PHYSICAL REVIEW E **88**, 022118 (2013)

Optimal sensor selection for noisy binary detection in stochastic pooling networks

Mark D. McDonnell,[1,*] Feng Li,[2,†] P.-O. Amblard,[3,4,‡] and Alex J. Grant[1,§]

[1]*Institute for Telecommunications Research, University of South Australia, SA 5095, Australia*

[2]*Department of Electrical and Electronic Engineering, University of Melbourne, Australia*

[3]*CNRS (GIPSA-lab, Department of Images and Signals), Grenoble, France*

[4]*Department of Mathematics & Statistics, University of Melbourne, Australia*

(Received 13 June 2013; published 12 August 2013)

Stochastic Pooling Networks (SPNs) are a useful model for understanding and explaining how naturally occurring encoding of stochastic processes can occur in sensor systems ranging from macroscopic social networks to neuron populations and nanoscale electronics. Due to the interaction of nonlinearity, random noise, and redundancy, SPNs support various unexpected emergent features, such as suprathreshold stochastic resonance, but most existing mathematical results are restricted to the simplest case where all sensors in a network are identical. Nevertheless, numerical results on information transmission have shown that in the presence of independent noise, the optimal configuration of a SPN is such that there should be partial heterogeneity in sensor parameters, such that the optimal solution includes clusters of identical sensors, where each cluster has different parameter values. In this paper, we consider a SPN model of a binary hypothesis detection task and show mathematically that the optimal solution for a specific bound on detection performance is also given by clustered heterogeneity, such that measurements made by sensors with identical parameters either should all be excluded from the detection decision or all included. We also derive an algorithm for numerically finding the optimal solution and illustrate its utility with several examples, including a model of parallel sensory neurons with Poisson firing characteristics.

DOI: 10.1103/PhysRevE.88.022118 PACS number(s): 05.40.Ca, 87.19.lc, 87.19.lt

I. INTRODUCTION

Stochastic pooling networks (SPN) [1–3] are systems that comprise the following aspects: (i) a common input signal to multiple parallel sensors is independently corrupted by noise, either before arriving at the sensor or due to the sensor's physical limitations; (ii) these noisy measurements are also nonlinearly compressed by each sensor; (iii) the resulting measurements are communicated across a common physical channel that combines them into a single measurement, in such a way that this "pooling" causes no (or negligible) further loss of information about the network's input signal, in comparison with the optimal performance that could be achieved by using all individual sensor measurements [2]. The notion was originally suggested in Ref. [1].

Unlike many other kinds of networks, communication in a SPN flows from a source common to all nodes in a sensor network, in a single direction to a receiver (Fig. 1). It is, therefore, similar to the "refining sensor network" concept discussed by Ref. [4]. We also restrict attention to cases where neither the sensors' attributes, nor the pooling of measurements, can be controlled or designed, such as when they are biological sensory neurons. The idea of a *stochastic pooling network* originated from models that attempt to extract principles about processing in biological neuronal populations [5,6]. There were, therefore, two reasons for introducing the concept: (i) as a model with utility in explaining and predicting computational principles in biological brains; and (ii) as a guide to designing engineered sensor networks.

*mark.mcdonnell@unisa.edu.au

†fenli@unimelb.edu.au

‡pierre-olivier.amblard@gipsa-lab.grenoble-inp.fr

§alex.grant@unisa.edu.au

The material in Ref. [2] was focused on a number of surprising emergent properties that arise in SPNs, due to the interaction between random noise with lossy compression and redundancy. For instance it has been demonstrated that pooling in SPNs can lead to optimal (or close to optimal) measurement fusion [2]. Also, the specific case of a binary-node SPN has been extensively studied in the context of suprathreshold stochastic resonance [5–13]. The focus in such work has been on how the performance of the network changes with changing noise conditions, since stochastic resonance is said to occur when performance is maximized by some nonzero noise level [14–16].

Here, we consider a SPN model that arises, for example, in communications theory, in the form of a distributed detection network with a multiaccess channel (MAC) [17–19]. The MAC setup differs from information theoretic ones, where the sensors can code their measurements into long blocks of data. For example, see Refs. [4,20] for discussion about several information theoretic approaches to studying distributed sensor networks, where the ability to design aspects of the system is assumed.

Prominent aspects of the MAC model include: (i) multiple parallel sensors that make independent noisy observations of the same information source; (ii) local processing—e.g., quantization and modulation—within each sensor; (iii) each sensor communicates its processed data over a common channel such that the network output is the sum of each sensor's messages. The problem of when to select sensors in networks such as this has recently been studied from several perspectives [19,21]. A specific case of a SPN investigated in Ref. [22] is nearly equivalent to the MAC detection model. Thus, we are partially motivated to investigate puzzling optimization results in Ref. [22] that imply that superior detection is achieved when some sensor measurements are excluded from processing and that those sensors should be

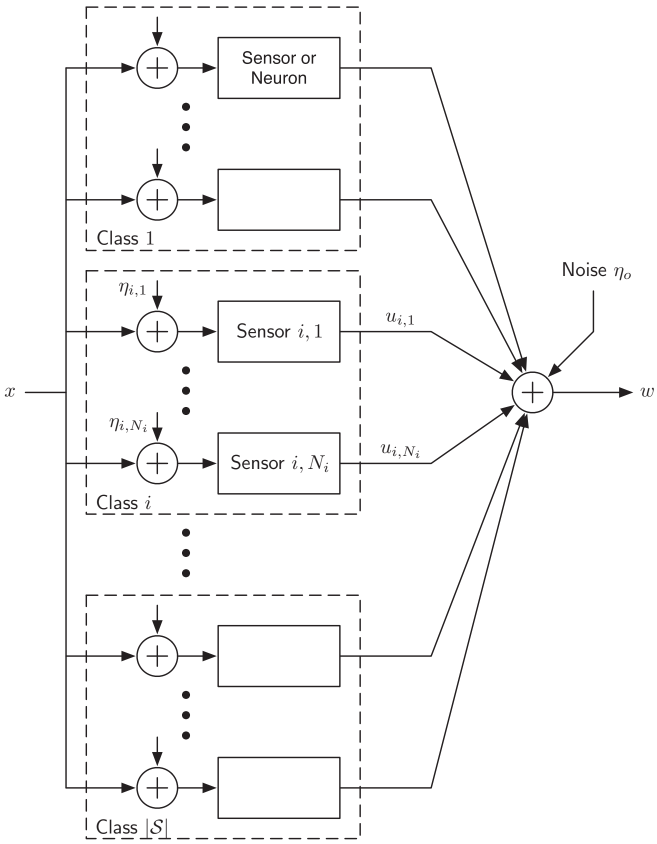

FIG. 1. Stochastic pooling network consisting of classes of identical sensors or neurons and pooling by summation of sensor outputs. Variables are defined in the text.

"switched" off, so as not to contribute to a detection decision. We note that there might be potential applications of this work in fluctuation-enhanced sensing [23] or in understanding how biological sensory systems make efficient use of their inherent redundancy [24].

This paper is organized as follows. In Sec. II we introduce notation and an objective function, which we seek to maximize by optimally selecting sensors for inclusion in a SPN. Several theorems and corollaries are provided in the Appendix that enable this section to conclude with a mathematical statement that sensors with identical attributes should either all be selected for inclusion or all excluded. Following this, in Sec. III, we state an algorithm that enables efficient numerical optimization of an arbitrary SPN of the form described in Sec. II. Section IV contains some examples that illustrate the use of the algorithm we derive. Next, Sec. V proposes some applications of this work and discusses the implications and assumptions of the derivation and results. Finally, Sec. VI introduces some possible generalizations of the results presented in this paper.

