^*)}{t} \\ &= -\int dy\, Q(y) \partial_i \partial_j \ln P(y|\theta^*). \end{aligned} \tag{8.4}$$

In the first equality, we have neglected the width the posterior for large times t. In the second equality, the time average has been replaced by the average over $Q(y)$. For this step, it is not necessary to assume that the data are generated independently, ergodicity is sufficient. It is easy to see that for the case $Q(y) = P(y|\theta^*)$, we have

$$\lim_{t\to\infty} \frac{C^{-1}(t)}{t} = J(\theta^*), \tag{8.5}$$

which is the Fisher Information matrix. This result should be compared with the natural gradient (4.4). It shows that asymptotically, the tensorial learning rate obtained from the Bayes online algorithm becomes proportional to the natural gradient if the probabilistic model is correctly specified and if the local fixpoint θ^* is the true parameter.

In order to calculate the asymptotic scaling of the estimation error, defined as the deviation between θ^* and the MAP $\hat{\theta}(t)$, we again assume that the MAP estimates are close to θ^* and the posterior is sharply

peaked around $\hat{\theta}$. We can then neglect the average over the posterior in (7.1) and linearize. Setting $\hat{\theta}_i(t) = \theta_i^* + \epsilon_i(t)$, we get the linear system

$$\Delta\epsilon_i(t) = \sum_l C_{il}\partial_l \ln P + \sum_{kl} C_{il}\epsilon_k(t)\partial_k\partial_l \ln P, \tag{8.6}$$

where $P \equiv P(y_{t+1}|\theta^*)$. We will introduce the matrices

$$\begin{aligned} B_{ij} &= \int dy\, Q(y)\partial_i \ln P(y|\theta^*)\partial_j \ln P(y|\theta^*) \\ A_{ij} &= -\int dy\, Q(y)\partial_i\partial_j \ln P(y|\theta^*). \end{aligned} \tag{8.7}$$

Taking the expectation (denoted by an overbar) over the distribution of the most recent example y_{t+1} and using (8.4) and (8.7), yields an equation of motion for the expected linear error $e_i = \overline{\epsilon_i}$

$$\frac{de_i}{dt} + \frac{e_i}{t} = \sum_j \frac{(A^{-1})_{ij}}{t}\, \overline{\partial_j \ln P} \tag{8.8}$$

valid for $t \to \infty$. Because of the fixed point condition (8.3), the right hand side vanishes and we conclude that the linear error (the bias) decays like $e_i \propto (1/t)$. More interesting is the dynamics of the matrix of quadratic errors $E_{ij} \doteq E_D[\epsilon_i(t)\epsilon_j(t)]$. We multiply equation (8.6) by $\epsilon_j(t)$ and average over the last example. Neglecting terms like $\epsilon_i C_{jk}\partial_k \ln P$, which decay faster than the others, we obtain

$$\begin{aligned} \frac{dE}{dt} &= CBC - CAE - EAC \\ &= \frac{1}{t^2}\, A^{-1}BA^{-1} - \frac{2E}{t}, \end{aligned} \tag{8.9}$$

which is solved by

$$E_D[\epsilon_i(t)\epsilon_j(t)] = \frac{1}{t}\,(A^{-1}BA^{-1})_{ij}, \qquad t \to \infty. \tag{8.10}$$

This is the same rate as the one which was obtained for batch algorithms (Max. Likelihood or Bayes) by (Amari & Murata 1993). For local minima θ^* of $-\int dyQ(y)\ln P(y|\theta)$, the matrix A (and trivially also B) is positive definite and we should always have the optimal $\propto 1/t$ decay of the error! This is in contrast to fixed learning rate schedules, where the prefactor of the learning rate $\eta \propto 1/t$ must be adjusted in order to allow for convergence.

