Decentralized Detection and Classification using Kernel Methods

XuanLong Nguyen XUANLONG@CS.BERKELEY.EDU
Martin J. Wainwright WAINWRIG@EECS.BERKELEY.EDU
Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division and Department of Statistics, U.C. Berkeley, CA 94720-1776 USA

Abstract

We consider the problem of decentralized detection under constraints on the number of bits that can be transmitted by each sensor. In contrast to most previous work, in which the joint distribution of sensor observations is assumed to be known, we address the problem when only a set of empirical samples is available. We propose a novel algorithm using the framework of empirical risk minimization and marginalized kernels, and analyze its computational and statistical properties both theoretically and empirically. We provide an efficient implementation of the algorithm, and demonstrate its performance on both simulated and real data sets.

1. Introduction

Most of the machine learning literature on classification is abstracted away from considerations of an underlying communication-theoretic infrastructure, constraints from which may prevent an algorithm from aggregating all relevant data at a central site. In many real-life applications, however, resource limitations make it necessary to transmit only partial descriptions of data. Examples include sensor networks, in which each sensor operates under power or bandwidth constraints, and human decision-making, in which high-level executive decisions must often be based on lower-level summaries. Assessing losses in classification accuracy, and developing methods to mitigate their impact, is essential if machine learning algorithms are to make inroads in such problem domains.

There is a significant literature on decentralized decision-making that formally states the problem and characterizes possible solutions (Tsitsiklis, 1993). It is noteworthy, however, that this literature focuses almost entirely on the problem when the relevant probability distributions are known in advance (Blum et al., 1997). Consider, for example, a classification problem (i.e., a binary hypothesis-testing problem), in which each of a set of S sensors observe a single component of a vector X. Assume that these sensors must make a local decision to convert its observation into the corresponding component of a vector Z, and that a final decision regarding the value of a binary variable Y is to be made on the basis of Z. While most of the extant literature assumes the distribution $P(X, Y)$ is known, it is clearly of interest to consider decentralized decision-making when only samples from this distribution are available.[1]

In this paper, we address this empirically-based decentralized decision-making problem from the perspective of recent developments in the field of kernel methods (Scholkopf & Smola, 2002). As we will show, kernel methods are particularly natural for solving this problem. In particular, a key component of the methodology that we propose involves the notion of a *marginalized kernel*, where the marginalization is induced by the transformation from measured values X to local decisions Z.

The paper is organized as follows. Section 2 provides a formal statement of the decentralized decision-making problem. We show how the problem can be cast as a learning problem in Section 3, and in Section 4, we present a kernel-based algorithm for solving the problem. We also present error bounds for our algorithm in Section 4. In Section 5 we present the results of empirical experiments, and in Section 6 we present our conclusions.

2. Problem statement

The problem of decentralized classification can be succinctly stated as follows. Suppose Y is a discrete-valued random variable, representing a hypothesis about the environment, and that a set of S sensors collect observations. These signals are represented by a S-dimensional random vector $X = (X^1, \ldots, X^S)$, drawn from the conditional distribution $P(X|Y)$. In the decentralized setting, each sensor $t = 1 \ldots, S$ transmits a message $Z^t = \gamma^t(X^t)$ to the fusion center, which in turn applies some decision rule γ to compute $\widehat{Y} = \gamma(Z^1, \ldots, Z^S)$. Suppose that

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

[1]There is also a relationship to discretization algorithms for classification, which we discuss in Section 5.

each X^t is discrete-valued, taking one of M possible values. The key constraint, giving rise to the decentralized nature of the problem, is that the messages Z^t may take on only L possible values where $L \ll M$. The problem is to find the decision rules $\gamma^1, \ldots, \gamma^S$ for each sensor, and a rule γ for the fusion center so as to minimize the Bayes risk $P(Y \neq \gamma(Z))$. The joint distribution $P(X, Y)$ is unknown, and we are given n i.i.d. data samples $(x_i, y_i)_{i=1}^n$.

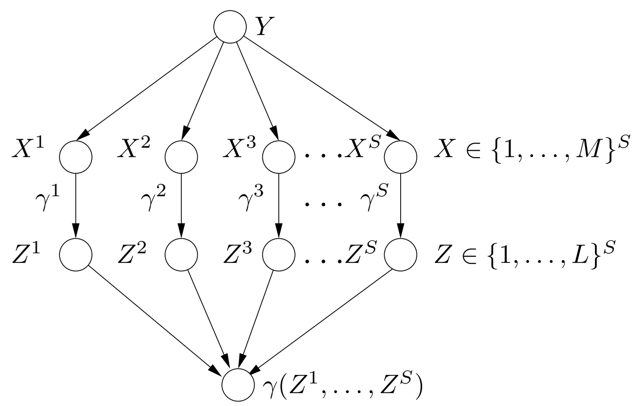


Figure 1. Decentralized detection system with $S = 4$, in which Y is the unknown hypothesis; $X = (X^1, \ldots, X^S)$ is the vector of sensor observations; and $Z = (Z^1, \ldots, Z^S)$ are the quantized messages.

3. Minimizing the empirical risk

Although the Bayes-optimal risk can always be achieved by a deterministic decision rule (Tsitsiklis, 1993), considering the larger space of stochastic decision rules confers some important advantages. First, such a space can be compactly represented and parameterized, and prior knowledge can be incorporated. Second, the optimal deterministic rules are often very hard to compute, and a probabilistic rule may provide a reasonable approximation in practice. Accordingly, we represent the rule for the sensors $t = 1, \ldots, S$ by a conditional probability distribution $Q(Z|X)$. The fusion center makes its decision by computing a deterministic function $\gamma(z)$ of z. The overall decision rule (Q, γ) consists of the individual sensor rules and the fusion center rule.