II. PROBLEM FORMULATION

A. Modeling binary detection in a stochastic pooling network

We assume a known number of sensors simultaneously sense an independently noisy sample of a common signal. We assume this signal consists of *iid* samples from a random variable, X. We write particular samples from the random variable as x and refer to x as the common input signal to all sensors.

We are interested in the case where there is a finite number of classes of sensors, such that within each class, each sensor has identical characteristics (defined mathematically below). Let $\mathcal{S}$ denote the set of all classes. Each class is associated with an integer index so that $\mathcal{S} = \{1,2,\ldots,|\mathcal{S}|\}$, where $|\mathcal{S}|$ is the cardinality of the set $\mathcal{S}$. Let N_i ($i \in \mathcal{S}$) denote the total number of sensors that belong to the ith class, and let the total number of sensors be $N = \sum_{i=1}^{|\mathcal{S}|} N_i$. We introduce indices (i,j) as subscript notation to label sensors by class and within classes; $i \in \mathcal{S}$ denotes which class the sensor belongs to and $j \in \{1,2,\ldots,N_i\}$ is the sensor's index in its class.

We also assume the signal, x, is corrupted by independent additive random sensor noise, with values $\eta_{i,j}$, so that sensor (i,j) observes $y_{i,j} = x + \eta_{i,j}$. Each sensor produces a response $u_{i,j} = f_{i,j}(x + \eta_{i,j})$ and communicates this through a multiaccess channel that, by definition, pools the responses. The function $f_{i,j}(\cdot)$ represents each sensor's local processing of its inputs; all sensors within a class have the same $f_{i,j}(\cdot)$. We assume the pooling function is one that simply sums the N individual sensor communications but that the pooled sum is subject to independent additive channel noise, η_o, with mean ξ_o and variance σ_o^2. We denote the observed SPN output as the signal $w = z + \eta_o$, where $z = \sum_{i=1}^{|\mathcal{S}|}\sum_{j=1}^{N_i} u_{i,j}$.

For a binary hypothesis task, the random variable X is binary. We therefore label two hypotheses, H_0 and H_1, such that the common input to the network, x, equals s_1 when H_1 is true and equals s_0 when H_0 is true (we assume $s_1 > s_0$ and $s_0,s_1 \in \mathbb{R}$). We denote the prior probabilities for H_1 and H_0 by P_1 and $P_0 = 1 - P_1$, respectively.

B. Measuring detection performance using Mahalanobis distance

We assume that the goal is to minimize the Bayesian probability of error with respect to correctly deciding which hypothesis is true, based on the network output, w. But, because explicit expressions for this are difficult to find, we instead aim to bound this measure. Let the network's output have conditional mean μ_1, μ_0 and variance Σ_1, Σ_0 under H_1 and H_0, respectively. The *Mahalanobis distance* is defined [25] as

$$\Delta := \sqrt{\frac{(\mu_1 - \mu_0)^2}{P_1\Sigma_1 + P_0\Sigma_0}}. \tag{1}$$

Maximizing Δ is equivalent to minimizing an upper bound on the error probability [26]. Note that we have previously studied a similar problem using a related metric [19]. However, in that work, there was no such relationship between error probability and the metric used, nor did we derive an algorithm that enables efficient numerical solutions.

We can write the conditional mean of the network output w given each hypothesis ($m = 0,1$) in terms of z and the channel noise mean and variance as $\mu_m = \mathrm{E}[w|H_m] = \mathrm{E}[z|H_m] + \xi_o$, and the conditional variance as $\Sigma_m = \mathrm{var}[w|H_m] = \mathrm{var}[z|H_m] + \sigma_o^2$, since η_o is independent of z. Since

$z = \sum_{i=1}^{|\mathcal{S}|} \sum_{j=1}^{N_i} u_{i,j}$, we have

$$\mathrm{E}[z|H_m] = \sum_{i=1}^{|\mathcal{S}|} \sum_{j=1}^{N_i} \mathrm{E}[u_{i,j}|H_m]. \tag{2}$$

We have defined classes such that all sensors within a class have the same conditional mean and variance. Let us denote the conditional mean response for a sensor in class i under hypothesis m as $\bar{u}_i^m$, and the conditional variance of its response as $\bar{v}_i^m$.

Recall we assume that $s_1 > s_0$. Similarly, we assume $\bar{u}_i^1 > \bar{u}_i^0$, so that the mean response of each sensor under H_1 is greater than the mean response of each sensor under H_0.

Note that due to the independence of the additive noise signals, the responses of each sensor are independent when conditioned on a hypothesis. Therefore, we can write

$$\mathrm{E}[z|H_m] = \sum_{i=1}^{|\mathcal{S}|} N_i \bar{u}_i^m, \quad \mathrm{var}[z|H_m] = \sum_{i=1}^{|\mathcal{S}|} N_i \bar{v}_i^m. \tag{3}$$

We introduce several vectors: $\mathbf{N} = [N_1, \ldots, N_{|\mathcal{S}|}]^\top$, $\boldsymbol{\mu}_m := [\bar{u}_1^m, \ldots, \bar{u}_{|\mathcal{S}|}^m]^\top$, $\mathbf{v}_m := [\bar{v}_1^m, \ldots, \bar{v}_{|\mathcal{S}|}^m]^\top)$, $\mathbf{a} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$, and $\mathbf{b} = P_0\mathbf{v}_0 + P_1\mathbf{v}_1$. The Mahalanobis distance can be expressed as

$$\Delta = \sqrt{\frac{(\mathbf{a}^\top \mathbf{N})^2}{\mathbf{b}^\top \mathbf{N} + \sigma_o^2}}. \tag{4}$$

We assume that $\bar{v}_i^m > 0$ for all i, which forces $b_i > 0$ for all i. Since $\bar{u}_i^1 > \bar{u}_i^0$, we have $a_i > 0\ \forall i$.

C. Optimizing performance by inclusion or exclusion of sensors

The Mahalanobis distance may be increased or decreased if the number of sensors in any class are changed. We assume that a larger control system may adaptively modify the number of sensors by switching them on or off, depending on whether they increase the Mahalanbis distance. It is therefore of interest to determine the optimal selection of sensors, in terms of the number in each class that should be switched on, given that N_i are available for each class.

We therefore introduce a vector $\mathbf{n} = [n_1, \ldots, n_{|\mathcal{S}|}]^\top$, where each element denotes the number of sensors in a class that are switched on in a trial solution. The aim is to maximize Δ by choosing the optimal vector $\mathbf{n}$. We denote the optimal solution as $\mathbf{n}^*$.