The result (8.10) simplifies further for a wellspecified model. In this case, we can use that $B = A = J(\theta^*)$ such that

$$E_D[\epsilon_i(t)\epsilon_j(t)] = \frac{1}{t}\,(J^{-1}(\theta^*))_{ij}, \qquad t \to \infty.$$

By comparing with (3.3), we see that the Bayes online algorithm becomes asymptotically efficient.

The quadratic estimation error has in general no direct interpretation for the ability of a learning device to predict novel data. One can study a more natural measure for the learning performance which is given by the expected relative entropy distance between the predictive distribution constructed from the approximative posterior and the true data generating distribution.

$$\varepsilon_{entro} = E_{D_t} E_y \left[\log \frac{Q(y)}{\hat{P}(y|D_t)} \right] . \tag{8.11}$$

Using an asymptotic expansion as before, we get

$$\varepsilon_{entro} = E_y \left[\log \frac{Q(y)}{P(y|\theta^*)} \right] + \frac{Tr\,(BA^{-1})}{2t} \tag{8.12}$$

for $t \to \infty$. This result gives the same performance as the one derived for the *batch* maximum likelihood estimate (Seung et al 1992; Amari & Murata 1993). For the well specified case this reduces to the universal asymptotics (Seung et al 1992; Amari & Murata 1993; Opper & Haussler 1995) for Bayes- and maximum likelihood estimators

$$\varepsilon_{entro} = \frac{N}{2t}$$

for $t \to \infty$, which depends only on the number of degrees of freedom.

9 Application

For most models, it will not be easy to perform the Gaussian averages in (7.1) and (7.2) exactly in order to implement the algorithm. Hence, further approximations may be necessary. However, there are a few non-trivial and relevant probabilistic models, where these averages can be performed analytically. The simplest choice is linear models, where for a Gaussian prior also the posterior distribution is a Gaussian and the online approximation becomes exact. A further family of models where we can expect that some of the averages can be performed by hand is the class of mixtures of Gaussians. In the following, we will look at a third case in more detail. This is a model for binary classification which is defined by

$$P(S|\theta, \mathbf{x}) = \phi \left(\frac{S\ \theta \cdot \mathbf{x}}{\sigma_0} \right) , \tag{9.1}$$

where $S = \pm 1$ is a binary class label, θ and $\mathbf{x}$ are N dimensional vectors with inner (dot) product $\theta \cdot \mathbf{x}$ and

$$\phi(z) = \int_{-\infty}^{z} dz\ e^{-t^2/2}/\sqrt{2\pi}$$

is a sigmoidal function. (9.1) may also be related to a perceptron rule with weight noise (Opper & Kinzel 1996). For this case, the Gaussian averages (7.1,7.2) can be carried out explicitely and we obtain

$$E_u P = \phi\left(\frac{S\ \theta \cdot \mathbf{x}}{\sigma(t)}\right) \tag{9.2}$$

with

$$\sigma^2(t) = \sigma_0^2 + \sum_{ij} x_i C_{ij}(t) x_j.$$

Explicit updates (7.1,7.2) for the algorithm can be constructed from

$$\begin{aligned} \partial_j \ln E_u P &= \frac{\phi'}{\phi} S x_j/\sigma(t) \\ \partial_i \partial_j \ln E_u P &= \left\{\frac{\phi''}{\phi} - \left(\frac{\phi'}{\phi}\right)^2\right\} \frac{x_i x_j}{\sigma^2(t)}. \end{aligned}$$

To illustrate the performance of the algorithm, we have studied a one dimensional toy model first. In this model, we assume that scalar inputs x with $-1 \leq x \leq +1$ are classified as $S = \pm 1$ according to whether x is greater or less than θ^*. In addition, Gaussian noise is added to θ^*, hence (9.1) is replaced by $P(S|\theta, \mathbf{x}) = \phi(S(x-\theta)/\sigma_0)$. The expected quadratic estimation error $E_D(\theta_t - \theta^*)^2$ as a function of t is shown in Fig.1 for a true parameter $\theta^* = 0.1$ and $\sigma_0 = 0.5$. The asymptotic approach to efficiency (straight line) can be seen.