The decentralization requirement for our detection/classification system—i.e., that the decision rule for sensor t must be a function only of the observation x^t—can be translated into the probabilistic statement that $Z^1, \ldots, Z^S$ be conditionally independent given X:

$$Q(Z|X) = \prod_{t=1}^{S} Q^t(Z^t|X^t). \qquad (1)$$

In fact, this constraint turns out to be advantageous from a computational perspective, as will be clarified in the sequel. We use $\mathcal{Q}$ to denote the space of all factorized conditional distributions $Q(Z|X)$, and $\mathcal{Q}_0$ to denote the subset of factorized conditional distributions that are also deterministic.

Let X denote the signal vector $(X^1, \ldots, X^S)$, and suppose that we have as our training data n pairs (x_i, y_i) for $i = 1, \ldots, n$. Note that x_i is an S-dimensional signal vector, $x_i = (x_i^1, \ldots, x_i^S)$. Let P be the unknown underlying probability distribution for (X, Y). The probabilistic set-up makes it simple to estimate the optimal Bayes risk, which is to be minimized. Although our framework can be applied to general multi-class classification and regression problems, in this paper we focus on binary classification, that is, $Y = \pm 1$.

For each collection of decision rules made at the sensors and represented by $Q(Z|X)$, the optimal Bayes risk is defined by:

$$R_{opt} = \frac{1}{2} - \frac{1}{2}\mathbb{E}\Big|P(Y = 1|Z) - P(Y = -1|Z)\Big|.$$

Here the expectation is with respect to the probability distribution $P(X, Y, Z) := P(X, Y)Q(Z|X)$. It is well known that no decision function made at the fusion center (which is a function of z) has Bayes risk smaller than R_{opt}. In addition, the Bayes risk R_{opt} can be achieved by using the decision function

$$\gamma_{opt}(z) = \text{sign}(P(Y = 1|z) - P(Y = -1|z)).$$

Of course, this decision rule cannot be computed, because $P(X, Y)$ is not known, and $Q(Z|X)$ is to be determined. Thus, our goal is to determine the rule $Q(Z|X)$ that minimizes an empirical estimate of the Bayes risk based on the training data $(x_i, y_i)_{i=1}^n$. In Lemma 1, we show that the following is one such unbiased estimate of the Bayes risk:

$$R_{emp} = \frac{1}{2} - \frac{1}{2n}\sum_{z}\Big|\sum_{i=1}^{n} Q(z|x_i)y_i\Big|. \qquad (2)$$

In addition, $\gamma_{opt}(z)$ can be estimated by the decision function $\gamma_{emp}(z) = \text{sign}\big(\sum_{i=1}^{n} Q(z|x_i)y_i\big)$. Since Z is a discrete random vector, the optimal Bayes risk can be estimated easily, regardless of whether the input signal X is discrete or continuous.

Lemma 1. *(a) Assume that $P(z) > 0$ for all z. Define*

$$\kappa(z) = \frac{\sum_{i=1}^{n} Q(z|x_i)\mathbb{I}(y_i = 1)}{\sum_{i=1}^{n} Q(z|x_i)}.$$

Then $\lim_{n\to\infty} \kappa(z) = P(Y = 1|z)$.

(b) As $n \to \infty$, R_{emp} and $\gamma_{emp}(z)$ tend to R_{opt} and $\gamma_{opt}(z)$, respectively.

This lemma[2] motivates the goal of finding a rule $Q(Z|X)$ that minimizes R_{emp}. It is equivalent, using eqn. (2), to maximize

$$C(Q) = \sum_z \Big| \sum_{i=1}^n Q(z|x_i) y_i \Big|, \tag{3}$$

subject to the natural constraints on a probability distribution (i.e., $Q(z|x) = \prod_{t=1}^S Q^t(z^t|x^t)$; $\sum_{z^t} Q^t(z^t|x^t) = 1$; and $Q^t(z^t|x^t) \in [0,1]$). The major computational difficulty in this optimization problem lies in the summation over all L^S possible values of z. One way to avoid this obstacle is by maximizing instead the following function:

$$\begin{aligned} C_2(Q) &= \sum_z \Big(\sum_{i=1}^n Q(z|x_i) y_i \Big)^2 \\ &= \sum_{i,j} y_i y_j \prod_{t=1}^S \sum_{z^t=1}^L Q^t(z^t|x_i^t) Q^t(z^t|x_j^t), \end{aligned}$$

where the final line follows after expanding the square and summing. Note that the constraints (1) on Q allow us to compute $C_2(Q)$ in $O(SL)$ time, as opposed to $O(L^S)$.

While this simple strategy is based directly on the empirical risk, it does not exploit any prior knowledge about the class of discriminant functions for $\gamma(z)$. As we discuss in the following section, such knowledge can be incorporated into the classifier using kernel methods. Moreover, the kernel-based decentralized detection algorithm that we develop turns out to have an interesting connection to the simple approach based on $C_2(Q)$.