The Mahalanobis distance can be expressed for any trial solution $\mathbf{n}$ as

$$\Delta = \sqrt{\frac{(\mathbf{a}^\top \mathbf{n})^2}{\mathbf{b}^\top \mathbf{n} + \sigma_o^2}}. \tag{5}$$

Since maximization of Δ is equivalent to maximizing Δ^2, we seek to solve the following optimization problem:

$$\mathcal{P}_1 : \max_{\mathbf{n}} \frac{(\mathbf{a}^\top \mathbf{n})^2}{\mathbf{b}^\top \mathbf{n} + \sigma_o^2}$$
$$\text{s.t. } n_i \in \{0,1,\ldots,N_i\}\ (i = 1,\ldots,|\mathcal{S}|). \tag{6}$$

We assume there exists at least one $a_i \neq 0$, so that the maximum of $\mathcal{P}_1$ is greater than 0. We also assume that $\sigma_o > 0$, which ensures at least one $n_i^* > 0$. Since the denominator of the objective function is affine and the numerator is convex, the objective function of $\mathcal{P}_1$ is also convex [27]. However, solving $\mathcal{P}_1$ is not a trivial problem, because we need to maximize the objective function instead of minimizing it and what makes it more complicated is that the set of all feasible $\mathbf{n}$ is discrete.

However, in the Appendix we prove that the problem can be manipulated in a way that leads to a polynomial-time solution algorithm. This relies on the following result, that is proven in the Appendix and stated as Corollary 2. A key aspect is the consideration of the ratio b_i/a_i, i.e., the ratio of the ith element of $\mathbf{b}$ to the ith element of $\mathbf{a}$, where $i = 1,\ldots,|\mathcal{S}|$. We restate the Corollary here:

Corollary 2: The optimal solution to the original problem, Problem $\mathcal{P}_1$, where sensors are classed into groups where all ratios b_i/a_i are identical within that group, is such that all sensors in a group are either all included or all excluded.

We now use this result to state an algorithm that enables numerical solution of problem $\mathcal{P}_1$.

III. AN ALGORITHM FOR DETERMINING THE OPTIMAL SOLUTION

Corollaries 1 and 2 proven in the Appendix suggest the following polynomial time algorithm for finding the optimal solution to problem $\mathcal{P}_1$. The order of complexity of this algorithm is actually $O[N \log(N)]$, which is a direct consequence of the requirement to use a sorting algorithm on a vector of length N [28].

Algorithm 1.

(1) Evaluate all ratios $\frac{b_i}{a_i}$ for each class, i, and sort these in ascending order.

(2) Set $k = 1$ and $n_i = 0\,\forall i$

(3) Loop until $k = |\mathcal{S}|$

(a) Set $n_i = N_i$ for the k smallest ratios, and $n_i = 0$ otherwise, and evaluate the resulting objective function, z_k from Problem $\mathcal{P}_1$.

(b) Set $k = k + 1$.

(4) Search all z_k for the largest value. The index k^* where this occurs provides the optimal solution where $n_i^* = N_i$ for the k^* smallest ratios and $n_i^* = 0$ otherwise.

IV. EXAMPLES

A. Example 1: local binary quantizing by each sensor, Gaussian noise, and equiprobable hypotheses

We assume that each sensor operates on its inputs to produce an output $u_{i,j}$, with a binary quantizing function as follows:

$$u_{i,j} = f_{i,j}(x + \eta_{i,j}) = \begin{cases} 1, & x + \eta_{i,j} \geqslant \theta_i \\ 0, & x + \eta_{i,j} < \theta_i, \end{cases} \tag{7}$$

where $j = 1,..,N_i$ and $i = 1,..,|\mathcal{S}|$.

Using this specific function, and a zero-mean Gaussian noise assumption, we write the probability that the response of any sensor in group i is $u_{i,j} = 1$, given hypothesis 1 as

$$P_{11,i} = 0.5 - 0.5\mathrm{erf}\left(\frac{\theta_i - s_1}{\sqrt{2}\sigma_i}\right) \tag{8}$$

and the probability that the response of any sensor in group i is $u_{i,j} = 1$, given hypothesis 0 as

$$P_{10,i} = 0.5 - 0.5\text{erf}\left(\frac{\theta_i + s_0}{\sqrt{2}\sigma_i}\right), \tag{9}$$

where erf(·) is the error function [29].

Assuming that the probabilities of each hypothesis are $P_0 = P_1 = 0.5$, we can easily derive

$$\frac{b_i}{a_i} = \frac{P_{10,i}(1 - P_{10,i}) + P_{11,i}(1 - P_{11.i})}{2(P_{11.i} - P_{10.i})}, \tag{10}$$

where $i = 1,..,|\mathcal{S}|$. This result uses the fact that $\text{E}[u_{i,j}|x] = P_{1m,i}$ and $\text{var}[u_{i,j}|x] = P_{1m,i}(1 - P_{1m,i})$ with $m = 0,1$.

1. *Specific results 1*

We consider a scenario with the following parameter values:

(1) input states: $s_0 = -1$ under hypothesis H_0 and $s_1 = 1$ under hypothesis H_1;

(2) number of sensor groups: $|\mathcal{S}| = 3$, with N_i identical for all groups;

(3) Gaussian noise with variance $\sigma_i^2 = 1$ for all sensors;

(4) output noise variance: $\sigma_o^2 = 10^{-20}$.

When all sensor groups are the same size, N, the Mahalanobis distance, is proportional to the Mahalanobis distance obtained when $N_i = 1\,\forall i$, with σ_o^2 being N times larger. We define Δ_N to represent the former and Δ_1 to represent the latter, and it is straightforward based on Eq. (4) to show that $\Delta = \sqrt{N}\Delta_1$. Therefore, under the conditions of this example where $\sigma_0^2 \simeq 0$, we consider results for the case of $N_i = 1\,\forall i = 1,2,3$.

In order to show how the number of sensors that should be included varies with parameter values, we consider the case where group 3 has a fixed threshold value of $\theta_3 = 0.3$ and allow θ_1 and θ_2 to vary such that $\theta_1 \in [-5,5]$ and $\theta_2 \in [-5,5]$.

Figure 2 shows the optimal solution for this scenario, and Fig. 3 shows the log of the resultant optimal value of the Mahalanobis distance, Δ^*, divided by $\sqrt{N}$.

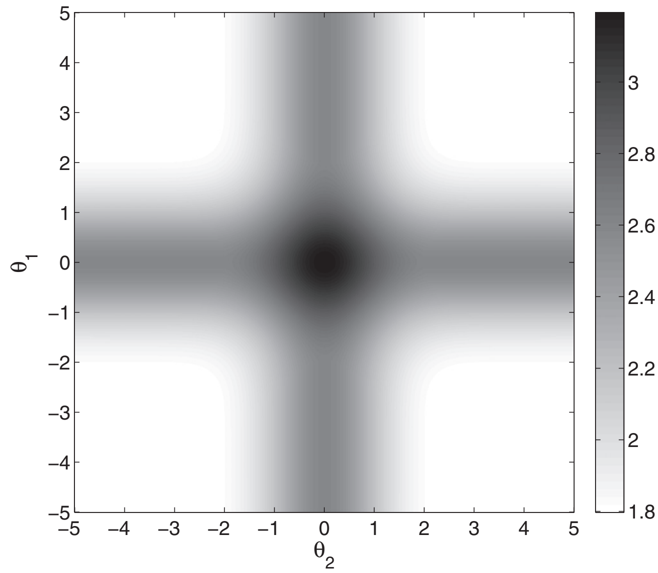

FIG. 3. Maximum value of the objective function, Δ_1^*, for the scenario in Sec. IV A1.

2. *Specific results 2*

We consider the same parameters as for the previous results, except we now include larger output noise with variance $\sigma_o^2 = 0.015$. The optimal solution is shown in Fig. 4. The increased output noise has the effect of increasing the parameter range over which it is optimal for all classes to be included. Indeed, we found that for $\sigma_o^2 > 0.448$, it was optimal to include all classes for the conditions considered.