Next, we consider the full model (9.1). The simulations (dashed line in Fig.2) are performed with $N = 50$ and the vectors θ^* and $\mathbf{x}$ have independent normally distributed components. The results were averaged over 50 samples. As the initial conditions, we have chosen $\theta = 0$ and the true spherical Gaussian prior. The curves show the $(0-1)$ generalization error

$$\varepsilon = \frac{1}{\pi} \arccos\left(\frac{\theta^* \cdot \hat{\theta}}{||\theta^*||\ ||\hat{\theta}||}\right), \tag{9.3}$$

as a function of $\alpha = \frac{t}{N}$. This quantity measures the probability of disagreement between the classifications of a perceptron defined by the weight vector $\hat{\theta}$ and the noise free perceptron (setting $\sigma_0 = 0$ for independent test data) defined by θ^*. The data were generated with $\sigma_0 = 6.24$. For comparison, we have shown the error for the true (batch)

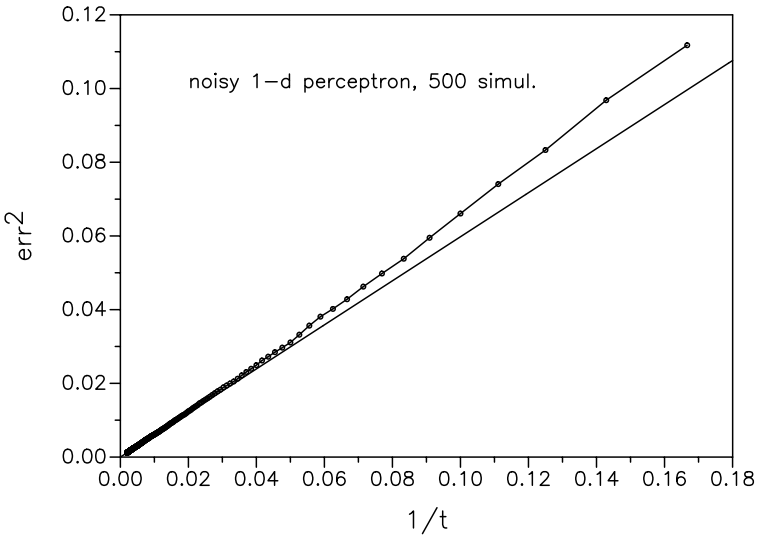

Fig. 1 - Quadratic Error $E_D(\theta - \theta^*)^2$ for a one dimensional noisy perceptron. The straight line gives the bound (3.1).

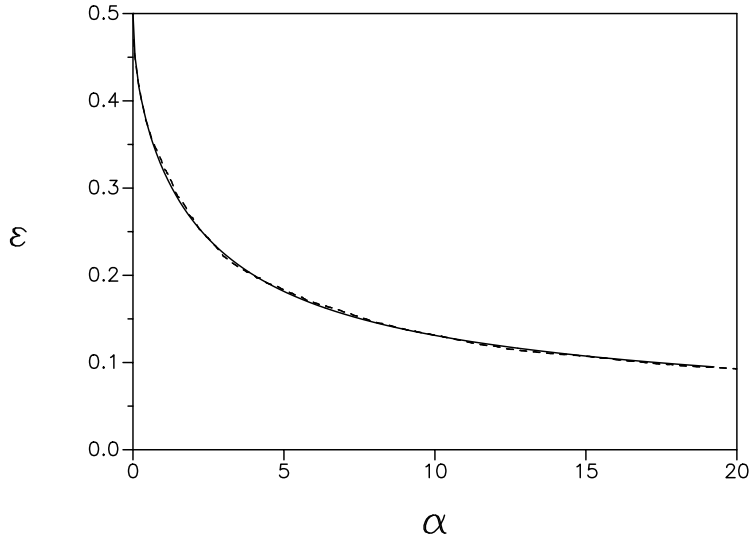

Fig. 2 - Generalization error (9.3) for the classification model (9.1). The dashed line is obtained from the Bayes algorithm. The solid line is an analytical result for the batch error in the thermodynamic limit.