4. A kernel-based algorithm

In this section, we shall apply Mercer kernels (Scholkopf & Smola, 2002) to our decentralized classification problem, focusing on the case of binary labels $Y = \pm 1$. Kernel-based methods for discrimination entail choosing a discriminant function f from within a function class $\mathcal{F}$ defined by a *feature space* $\{\Phi(x)\}$. This space is equipped with an inner product $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$, which defines the *kernel function* K. As a reproducing kernel Hilbert space, any function $f \in \mathcal{F}$ expressed as an inner product $f(x) = \langle w, \Phi(x) \rangle$, where w has the form $w = \sum_{i=1}^n \alpha_i \Phi(x_i)$. Equivalently, we can write f as a linear combination of kernel functions as follows:

$$f(x) = \sum_{i=1}^n \alpha_i K_x(x_i, x) \tag{4}$$

In the framework of empirical risk minimization, the parameters α_i are chosen so as to minimize a cost function given by the sum of the empirical ϕ-risk $\hat{\mathbb{E}}\phi(Yf(X))$ with a suitable regularization term (e.g., ℓ_2-regularization), where ϕ denotes an appropriate loss function. The function ϕ is typically a convex surrogate for the 0-1 loss. The final decision rule is then given by $\widehat{y} := \mathrm{sign}(f(x))$. It has been shown (Zhang, 2004; Bartlett et al., 2003) that a function f with small ϕ-risk $\mathbb{E}\phi(Yf(X))$ also has small Bayes risk $P(Y \neq \mathrm{sign}(f(X)))$.

4.1. Fusion center and marginalized kernels

With this background, we first consider how to design the decision rule γ at the fusion center for a *fixed* setting $Q(Z|X)$ of the sensor decision rules. Since the fusion center rule can only depend on $z = (z^1, \ldots, z^S)$, our starting point is a feature space $\{\Phi'(z)\}$ with associated kernel K_z. We consider fusion center rules defined by taking the sign of linear discriminants $\gamma(z) := \langle w, \Phi'(z) \rangle$. We then link the performance of γ to another kernel-based discriminant function f that acts *directly* on $x = (x^1, \ldots, x^S)$, where the associated kernel K_Q is defined as a *marginalized kernel* in terms of $Q(Z|X)$ and K_z.

The relevant optimization problem is to minimize (as a function of w) the following regularized form of the empirical ϕ-risk associated with the discriminant γ

$$\min_w \sum_z \sum_{i=1}^n \phi(y_i \gamma(z)) Q(z|x_i) + \frac{\lambda}{2} ||w||^2, \tag{5}$$

where $\lambda > 0$ is a regularization parameter. In its current form, the objective function (5) is intractable to compute (because it involves summing over all L^S possible values of z of a loss function that is generally non-decomposable). However, exploiting the convexity of ϕ allows us to compute it exactly for deterministic rules in $\mathcal{Q}_0$, and also leads to a natural relaxation for an arbitrary decision rule $Q \in \mathcal{Q}$, as formalized in the following:

Proposition 2. *Define the quantities*

$$\Phi_Q(x) = \sum_z Q(z|x) \Phi'(z), \quad f(x; Q) = \langle w, \Phi_Q(x) \rangle. \tag{6}$$

For any convex ϕ, the optimal value of the following optimization problem is a lower bound on the optimal value in problem (5)*:*

$$\min_w \sum_i \phi(y_i f(x_i; Q)) + \frac{\lambda}{2} ||w||^2 \tag{7}$$

Moreover, the relaxation is tight for any deterministic rule $Q(Z|X)$.

Proof. Applying Jensen's inequality to the function ϕ yields $\phi(y_i f(x_i; Q)) \leq \sum_z \phi(y_i \gamma(z)) Q(z|x_i)$ for each $i = 1, \ldots n$, from which the lower bound follows. Equality for deterministic $Q \in \mathcal{Q}_0$ is immediate. $\square$

[2] Proofs of this and other technical results are presented in the long version of the paper (Nguyen et al., 2004).

A key point is that the modified optimization problem (7) involves an ordinary regularized empirical ϕ-loss, but in terms of a linear discriminant function $f(x;Q) = \langle w, \Phi_Q(x)\rangle$ in the *transformed* feature space $\{\Phi_Q(x)\}$ defined in eqn. (6). Moreover, the corresponding *marginalized kernel* function takes the form:

$$K_Q(x,x') := \sum_{z,z'} Q(z|x)Q(z'|x')\; K_z(z,z'), \qquad (8)$$

where $K_z(z,z') := \langle \Phi'(z), \Phi'(z')\rangle$ is the kernel in $\{\Phi'(z)\}$-space. From a computational point of view, we have converted the marginalization over loss function values to a marginalization over kernel functions. While the former is intractable, the latter marginalization can be carried out in many cases (see Section 4.2) by exploiting the structure of the conditional distributions $Q(Z|X)$. From the modeling perspective, it is interesting to note that the class of marginalized kernels, exemplified by eqn. (8), underlie much recent work that aims to combine the advantages of graphical models and kernel methods (Jaakkola & Haussler, 1999; Tsuda et al., 2002).