Note, however, that this result was for $N_i = N = 1\,\forall i$. If each of our three classes has larger N, then this is equivalent to reducing the effective output noise from σ_o^2 to $\sigma_o^2/\sqrt{N}$, and N need not be very large for the optimal solution to approach that for the absence of output noise.

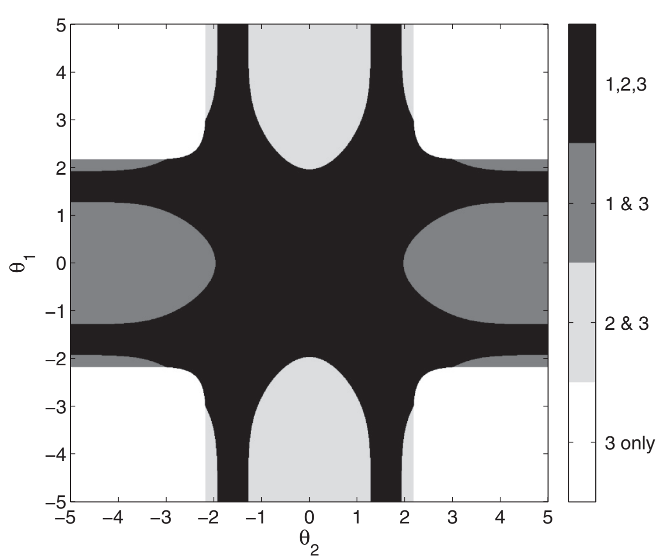

FIG. 2. Optimal solution for which classes to include to maximize the Mahalanobis distance, for the scenario in Sec. IV A1.

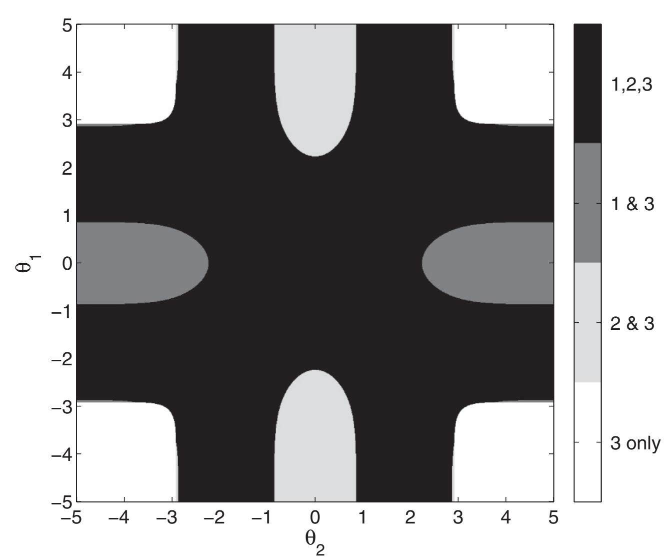

FIG. 4. Optimal solution for which classes to include to maximize the Mahalanobis distance, for the scenario in Sec. IV A2.

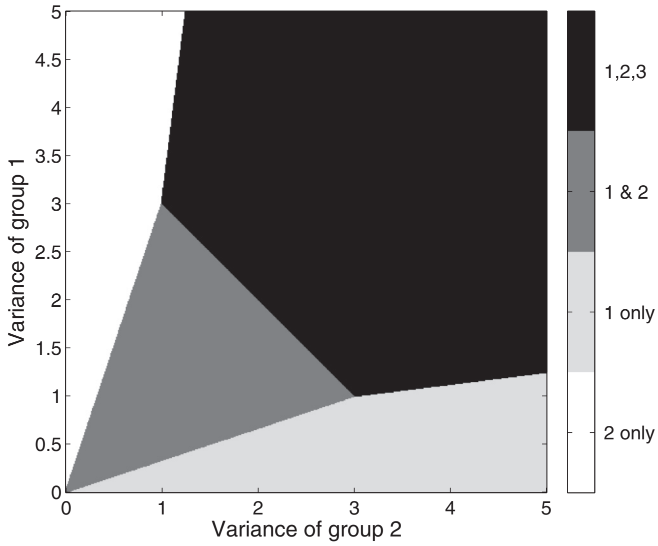

FIG. 5. Optimal solution for which classes to include to maximize the Mahalanobis distance, for the linear sensor scenario in Sec. IV B.

B. Example 2: linear sensors with different sensor noise variances

We consider now the case where $u_{i,j} = y_{i,j}$ but where each sensor group may have a different variance, σ_i^2, for the additive input noise, $\eta_{i,j}$, on each of its sensors $j = 1,..,N_i$. We again consider a scenario with $s_0 = -1$ under hypothesis H_0 and $s_1 = 1$ under hypothesis H_1, $|\mathcal{S}| = 3$, with N_i identical for all groups, Gaussian input noise, and small output noise variance: $\sigma_o^2 = 0.01$.

We now vary the input noise variance for sensor classes 1 and 2 on [0,5] and fix the variance of class 3 to $\sigma_3^2 = 5$. Figure 5 shows the optimal solution for this scenario. It is optimal for class 3 to be excluded unless the variance for both classes 1 and 2 are relatively high, and optimal to include only class 1 or only class 2, where those classes' variance are much smaller than that of the other classes.

C. Example 3: A feedforward network of Poisson neuron models

Suppose each sensor in a SPN is a Poisson neuron [30] with a rate that depends on the sum of the common input signal sample, x, and input noise that models synaptic noise. Then, for binary X where we have the two hypotheses $x = s_0$ and $x = s_1$, when conditioned on the hypothesis and the synaptic noise, we may model the output of the jth neuron in the ith class, $u_{i,j}$, as distributed according to $\mathcal{P}[\lambda_i(x + \eta_{i,j})]$, where $x = s_0$ or s_1 according, respectively, to hypothesis 0 or 1, and $\mathcal{P}$ represents the Poisson distribution.

We need to obtain the mean and variance of $u_{i,j}$ conditionally to the hypothesis. When conditioned on the synaptic noise and the hypothesis, the overall SPN output z is a Poisson random variable, Z, with rate function given by

$$\lambda_Z = \sum_{i=1}^{|\mathcal{S}|}\sum_{j=1}^{N_i} \lambda_i(x + \eta_{i,j}). \tag{11}$$

Since Z is conditionally Poisson, λ_Z is also both the conditional mean and conditional variance of Z, given x and the noise. The mean and variance of Z, given the hypothesis, can be calculated from this as follows:

$$E[Z|x] = E_\eta\left[\sum_{i=1}^{|\mathcal{S}|}\sum_{j=1}^{N_i} \lambda_i(x + \eta_{i,j})\right]. \tag{12}$$

For the variance, we get

$$\mathrm{var}[Z|x] = E_\eta[\mathrm{var}[Z|x,\eta]] + \mathrm{var}_\eta[E[Z|x,\eta]] \tag{13}$$

$$= E_\eta[\lambda_Z] + \mathrm{var}_\eta[\lambda_Z]. \tag{14}$$

To go further, we have to choose a specific form for the rate function, $\lambda_i(\cdot)$, for each sensor. We choose here, for the sake of illustrating using an example with neuroscience application, the limiting form of a sigmoid, or $\lambda_i(y) = \lambda_0 + (\lambda_1 - \lambda_0)\mathcal{I}(y - \theta_i)$, where $\mathcal{I}$ is the indicator function, and θ_i is a threshold value. Thus, neuron i fires at a rate λ_1 if the threshold θ_i is exceeded by the input, and at a lower rate $\lambda_0 \geqslant 0$ in its rest state. This ensures $a_i > 0\,\forall i$. We set $\Delta\lambda = \lambda_1 - \lambda_0$. Under this model, the calculation of the ingredients entering the Mahalanobis distance can be written explicitly.