Bayes prediction (solid line) analytically calculated for the thermodynamic limit $N \to \infty$.

The number of parameters to be updated in the online algorithm can be reduced drastically, if the general covariance matrix C is replaced by a diagonal matrix or even simpler, by a single number c. This is equivalent to approximating the posterior by a spherical Gaussian. (7.2)

will simplify to

$$c(t+1) = c(t) + c^2(t)\frac{1}{N}\sum_i \partial_i^2 \ln E_u[P(y_{t+1}|\hat{\theta}(t)+u)]. \qquad (9.4)$$

For the model (9.1) it can be shown that this approximation actually leads to the same update as defined by the locally optimal weighted Hebbian scheme derived for the thermodynamic limit framework (Biehl et al 1995), provided that the model is well specified. This agreement will be lost for misspecified cases.

10 Outlook

One of the greatest challenges for the Bayesian online approach will be the practical realizability for more complicated models like multilayer neural networks. Here, one has to find additional useful approximations in order to perform the Gaussian averages. For example, one may try a stochastic version of the algorithm based on Monte Carlo sampling. Mean field methods may also be helpful. Another possibility is to study distance measures different from (6.3) in the projection step. This may lead to a different algorithm for which averages maybe obtained easier.

From a theoretical viewpoint, better, nonasymptotic estimates of the performance of the Bayes online algorithm are highly desirable. Such results are necessary to understand the global convergence properties. A further interesting question is the performance of the algorithm for nonsmooth models, like the noisefree perceptron. For this case even the asymptotics is unknown.

References

Amari S *Differential-Geometrical Methods in Statistics* Lecture Notes in Statistics, Springer Verlag New York

Amari S 1993 *Neural Networks* **6** 161

Amari S and Murata N 1993 *Neural Computation* **5** 140

Amari S *Neural learning in structured parameter spaces — Natural Riemannian gradient* in *NIPS'96*, vol. 9, MIT Press.

Amari S 1997 (preprint)

Barkai N Seung H S and Sompolinsky H 1995 *Phys. Rev. Lett.* **75** 1415

Berger J O 1985 *Statistical Decision theory and Bayesian Analysis* Springer-Verlag New York

Biehl M and Riegler P 1994 *Europhys. Lett.* **28** 525

Biehl M Riegler P and Stechert M 1995 *Phys. Rev* **E 52** 4624

Copelli M and Caticha N 1995 *J. Phys.* **A 28** 1615

Kinouchi O and Caticha N 1992 *J. Phys.* **A 25** 6243

Kim J W and Sompolinsky H 1996 *Phys. Rev. Lett.* **76** 3021

LeCam L M 1953 *Univ. of California Publications in Statistics* **1** 277

Opper M 1996 *Phys. Rev. Lett.* **77** 4671

Opper M and Haussler D 1995 *Phys. Rev. Lett.* **75** 3772

Opper M and Kinzel W 1996 *Statistical Mechanics of Generalization* in *Physics of Neural Networks*, ed. by J. L. van Hemmen, E. Domany and K. Schulten (Springer Verlag, Berlin)

Saad D and Rattray M 1997 *Phys. Rev. Lett.* **79** 2578

Seung H, Sompolinsky H and Tishby N 1992 *Physical Review* **A 45** 6056

Schervish M J 1995 *Theory of Statistics* Springer -Verlag New York

Van den Broeck C and Reimann P 1996 *Phys. Rev. Lett.* **76** 2188

Vapnik V 1982 *Estimation of dependecies based on emprical data* Springer-Verlag New York

Winther O and Solla S A 1997 *Optimal Bayesian online learning*; Proceedings of the Honkong Int. Workshop on Theor. Aspects of Neural Comp (TANC97)