As a standard kernel-based formulation, the optimization problem (7) can be solved by the usual Lagrangian dual formulation (Scholkopf & Smola, 2002), thereby yielding an optimal weight vector w. This weight vector defines the decision rule for the fusion center by $\gamma(z) := \langle w, \Phi'(z)\rangle$. By the Representer Theorem, the optimal solution w to problem (7) has an expansion of the form

$$w = \sum_{i=1}^{n} \alpha_i y_i \Phi_Q(x_i) \;=\; \sum_{i=1}^{n}\sum_{z'} \alpha_i y_i Q(z'|x_i)\Phi'(z'),$$

where α is an optimal dual solution, and the second equality follows from the definition of $\Phi_Q(x)$ given in eqn. (6). Substituting this decomposition of w into the definition of γ yields

$$\gamma(z) := \sum_{z'}\sum_{i=1}^{n} \alpha_i y_i Q(z'|x_i) K_z(z,z'). \qquad (9)$$

Note that there is an intuitive connection between the discriminant functions f and γ. In particular, using the definitions of f and K_Q, it can be seen that $f(x) = \mathbb{E}[\gamma(Z)|x]$, where the expectation is taken with respect to $Q(Z|X=x)$. The interpretation is quite natural: when conditioned on some x, the average behavior of the discriminant function $\gamma(Z)$, which does *not* observe x, is equivalent to the optimal discriminant $f(x)$, which does have access to x.

4.2. Computation of marginalized kernels

When $Q(Z|X)$ is not deterministic, the computation of $K_Q(x,x')$ entails marginalizing over Z, resulting generally in $O(L^S)$ computational cost. In such a case, we have to resort to an approximation of K. However, when the kernel function $K_z(z,z')$ is decomposed into local functions, the computation becomes feasible. Here we provide a few examples of computationally tractable kernels.

Perhaps the simplest example is the *linear kernel* $K_z(z,z') = \sum_{t=1}^{S} z^t z'^t$, for which it is straightforward to derive $K_Q(x,x') = \sum_{t=l}^{S} \mathbb{E}[z^t|x^t]\; \mathbb{E}[z'^t|x'^t]$. A second example, natural for applications in which X^t and Z^t are multinomial, is the *count kernel*. Each multinomial value u is represented as a vector $(0,\ldots,1,\ldots,0)$, whose u-th coordinate takes value 1. If we define the first-order count kernel $K_z(z,z') := \sum_{t=1}^{S} \mathbb{I}[z^t = z'^t]$, then the resulting marginalized kernel takes the form

$$K_Q(x,x') \;=\; \sum_{t=1}^{S} P(z^t = z'^t|x^t, x'^t).$$

A natural generalization is the *second-order count kernel* $K_z(z,z') = \sum_{t,r=1}^{s} \mathbb{I}[z^t = z'^t]\mathbb{I}[z^r = z'^r]$ that accounts for the pairwise interaction between coordinates z^t and z^r. For this example, the associated marginalized kernel $K_Q(x,x')$ takes the form:

$$2\sum_{1\le t<r\le S} P(z^t = z'^t|x^t, x'^t)P(z^r = z'^r|x^r, x'^r). \quad (10)$$

Remarks: First, note that even for a linear base kernel K_z, the kernel function K_Q inherits additional (non-linear) structure from the marginalization over $Q(Z|X)$. As a consequence, the associated discriminant functions (i.e., γ and f) are certainly not linear. Second, our formulation allows any available prior knowledge to be incorporated into K_Q in at least two possible ways: (i) The base kernel representing a similarity measure in the quantized space of z can reflect the structure of the sensor network, or (ii) More structured decision rules $Q(Z|X)$ can be considered, such as chain or tree-structured decision rules.

4.3. Joint optimization

Our next task is to perform joint optimization of both the fusion center rule, defined by w or equivalently α (as in eqn. (9)), and the sensor rules Q. Observe that the cost function (7) can be re-expressed as a function of both w and Q as follows:

$$G(w;Q) \;:=\; \frac{1}{\lambda}\sum_i \phi\Big(y_i\langle w, \sum_z Q(z|x_i)\Phi'(z)\rangle\Big) + \frac{1}{2}||w||^2 \qquad (11)$$

Of interest is the joint minimization of the function G in both w and Q. It can be seen easily that (a) G is convex in w with Q fixed; and (b) moreover, G is convex in Q^t, when both w and all other $\{Q^r, r\neq t\}$ are fixed. These observations motivate the use of blockwise coordinate descent to perform the joint minimization.

Optimization of w**:** As described in Section 4.1, when Q is fixed, then $\min_w G(w; Q)$ can be computed efficiently by a standard dual reformulation. Specifically, using standard results from convex duality (Rockafellar, 1970), we can show that a dual reformulation of $\min_w G(w; Q)$ is given by

$$\max_{\alpha \in \mathbb{R}^n} \Big\{ -\frac{1}{\lambda} \sum_{i=1}^n \phi^*(-\lambda\alpha_i) - \frac{1}{2}\alpha^T \big[(yy^T) \circ K_Q\big]\alpha \Big\}, \quad (12)$$

where $\phi^*(u) := \sup_{v \in \mathbb{R}} \{u \cdot v - \phi(v)\}$ is the conjugate dual of ϕ; $[K_Q]_{ij} := K_Q(x_i, x_j)$ is the empirical kernel matrix; and $\circ$ denotes Hadamard product (Nguyen et al., 2004). Any optimal solution α to problem (12) defines the optimal primal solution $w(Q)$ to $\min_w G(w; Q)$ via $w(Q) = \sum_{i=1}^n \alpha_i y_i \Phi_Q(x_i)$.