Conditionally to η, the rate function is the sum of binary independent random variables. We assume Gaussian noise, and therefore we can use the notation from example 1 as follows: $\lambda_i(x + n_{i,j})$ takes the value λ_1 if $x + n_{i,j} > \theta_i$, and this occurs with probability $P_{1m,i}$, and the value λ_0 with probability $P_{0m,i} = 1 - P_{1m,i}$.

We obtain from this that the vector **a** has components $a_i = \Delta\lambda(P_{11,i} - P_{10,i})$, and the vector **b** has the components $b_i = \lambda_0 + \Delta\lambda\sum_{m=1}^{2} P_m P_{1m,i}[1 + \Delta\lambda(1 - P_{1m,i})]$. We also find that

$$E[Z|x] = N\lambda_0 + \Delta\lambda\sum_{i=1}^{|\mathcal{S}|} N_i P_{1m,i}, \tag{15}$$

$$\mathrm{var}[Z|x] = N\lambda_0 + \Delta\lambda\sum_{i=1}^{|\mathcal{S}|} N_i P_{1m,i} + (\Delta\lambda)^2\sum_{i=1}^{|\mathcal{S}|} N_i P_{1m,i}(1 - P_{1m,i}). \tag{16}$$

An example for the optimal solution for three classes with different threshold levels is shown in Fig. 6. The parameter values used for Fig. 6 are as follows:

(1) input states: $s_0 = -1$ under hypothesis H_0 and $s_1 = 1$ under hypothesis H_1;
(2) number of sensor groups: $|\mathcal{S}| = 3$, with N_i identical for all groups;
(3) Gaussian noise with variance $\sigma_i^2 = 0.5$ for all sensors;
(4) Output noise variance $\sigma_o^2 = 10^{-20}$;
(5) Poisson rates $\lambda_0 = 1$, $\lambda_1 = 3$.

As with Example 1, we consider the case where group 3 has a fixed threshold value, which in this case is $\theta_3 = 3$ and allow θ_1 and θ_2 to vary such that $\theta_1 \in [-5,5]$ and $\theta_2 \in [-5,5]$.

V. DISCUSSION

We have demonstrated that although adding sensors will generally enhance performance in a sensor network that

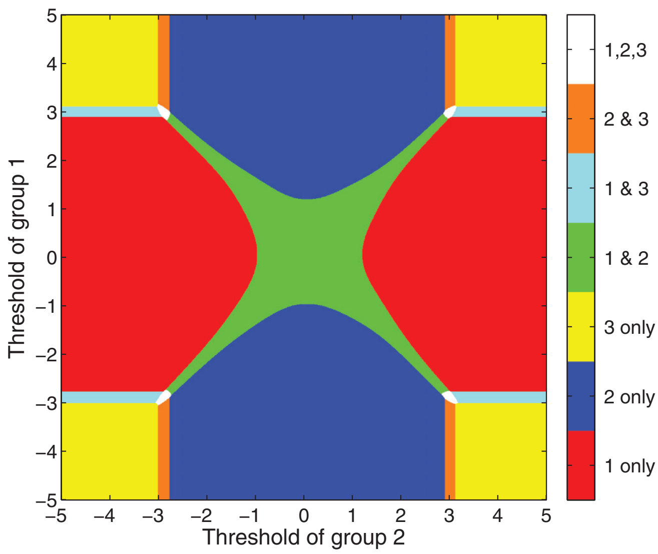

FIG. 6. (Color online) Optimal solution for which classes to include to maximize the Mahalanobis distance, for the Poisson neuron population scenario in Sec. IV C.

employs parallel sensors whose measurements are pooled, adding sensors with the wrong parameter values may diminish performance, and it is superior to use fewer sensors by excluding these. Our theorems lead to a criteria for deciding whether to add or exclude sensors such that either all sensors in a class of identical sensors should be included or all in the class should be excluded.

Given that the sensor model we use has been applied in simple neuronal network models, this result suggests the possibility that the brain may benefit from selectively inhibiting inputs from some sensory neurons during integration of information. Alternatively, structural plasticity [31] could potentially cause exclusion of responses from sensory neurons through removal of synaptic connections between those neurons and a pooling neuron. However, drawing conclusions about real neurobiology will require more biophysically realistic neuron models than those considered here, and that is left for future work.

VI. EXTENSIONS

In this final section of this paper we discuss several possible extensions to the work presented in this paper.

A. Constraining the maximum number of "on" sensors

It is straightforward to extend the optimization problem considered above to include a constraint that limits the number of sensor groups that should be switched on to some maximum number. That is, we can add a constraint to problem $\mathcal{P}_1$ such that $\sum_{i=1}^{|\mathcal{S}|} n_i \leqslant C < N$. Inclusion of such a constraint does not lead to any changes in the derivation of the sensor group selection algorithm, since including it the Lagrangian leads to very similar results to those of Theorem 2 and its corollaries. The only change required in the algorithm would be to exclude from the search for the maximum value of z_k any corresponding solutions that violate the constraint.

B. Correlated sensor noise

The formulation contained in Ref. [22] differs from that here, in that in Ref. [22], (i) each hypothesis is a Gaussian distribution with different means, rather than constants; and (ii) the performance measure is based on an output decision threshold and probability of error. The first difference is equivalent to assuming all sensors are subject to a common noise signal, e.g., interference, in addition to *iid* sensor noise. Despite these small differences, our results partially support the numerical optimization results in Ref. [22], which show that sometimes sensor classes should be removed. In future work, we plan to examine whether the exact form of the numerical optimization results considered in Ref. [22] is dependent on the measure used.

However, regarding the problem description in Ref. [22], if we attempt to generalize our results to the case of a common noise source across all sensors, we obtain a more general version of the problem solved above as follows.

The problem studied so far can be viewed as discriminating between two deterministic stimuli, which are drawn randomly according to the *a priori* probabilities P_0 and P_1. In a more general setting, we can assume that each stimulus is also randomly distributed. This can either model lack of perfect knowledge or the presence of measurement noise. However, in this context, the calculation performed earlier becomes invalid, because the outputs of each sensor are no longer conditionally independent, for a given hypothesis. The theory has to be modified to take into account this fact. Indeed, if the input is $x + s_m$ under H_m where x is some random variable, the u_{ij} are conditionally independent to H_m and x but indeed dependent when conditioned only on H_m.

We still have

$$E[z|H_m] = \sum_{i=1}^{|\mathcal{S}|} \sum_{j=1}^{N_i} E[u_{ij}|H_m] \tag{17}$$

$$= \sum_{i=1}^{|\mathcal{S}|} n_i \bar{u}_i^m, \tag{18}$$

where $\bar{u}_i^m = E_{\eta_i,m}[u_i(x,\eta_i)]$. Here, we emphasize η_i is the fluctuation or random element of the sensors of the ith class, and x is the common observation distributed according to P_m, where $m = 0,1$ is the hypothesis. To calculate the variance under H_m, we use the well-known formula for the decomposition of the variance $\text{var}[z] = E[\text{var}[z|x]] + \text{var}[E[z|x]]$. The first calculation is the same as the calculation performed above, except that we have to average over x as well. We have, using conditional independence,

$$\text{var}[z|x,m] = \sum_{i=1}^{|\mathcal{S}|} \sum_{j=1}^{N_i} \text{var}[u_{ij}|x,m], \tag{19}$$

and thus

$$E[\text{var}[z|x]] = \sum_{i=1}^{|\mathcal{S}|} n_i \bar{v}_i^m, \tag{20}$$

where $\bar{v}_i^m = E_x \mathrm{var}_{\eta_i}[u_i|x,m]$. Now the last term is the variance of the conditional mean. Let us write

$$E[z|x,H_m] = \sum_{i=1}^{|\mathcal{S}|}\sum_{j=1}^{N_i} E[u_{ij}|x,H_m] \quad (21)$$

$$= \sum_{i=1}^{|\mathcal{S}|} n_i u_i(x), \quad (22)$$

where we have introduced $u_i^m(x) = E_{\eta_i}[u_{ij}|x]$. Thus, the variance reads

$$\mathrm{var}[E[z|x]] = \sum_{i=1}^{|\mathcal{S}|}\sum_{j=1}^{N_i} n_i n_j \mathrm{cov}[u_i(x),u_j(x)|H_m]. \quad (23)$$

We now introduce the same notation as the first part of the paper: $\mathbf{a}$ is the vector containing the differences $\bar{u}_i^1 - \bar{u}_i^0$; $\mathbf{b}$ contains the terms $P_0\bar{v}_i^0 + P_1\bar{v}_i^1$, and $\mathbf{n}$ contains the n_i. Let us introduce the covariance matrices Γ^m defined elementwise by $\Gamma_{ij}^m = \mathrm{cov}[u_i(x),u_j(x)|H_m]$. These matrices are obviously positive semidefinite, as are the convex combination of the matrices for $m = 0,1$. Let $\Gamma = P_0\Gamma^0 + P_1\Gamma^1$ be such a combination. We then have for the Mahalanobis distance

$$\Delta = \sqrt{\frac{(\mathbf{a}^\top\mathbf{n})^2}{\mathbf{n}^\top\Gamma\mathbf{n} + \mathbf{b}^\top\mathbf{n}}}. \quad (24)$$

This new form, however, is no longer convex. Nevertheless, this form leads to the new optimization problem:

$$\mathcal{P}_4 : \max_{\mathbf{n}} \frac{(\mathbf{a}^\top\mathbf{n})^2}{\mathbf{n}^\top\Gamma\mathbf{n} + \mathbf{b}^\top\mathbf{n} + \sigma_o^2}$$
$$\text{s.t. } n_i \in \{0,1\}\ (i = 1,\ldots,N), \quad (25)$$

where $b_i > 0\,\forall i$, and Γ is a covariance matrix, and thus is positive semidefinite with $\Gamma_{i,j} \geqslant 0$ for all (i,j).

The case we considered previously in this paper is a special case where $a_i > 0\,\forall i$, and $\Gamma = \mathbf{0}$.

We assume there exists at least one $a_i \neq 0$, and therefore $\mathbf{n} = \mathbf{0}$ is never optimal. Note that we can write $(\mathbf{a}^\top\mathbf{n})^2 = \mathbf{n}^\top A\mathbf{n}$, where $A = \mathbf{a}\mathbf{a}^\top$ is a square symmetric matrix that has a single nonzero positive eigenvalue equal to $|\mathbf{a}|^2$. This follows because the only nonzero eigenvalue of a matrix $\mathbf{A}_1$ such that the only nonzero entry is $a_{11} = |\mathbf{a}|^2$ also has a single nonzero eigenvalues equal to $|\mathbf{a}|^2$. Thus, A is positive semidefinite.

Suppose we relax this problem so that the constraints are no longer integer to get

$$\mathcal{P}_{4a} : \max_{\mathbf{n}} \frac{\mathbf{n}^\top A\mathbf{n}}{\mathbf{n}^\top\Gamma\mathbf{n} + \mathbf{b}^\top\mathbf{n} + \sigma_o^2}$$
$$\text{s.t. } -n_i \leqslant 0\ (i = 1,\ldots,N),$$
$$n_i - 1 \leqslant 0\ (i = 1,\ldots,N). \quad (26)$$

This problem is a *fractional program* written in standard form [32,33]. A fractional program is classified as a *concave-convex fractional program* (or simply concave fractional program) if the numerator is concave and the denominator and constraints are convex, where if the denominator is not affine then the numerator must also be nonnegative [33]. Such programs have been the subject of much study in the field of fractional programming, since they can be converted into concave programs [33].

However, here since the numerator and denominator are both quadratic forms and the constraints form a convex set, problem $\mathcal{P}_{4a}$ is specifically a quadratic fractional program [33], and since both quadratic forms involve positive semidefinite matrices, we are maximizing the ratio of a convex function to a convex function. This case of a quadratic fractional program that cannot be converted into a concave-convex fractional program has been studied recently [34–36].

Note also that the original integer program, $\mathcal{P}_4$, can also be recast as a nonlinear program with polynomial constraints, as studied by [37]:

$$\mathcal{P}_{4b} : \min_{\mathbf{n}} \frac{-\mathbf{n}^\top A\mathbf{n}}{\mathbf{n}^\top\Gamma\mathbf{n} + \mathbf{b}^\top\mathbf{n} + \sigma_o^2}$$
$$\text{s.t. } n_i^2 - n_i = 0\ (i = 1,\ldots,N). \quad (27)$$

This problem is precisely one of the special cases studied by Ref. [37], since the denominator of the objective function is a positive real multivariate polynomial, and the numerator is a twice continuously differentiable concave function.

Unfortunately, the inclusion of Γ in the problem makes the solution method we derived invalid for problem $\mathcal{P}_4$. Recently published work suggests two alternative algorithms for solving this more general problem; the first is based on derivation of a transformation of the problem that allows solution using a method based on a form of branch-and-bound [34], and the second is the method presented in Ref. [36], which is based on the classical Dinkelbach method, combined with integer programming. Although the latter method seems more promising, because the problem solved in that paper is of the form of the relaxed problem $\mathcal{P}_{4a}$, it is not clear how to justify solving problem $\mathcal{P}_{4a}$ as a proxy for solving problem $\mathcal{P}_4$. Moreover, the method of Ref. [36] requires solving a linear integer programming problem (the original noninteger program is manipulated into an integer program), and integer programming in general is NP-hard and may not readily enable numerical solutions in reasonable runtimes.

ACKNOWLEDGMENTS

M. D. McDonnell's contribution was supported by an Australian Research Fellowship from the Australian Research Council (Project No. DP1093425). P.-O. Amblard is supported by a Marie Curie International Outgoing Fellowship from the European Union.

APPENDIX: SOLVING THE OPTIMAL SENSOR SELECTION PROBLEM

The mathematical pathway to a solution is expressed more readily if sensors are not grouped into classes. Therefore, we introduce the vector $\mathbf{m}$ of length N, whose elements, $m_i \in \{0,1\}$, represent whether sensor i is switched on or switched off. In order to maintain the same notation for $\mathbf{a}$ and $\mathbf{b}$, note that we can assume without loss of generality that there are as many classes as sensors, and so $|\mathcal{S}| = N$ and $N_i = 1\,\forall i$ for the purposes of this section.