As a particular example, consider the case of hinge loss function $\phi(u) := (1 - u)_+$, as used in the SVM algorithm (Scholkopf & Smola, 2002). A straightforward calculation yields

$$\phi^*(u) = \begin{cases} u & \text{if } u \in [-1, 0] \\ +\infty & \text{otherwise.} \end{cases}$$

Substituting this formula into (12) and yields the familiar dual formulation for the SVM:

$$\max_{0 \le \alpha \le 1/\lambda} \Big\{ \sum_i^n \alpha_i - \frac{1}{2}\alpha^T \big[(yy^T) \circ K_Q\big]\alpha \Big\}.$$

Optimization of Q**:** The second step is to minimize G over Q^t, with w and all other $\{Q^r, r \neq t\}$ held fixed. Our approach is to compute the derivative (or more generally, the subdifferential) with respect to Q^t, and then apply a gradient-based method. A challenge to be confronted is that G is defined in terms of feature vectors $\Phi'(z)$, which are typically high-dimensional quantities. Indeed, although it is intractable to evaluate the gradient at an arbitrary w, the following result establishes that it can always be evaluated at the point $(w(Q), Q)$ for any $Q \in \mathcal{Q}$.

Lemma 3. *Let $w(Q)$ be the optimizing argument of $\min_w G(w; Q)$, and let α be an optimal solution to the dual problem (12). Then the following element*

$$-\lambda \sum_{(i,j)(z,z')} \alpha_i \alpha_j Q(z'|x_j) \frac{Q(z|x_i)}{Q(z^t|x_i^t)} K_z(z, z') \mathbb{I}[x_i^t = \bar{x}^t]\; \mathbb{I}[z^t = \bar{z}^t]$$

is an element of the subdifferential[3] $\partial_{Q^t(\bar{z}^t|\bar{x}^t)} G$ *evaluated at* $(w(Q), Q)$.

Observe that this representation of the (sub)gradient involves marginalization over Q of the kernel function K_z, and therefore can be computed efficiently in many cases, as described in Section 4.2.

Overall, the blockwise coordinate descent algorithm takes the following form:

(a) With Q fixed, compute the optimizing $w(Q)$ by solving the dual problem (12).

(b) For some index t, fix $w(Q)$ and $\{Q^r, r \neq t\}$ and take a gradient step in Q^t using Lemma 3.

Remarks: It is interesting to note that if we fix w such that all α_i are equal to 1, and the base kernel K_z is a constant—and thus uninformative—kernel, then the optimization of G with respect to Q reduces to the optimization problem underlying the simple algorithm in Section 3.

4.4. Estimation error bounds

We now turn to the analysis of the statistical properties of our algorithm. In particular, we relate bounds on the *ϕ-risk* $\mathbb{E}\phi(Y\gamma(Z))$ to the ϕ-risk $\mathbb{E}\phi(Yf(X)$ for functions $f \in \mathcal{F}$ (and $f \in \mathcal{F}_0$) that are computed by our algorithm. The latter quantities are well-studied objects in statistical learning theory. In general, the ϕ-risk for a function f in some class $\mathcal{F}$ is bounded by the empirical ϕ-risk plus a complexity term that captures the richness of $\mathcal{F}$.

We first need to isolate the class of functions over which we optimize. Define, for a fixed $Q \in \mathcal{Q}$, the function space $\mathcal{F}_Q$ as

$$\big\{x \mapsto \langle w, \Phi_Q(x)\rangle = \sum_i \alpha_i y_i K_Q(x, x_i) \;\big|\; \text{s. t. } ||w|| \le B\big\}.$$

Note that $\mathcal{F}_Q$ is simply the class of functions associated with the marginalized kernel K_Q. We then define $\mathcal{F} = \cup_{Q \in \mathcal{Q}} \mathcal{F}_Q$, which corresponds to the function class over which our algorithm optimizes. Finally, we let $\mathcal{F}_0$ denote $\cup_{Q \in \mathcal{Q}_0} \mathcal{F}_Q$, corresponding to the union of the function spaces defined by marginalized kernels with deterministic Q. Of particular interest in the current context is the growth in the complexity of $\mathcal{F}$ and $\mathcal{F}_0$ with respect to the number of training samples n, as well as the number of quantization levels L and M.

Any discriminant function f, defined by a vector α, induces an associated discriminant function γ_f via eqn. (9). Relevant to the performance of the classifier γ_f is the expected ϕ-loss $\mathbb{E}\phi(Y\gamma_f(Z))$ (or its empirical version), whereas the algorithm actually minimizes (the empirical version of) $\mathbb{E}\phi(Yf(X))$. The relationship between these two quantities is expressed in the following proposition.

Proposition 4.
(a) We have $\mathbb{E}\phi(Y\gamma_f(Z)) \ge \mathbb{E}\phi(Yf(X))$, with equality when $Q(Z|X)$ is deterministic.

[3] For the case of differentiable ϕ (e.g., logistic loss $\phi(u) := \log[1 + \exp(-u)]$), the subdifferential reduces to a single gradient.

(b) Moreover, there holds

$$\inf_{f\in\mathcal{F}} \mathbb{E}\phi(Y\gamma_f(Z)) \quad \leq \quad \inf_{f\in\mathcal{F}_0} \mathbb{E}\phi(Yf(X) \qquad \text{(13a)}$$
$$\inf_{f\in\mathcal{F}} \mathbb{E}\phi(Y\gamma_f(Z)) \quad \geq \quad \inf_{f\in\mathcal{F}} \mathbb{E}\phi(Yf(X)). \qquad \text{(13b)}$$

The same statement also holds for empirical expectations.