Through the same reasoning that led to stating problem $\mathcal{P}_1$, re-expression in terms of $\mathbf{m}$ leads to the following integer programming problem:

$$\mathcal{P}_2 : \max_{\mathbf{m}} \frac{(\mathbf{a}^\top \mathbf{m})^2}{\mathbf{b}^\top \mathbf{m} + \sigma_o^2} \quad \text{s.t. } m_i \in \{0,1\}\ (i = 1,\ldots,N). \quad (A1)$$

We consider a relaxed version of this nonlinear integer program, namely the convex maximization problem:

$$\begin{aligned} \mathcal{P}_{2a} : \max_{\mathbf{m}} & \frac{(\mathbf{a}^\top \mathbf{m})^2}{\mathbf{b}^\top \mathbf{m} + \sigma_o^2} \\ \text{s.t. } & -m_i \leqslant 0\ (i = 1,\ldots,N) \\ & m_i - 1 \leqslant 0\ (i = 1,\ldots,N). \end{aligned} \quad (A2)$$

Since the objective function in $\mathcal{P}_2$ is convex, and the problem is a maximization problem, the relaxed version $\mathcal{P}_{2a}$ will be optimized by the same optimal solution as $\mathcal{P}_2$, since the optimal solution of a convex maximization with linear constraints is at extreme points of the convex constraint set, and the extreme points of the relaxed convex constraint set also consist of $n_i \in \{0,1\}$.

In order to reduce the solution space, we look for an equivalent convex or linear problem. We can construct such a problem, which while not directly solvable, does allow derivation of conditions that suggest a polynomial time solution algorithm for the original problem.

Let $\mathbf{m}^*$ be the optimal solution to $\mathcal{P}_{2a}$ and define $\beta = \mathbf{a}^\top \mathbf{m}^*$ (notice that $\beta > 0$ due to the assumption that at least one a_i is not equal to zero). We now construct a new problem

$$\mathcal{P}_3 : \min \mathbf{b}^\top \mathbf{x} \quad (A3)$$

$$\text{s.t. } \mathbf{a}^\top \mathbf{x} = \beta \quad (A4)$$

$$-x_i \leqslant 0\ (i = 1,\ldots,N) \quad (A5)$$

$$x_i - 1 \leqslant 0\ (i = 1,\ldots,N). \quad (A6)$$

This problem can be thought of as one where the objective is to solve Problem $\mathcal{P}_2$ by minimizing the denominator of Eq. (A2), subject to the numerator being maximal.

We now state the following theorem.

Theorem 1. $\mathbf{x} = \mathbf{m}^*$ is the optimal solution to $\mathcal{P}_3$.

Proof. We shall now prove the theorem by contradiction. Suppose there exists a $\bar{\mathbf{x}} \neq \mathbf{m}^*$ such that constraint Eqs. (A4) to (A6) are all satisfied (which implies that $\mathbf{m} = \bar{\mathbf{x}}$ is feasible for $\mathcal{P}_{2a}$) and $\mathbf{b}^\top \bar{\mathbf{x}} < \mathbf{b}^\top \mathbf{m}^*$. Then we have

$$\frac{(\mathbf{a}^\top \bar{\mathbf{x}})^2}{\mathbf{b}^\top \bar{\mathbf{x}} + \sigma_o^2} = \frac{\beta^2}{\mathbf{b}^\top \bar{\mathbf{x}} + \sigma_o^2} = \frac{(\mathbf{a}^\top \mathbf{m}^*)^2}{\mathbf{b}^\top \bar{\mathbf{x}} + \sigma_o^2} > \frac{(\mathbf{a}^\top \mathbf{m}^*)^2}{\mathbf{b}^\top \mathbf{m}^* + \sigma_o^2}, \quad (A7)$$

which means that $\mathbf{m}^*$ is not an optimal solution to $\mathcal{P}_{2a}$. This contradicts the fact that $\mathbf{m}^*$ is the optimal solution to $\mathcal{P}_{2a}$. ■

It is easy to verify that $\mathcal{P}_3$ is a convex optimzation problem (indeed, it is a linear program). In order to simplify the problem, it is of value to formulate the corresponding Lagrangian function,

$$\mathcal{L} = \mathbf{b}^\top \mathbf{x} + \sum_i \lambda_i(-x_i) + \sum_i \gamma_i(x_i - 1) + \nu(\mathbf{a}^\top \mathbf{x} - \beta). \quad (A8)$$

Since the problem is a convex program, the optimal solution $\mathbf{x} = \mathbf{m}^*$ will satisfy the following Karush-Kuhn-Tucker (KKT) [27] conditions

$$\lambda_i(-m_i^*) = 0 \quad i = 1,\ldots,N \quad (A9)$$

$$\gamma_i(m_i^* - 1) = 0 \quad i = 1,\ldots,N \quad (A10)$$

$$\lambda_i \geqslant 0, \quad \gamma_i \geqslant 0 \quad i = 1,\ldots,N \quad (A11)$$

$$b_i + a_i\nu - \lambda_i + \gamma_i = 0 \quad i = 1,\ldots,N \quad (A12)$$

Combining all the constraints above, we have the following theorem

Theorem 2. $m_i^* = 0$ if $b_i + a_i\nu > 0$ and $m_i^* = 1$ if $b_i + a_i\nu < 0$.

Proof. If $\lambda_i > 0$ then Eq. (A9) means that $m_i^* = 0$, so that then Eq. (A10) means that $\gamma_i = 0$ and, therefore, from Eq. (A12), $b_i + a_i\nu > 0$. Similarly, if $\gamma_i > 0$ then $m_i^* = 1$, $\lambda_i = 0$ and, therefore, $b_i + a_i\nu < 0$. ■

If it happens that $b_i + a_i\nu = 0$, we need $\gamma_i = \lambda_i$ for Eq. (A12) to hold. If both γ_i and λ_i are greater than 0, it will be impossible for Eqs. (A9) and (A10) to hold simultaneously. As a result, we must have $\gamma_i = \lambda_i = 0$, and we can't determine m_i^* based on the KKT conditions. However, clearly there can only be one ratio $\nu = a_i/b_i$ for which this event occurs.

Since we have assumed $a_i > 0\,\forall i$, we can prove the following corollary based on Theorem A.

Corollary 1. When $a_i > 0$ for all i, $m_i^* = 0$ if b_i/a_i is greater than a certain threshold τ and $m_i^* = 1$ if b_i/a_i is less than or equal to τ, where $\tau \neq b_i/a_i\ \forall i$.

Proof. Since $a_i > 0$, $b_i + a_i\nu > 0$ will be equivalent to $b_i/a_i > -\nu$ and $b_i + a_i\nu < 0$ will be equivalent to $b_i/a_i < -\nu$. According to Theorem 2, $m_i^* = 0$ if $b_i/a_i > -\nu$ and $m_i^* = 1$ if $b_i/a_i < -\nu$.

Now we consider the case that $b_i + a_i\nu = 0$. Notice that a negative ν is a necessary condition for $b_i + a_i\nu = 0$ to hold, since both a_i and b_i are greater than 0. Let $\mathcal{S}_\nu = \{i \in \mathcal{S} | b_i/a_i = -\nu\}$. To obtain $m_i^*(i \in \mathcal{S}_\nu)$, we need to solve the following problem:

$$\begin{aligned} \max & \frac{\left(A + \sum_{i\in\mathcal{S}_\nu} a_i y_i\right)^2}{B + \sum_{i\in\mathcal{S}_\nu} b_i y_i} \\ \text{s.t. } & y_i \in \{0,1\} \quad \text{for} \quad i \in \mathcal{S}_\nu, \end{aligned} \quad (A13)$$

where

$$A = \sum_{i\in\mathcal{S}\backslash\mathcal{S}_\nu} a_i m_i^* \geqslant 0, \quad (A14)$$

$$B = \sum_{i\in\mathcal{S}\backslash\mathcal{S}_\nu} b_i m_i^* \geqslant 0. \quad (A15)$$

Let $z = \sum_{i\in\mathcal{S}_\nu} a_i y_i$. Since $b_i + a_i\nu = 0\ (i \in \mathcal{S}_\nu)$, the objective function can be written as

$$f_\nu = \frac{(A + \sum_{i\in\mathcal{S}_\nu} a_i y_i)^2}{B + (-\nu)\sum_{i\in\mathcal{S}_\nu} a_i y_i} = \frac{(A+z)^2}{B + (-\nu)z}. \quad (A16)$$

If $A = B = 0$ (this happens when $m_i^* = 0$ for all $i \in \mathcal{S}\backslash\mathcal{S}_\nu$ or $\mathcal{S}\backslash\mathcal{S}_\nu = \emptyset$), the optimal solution will be $y_i^* = 1 (i \in \mathcal{S}_\nu)$. If $A > 0$ and $B > 0$ (this happens when $m_i^* \neq 0$ for some $i \in \mathcal{S}\backslash\mathcal{S}_\nu$), the objective function, f_ν, is convex in Z, and is, therefore, maximized either by $z = 0$ or $z = \sum a_i$, which are the endpoints of the support in a relaxed problem where $y_i \in [0,1]$. This corresponds to either $y_i^* = 0$ for all $i \in \mathcal{S}_\nu$

or $y_i^* = 1$ for all $i \in \mathcal{S}_\nu$. In the event that $y_i^* = m_i^* = 0$ for all $i \in \mathcal{S}_\nu$, we can choose τ such that $\tau < -\nu$ and for all $b_i/a_i < -\nu$ we have $b_i/a_i < \tau$. In the event that $y_i^* = m_i^* = 1$ for all $i \in \mathcal{S}_\nu$, we can choose τ such that $\tau > -\nu$ and for all $b_i/a_i > -\nu$ we have $b_i/a_i > \tau$. ■

Finally, we can use Corollary 1 to make a statement about the optimal solution to problem $\mathcal{P}_1$.

Corollary 2. The optimal solution to the original problem, Problem $\mathcal{P}_1$, where sensors are classed into groups where all ratios b_i/a_i are identical within that group, is such that all sensors in a group are either all included or all excluded.

Proof. This corollary follows from Corollary 1, since the ratio b_i/a_i will be identical for all sensors within each class. ■

[1] S. Zozor, P. Amblard, and C. Duchêne, Fluct. Noise Lett. **7**, L39 (2007).

[2] M. D. McDonnell, P. O. Amblard, and N. G. Stocks, J. Stat. Mech.: Theory Exp. (2009) P01012.

[3] M. D. McDonnell, Phys. Rev. E **79**, 041107 (2009).

[4] M. Gastpar, M. Vetterli, and P. L. Dragotti, IEEE Signal Processing Magazine **23**, 70 (2006).

[5] N. G. Stocks, Phys. Rev. Lett. **84**, 2310 (2000).

[6] N. G. Stocks and R. Mannella, Phys. Rev. E **64**, 030902(R) (2001).

[7] M. D. McDonnell, D. Abbott, and C. E. M. Pearce, Microelectron. J. **33**, 1079 (2002).

[8] T. Hoch, G. Wenning, and K. Obermayer, Phys. Rev. E **68**, 011911 (2003).

[9] D. Rousseau, F. Duan, and F. Chapeau-Blondeau, Phys. Rev. E **68**, 031107 (2003).

[10] D. Rousseau and F. Chapeau-Blondeau, Phys. Lett. A **321**, 280 (2004).

[11] M. D. McDonnell and N. G. Stocks, Scholarpedia **4**, 6508 (2009).

[12] A. Patel and B. Kosko, Neural Networks **22**, 697 (2009).

[13] G. Ashida and M. Kubo, Physica D **239**, 327 (2010).

[14] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni, Rev. Mod. Phys. **70**, 223 (1998).

[15] M. D. McDonnell and D. Abbott, PLoS Comput. Biol. **5**, e1000348 (2009).

[16] M. D. McDonnell and L. M. Ward, Nature Rev. Neurosci. **12**, 415 (2011).

[17] K. Liu and A. M. Sayeed, IEEE Trans. Sig. Proc. **55**, 1899 (2007).

[18] F. Li and J. S. Evans, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, March 31–April 4, 2008* (IEEE, Piscataway, NJ, 2008), pp. 2417–2420.

[19] F. Li, M. D. McDonnell, P.-O. Amblard, and A. J. Grant, in *Proceedings of Australian Communications Theory Workshop (IEEE), Sydney, Australia, Feb 4–6, 2009* (IEEE, Piscataway, NJ, 2009), pp. 77–82.

[20] M. Gastpar, P. L. Dragotti, and M. Vetterli, IEEE Trans. Inf. Theory **52**, 5177 (2006).

[21] S. Joshi and S. Boyd, IEEE Trans. Sig. Proc. **57**, 451 (2009).

[22] P. O. Amblard, S. Zozor, N. G. Stocks, and M. D. McDonnell, in *Proceedings SPIE Noise and Fluctuations in Biological, Biophysical, and Biomedical Systems*, edited by S. M. Bezrukov (SPIE, Bellingham, Washington, 2007), Vol. 6602, p. 66020S.

[23] P. Makra, Z. Topalian, C. G. Granqvist, L. B. Kish, and C. Kwan, Fluct. Noise Lett. **11**, 1250010 (2012).

[24] J. J. Aticky, Network: Comput. Neural Syst. **22**, 4 (2011).

[25] P. Mahalanobis, Proceeding Nat. Inst. Sci. India **2**, 49 (1936).

[26] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach* (Prentice Hall, London, 1982).

[27] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004).

[28] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, UK, 1992).

[29] M. R. Spiegel and J. Liu, *Mathematical Handbook of Formulas and Tables* (McGraw-Hill, New York, 1999).

[30] A. P. Nikitin, N. G. Stocks, R. P. Morse, and M. D. McDonnell, Phys. Rev. Lett. **103**, 138101 (2009).

[31] P. Caroni, F. Donato, and D. Muller, Nature Rev. Neurosci. **13**, 478 (2012).

[32] S. Schaible, Eur. J. Operat. Res. **7**, 111 (1981).

[33] S. Schaible and T. Ibaraki, Eur. J. Operat. Res. **12**, 325 (1983).

[34] H. P. Benson, Eur. J. Operat. Res. **173**, 351 (2006).

[35] J. B. G. Frenk, Eur. J. Operat. Res. **176**, 641 (2007).

[36] R. Yamamoto and H. Konno, J. Optim. Theory Appl. **133**, 241 (2007).

[37] V. Jeyakumar and G. Y. Li, Math. Program. **126**, 393 (2011).