Proof. Applying Jensen's inequality to the convex function ϕ yields

$$\begin{aligned}\mathbb{E}\phi(Y\gamma_f(Z)) &= \mathbb{E}_{XY}\mathbb{E}[\phi(Y\gamma_f(Z))|XY] \\ &\geq \mathbb{E}_{XY}\phi(\mathbb{E}[Y\gamma_f(Z)|XY]) = \mathbb{E}\phi(Yf(X)),\end{aligned}$$

where we have used the conditional independence of Z and Y given X. This establishes part (a), and the lower bound (13b) follows directly. Moreover, part (a) also implies that $\inf_{f\in\mathcal{F}_0}\mathbb{E}\phi(Y\gamma_f(Z)) = \inf_{f\in\mathcal{F}_0}\mathbb{E}\phi(Yf(X))$, and the upper bound (13a) follows since $\mathcal{F}_0 \subset \mathcal{F}$. □

Our next step is to relate the empirical ϕ-risk for f (i.e., $\widehat{\mathbb{E}}(Yf(X))$) to the true ϕ-risk (i.e., $\mathbb{E}(Yf(X))$). Recall that the *Rademacher complexity* of the function class $\mathcal{F}$ is defined (van der Vaart & Wellner, 1996) as

$$R_n(\mathcal{F}) = \mathbb{E}\sup_{f\in\mathcal{F}} \frac{2}{n}\sum_{i=1}^{n}\sigma_i f(X_i),$$

where $\sigma_1,\ldots,\sigma_n$ are independent and uniform on $\{-1,+1\}$, and $X_1,\ldots,X_n$ are i.i.d. samples selected according to distribution P. In the case that ϕ is Lipschitz with constant ℓ, the empirical and true risk can be related via the Rademacher complexity (Koltchinskii & Panchenko, 2002) as follows. With probability at least $1-\delta$ with respect to training samples $(X_i,Y_i)_{i=1}^n$, drawn according to the empirical distribution P^n, there holds

$$\sup_{f\in\mathcal{F}} |\hat{\mathbb{E}}_n\phi(Yf(X)) - \mathbb{E}\phi(Yf(X))| \leq 2\ell R_n(\mathcal{F}) + \sqrt{\frac{\ln(2/\delta)}{2n}}. \qquad (14)$$

Moreover, the same bound applies to $\mathcal{F}_0$.

Combining the bound (14) with Proposition 4 leads to the following theorem, which provides generalization error bounds for the optimal ϕ-risk of the decision function learned by our algorithm in terms of the Rademacher complexities $R_n(\mathcal{F}_0)$ and $R_n(\mathcal{F})$:

Theorem 5. *Given n i.i.d. labeled data samples $(x_i,y_i)_{i=1}^n$, with probability at least $1-2\delta$,*

$$\begin{aligned}\inf_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\phi(y_i f(x_i)) - 2\ell R_n(\mathcal{F}) - \sqrt{\frac{\ln(2/\delta)}{2n}} \\ \leq \inf_{f\in\mathcal{F}}\mathbb{E}\phi(Y\gamma_f(Z)) \leq \\ \inf_{f\in\mathcal{F}_0} \frac{1}{n}\sum_{i=1}^{n}\phi(y_i f(x_i)) + 2\ell R_n(\mathcal{F}_0) + \sqrt{\frac{\ln(2/\delta)}{2n}}.\end{aligned}$$

To make use of this result, we need to derive upper bounds on the Rademacher complexity of the function classes $\mathcal{F}$ and $\mathcal{F}_0$. The following proposition derives such bounds for $\mathcal{F}_0$, exploiting the fact that the number of 0-1 conditional probability distributions $Q(Z|X)$ is a finite number (L^{MS}). While this rate is not tight in terms of the number of data samples n, the bound is nontrivial and is relatively simple (depending directly on the kernel function K and n, L, S, and M).

Proposition 6.

$$\begin{aligned}R_n(\mathcal{F}_0) \leq \frac{2B}{n}\Bigg[\mathbb{E}\sup_{Q\in\mathcal{Q}_0}\sum_{i=1}^{n}K_Q(X_i,X_i) \\ + 2(n-1)\sqrt{n/2}\sup_{z,z'}K_z(z,z')\sqrt{2MS\log L}\Bigg]^{1/2}.\end{aligned}$$

We can also provide a more general and possibly tighter upper bound on the Rademacher complexity based on entropy numbers. In general, define the covering number $N(\epsilon, S, \rho)$ for a set $\mathcal{S}$ to be the minimum number of balls of diameter ϵ that completely cover $\mathcal{S}$ according to a metric ρ. The ϵ-entropy number of S is then $\log N(\epsilon, S, \rho)$. It is well known (van der Vaart & Wellner, 1996) that for some absolute constant C, there holds:

$$R_n(\mathcal{F}) \leq C\int_0^\infty \sqrt{\frac{\log N(\epsilon,\mathcal{F},L_2(P_n))}{n}}\mathrm{d}\epsilon. \qquad (15)$$

Of particular interest is the increase of the entropy number for $\mathcal{F}$ over the supremum of the entropy number for a restricted function class $\mathcal{F}_Q$.

Proposition 7.

$$\begin{aligned}\log N(\epsilon,\mathcal{F},L_2(P_n)) \leq \sup_{Q\in\mathcal{Q}}\log N(\epsilon/2,\mathcal{F}_Q,L_2(P_n)) \\ +(L-1)MS\log\frac{2L^S\sup||\alpha||_1\sup_{z,z'}K_z(z,z')}{\epsilon}.\end{aligned}$$

Moreover, the same bound holds for $\mathcal{F}_0$.

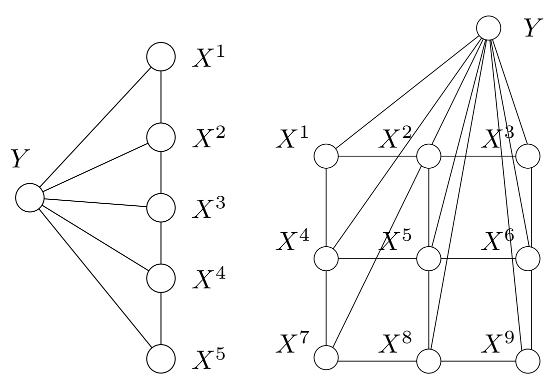
This proposition guarantees that the increase in the entropy number is only $O((L-1)MS\log(L^S/\epsilon))$, which results in only an $O([MS^2(L-1)\log L/n]^{\frac{1}{2}})$ increase in the upper bound (15) for $R_n(\mathcal{F})$ (respectively $R_n(\mathcal{F}_0)$). The Rademacher complexity increases with the square root of $L\log L$ of the number L of quantization levels.

5. Experimental Results

We evaluated our algorithm by testing with both simulated sensor networks and real-world data sets. We consider three types of sensor network configurations:

Naive Bayes networks: In this example, the observations $X^1, \ldots, X^S$ are independent conditional on Y, as illustrated in Figure 1. We consider networks with 10 sensors ($S = 10$), each of which receive signals with 8 levels ($M = 8$). We applied the algorithm to compute decision rules for $L = 2$. In all cases, we generate $n = 200$ training samples, and the same number for testing. We performed 20 trials on 20 randomly generated models $P(X, Y)$.

Chain-structured dependency: While widely used, the conditional independence assumption underlying the naive Bayes set-up is often unrealistic. For instance, consider the problem of detecting a random signal in noise (van Trees, 1990), in which $Y = 1$ represents the hypothesis that a certain random signal is present in the environment, whereas $Y = -1$ represents the hypothesis that only i.i.d. noise is present. Under these assumptions $X^1, \ldots, X^s$ will be conditionally independent given $Y = -1$, since all sensors receive i.i.d. noise. However, conditioned on $Y = +1$ (i.e., in the presence of the random signal), the observations at spatially adjacent sensors will be dependent, with the dependence decaying with distance. In a 1-D setting, this set-up can be modeled with a chain-structured dependency, and the use of a count kernel to account for the interaction among sensors. More precisely, we consider a set-up in which five sensors are located in a line such that only adjacent sensors interact with each other, i.e., X_{t-1} and X_{t+1} are independent given X_t *and* Y (see Figure 2). We implemented KQ with both first- and second-order count kernels. The loss function used is the hinge loss as in the SVM algorithm. The second-order kernel is specified in eqn. (10) but with the sum taken over only t, r such that $|t - r| = 1$.



(a) (b)

Figure 2. Examples of graphical models $P(X, Y)$ of our simulated sensor networks. (a) Chain-structured dependency. (b) Fully connected (not all connections shown).

Spatially-dependent sensors: As a third example, we consider a 2-D layout in which, conditional on the random target being present ($Y = +1$), all sensors interact but with the strength of interaction decaying with distance. Thus $P(X|Y = 1)$ is of the form:

$$\propto \exp\Big\{\sum_t h_{t;u}\mathbb{I}_u(X^t) + \sum_{t \neq r; uv} \theta_{tr;uv}\mathbb{I}_u(X^t)\mathbb{I}_v(X^r)\Big\}.$$

Here the parameter h represents observations at individual sensors, whereas θ controls the dependence among sensors. The distribution $P(X|Y = -1)$ can be modeled in the same way with observations h', and setting $\theta' = 0$ so that the sensors are conditionally independent. In simulations, we generated $\theta_{tr;uv} \sim N(1/d_{tr}, 0.1)$, where d_{tr} is the distance between sensor t and r, and the observations h and h' are randomly chosen in $[0, 1]^S$. We consider a sensor network with 9 nodes (i.e., $S = 9$), arrayed in the 3×3 lattice illustrated in Figure 2(b). Since computation of this density is intractable for moderate-sized networks, we generated an empirical data set (x_i, y_i) by Gibbs sampling.

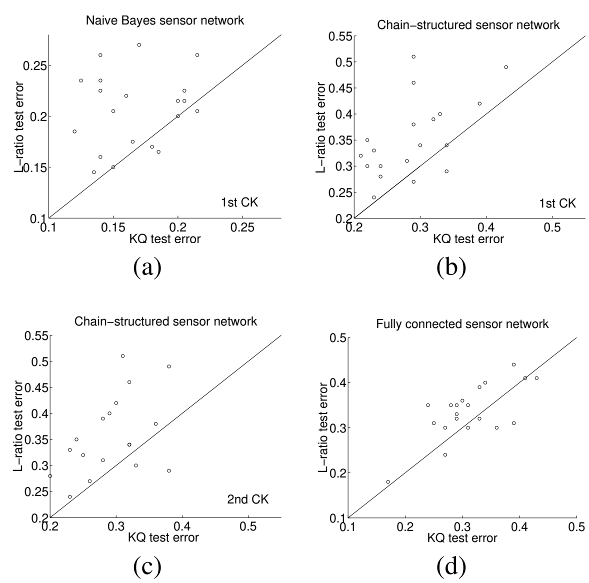


Figure 3. Scatter plots of the test error of the LT versus KQ methods. (a) naive Bayes network. (b) Chain model with 1st-order kernel. (c), (d) Chain model with 2nd order kernel. (d) Fully connected model.

We compare the results of our algorithm to an alternative decentralized classifier based on performing likelihood-ratio (LR) test at each sensor. Specifically, for each sensor t, the estimates $\frac{P(X^t=u|Y=1)}{P(X^t=u|Y=-1)}$ for $u = 1, \ldots, M$ of the likelihood-ratio are sorted and grouped evenly into L bins. Given the quantized input signal and label Y, we then construct a naive Bayes classifier at the fusion center. This choice of decision rule provides a reasonable comparison, since thresholded likelihood ratio tests are optimal in many cases (Tsitsiklis, 1993).

As we show in Figure 3, the kernel-based quantization (KQ) algorithm developed in Section 4 generally yields better classification results than the likelihood-ratio based algorithm. The figure provides scatter plots of LR versus KQ test error for four different set-ups, using $L = 2$ levels of quantization. Panel (a) shows the naive Bayes setting and the KQ method with first-order count kernel. Note that the KQ error is below the LR error for the large majority of examples. Panels (b) and (c) show the case of chain-structured dependency, as illustrated in Figure 2(a), using a first- and second-order count kernel respectively. Again, the KQ error improves significantly on the LR error. Finally, panel (d) shows the fully-connected case of Figure 2(b) with a first-order kernel. The performance of KQ is somewhat better than LR, although by a smaller amount than the other cases.

UCI repository data sets: We also applied our algorithm to several data sets from the UCI repository. In contrast to the sensor network setting, in which communication constraints must be respected, the problem here can be viewed as that of finding a good quantization scheme that retains information about the class label. Thus, the problem is similar in spirit to work on discretization schemes for classification (Dougherty et al., 1995). However, in our case, we assume that the data have already been crudely quantized (to $M = 8$ levels) and we retain no information on the relative magnitudes of the levels, thus rendering classical discretization algorithms inapplicable. The problem is one of hierarchical decision-making, in which a second-level decision follows a first-level set of decisions concerning the features.

Data	$L = 2$	4	6	NB	CK
Pima	0.212	0.217	0.212	0.223	0.212
Iono	0.091	0.034	0.079	0.056	0.125
Bupa	0.368	0.322	0.345	0.322	0.345
Ecoli	0.082	0.176	0.176	0.235	0.188
Yeast	0.312	0.312	0.312	0.303	0.317
Wdbc	0.083	0.097	0.111	0.083	0.083

Table 1: Experimental results for UCI data sets.

We used 75% of the data set for training and the remainder for testing. The results for $L = 2, 4, 6$ quantization levels are shown in Table 1. Note that in several cases the quantized algorithm actually outperforms a naive Bayes algorithm (NB) with access to the real-valued features. This result may be due in part to the fact that our quantizer is based on a discriminative classifier, but it is worth noting that similar improvements over naive Bayes have been reported in earlier empirical work using classical discretization algorithms (Dougherty et al., 1995).

6. Conclusions

We have presented a new approach to the problem of decentralized decision-making under constraints on the number of bits that can be transmitted by each of a distributed set of sensors. In contrast to most previous work in an extensive line of research on this problem, we assume that the joint distribution of sensor observations is unknown, and that only a set of data samples is available. We have proposed a novel algorithm based on kernel methods, and shown that it is quite effective on both simulated and real-world data sets.

This line of work can be extended in a number of directions. First, although we have focused on discrete observations X, it is natural to consider continuous signal observations. Doing so would require considering parameterized distributions $Q(Z|X)$. Second, our kernel design so far makes use of only rudimentary information from the sensor observation model, and could be improved by exploiting such knowledge more thoroughly.

References

Bartlett, P., Jordan, M. I., & McAuliffe, J. D. (2003). *Convexity, classification and risk bounds* (Technical Report 638). Department of Statistics, UC Berkeley.

Blum, R. S., Kassam, S. A., & Poor, H. V. (1997). Distributed detection with multiple sensors: Part II—advanced topics. *Proceedings of the IEEE*, *85*, 64–79.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Proceedings of the ICML*. San Mateo, CA: Morgan Kaufmann.

Jaakkola, T., & Haussler, D. (1999). Exploiting generative models in discriminative classifiers. *NIPS 11*. Cambridge, MA: MIT Press.

Koltchinskii, V., & Panchenko, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, *30*, 1–50.

Nguyen, X., Wainwright, M. J., & Jordan, M. I. (2004). *Decentralized detection and classification using kernel methods* (Technical Report 658). Dept. of Statistics, UC Berkeley.

Rockafellar, G. (1970). *Convex analysis*. Princeton: Princeton University Press.

Scholkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Tsitsiklis, J. N. (1993). Decentralized detection. In *Advances in statistical signal processsing*, 297–344. JAI Press.

Tsuda, K., Kin, T., & Asai, K. (2002). Marginalized kernels for biological sequences. *Bioinformatics*, *18*, 268–275.

van der Vaart, A. W., & Wellner, J. (1996). *Weak convergence and empirical processes*. New York, NY: Springer-Verlag.

van Trees, H. L. (1990). *Detection, estimation and modulation theory*. Melbourne, FL: Krieger Publishing Co.

Zhang, T. (2004). Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, *52*, 56–134.