A generative-discriminative learning model for noisy information fusion

Thomas Hecht* and Alexander Gepperth*†
*Computer science and system engineering department - ENSTA ParisTech
828 boulevard des Maréchaux, 91762 Palaiseau Cedex - France
† Inria FLOWERS, Inria Bordeaux Sud-Ouest
200 avenue de la vieille tour, 33405 Talence - France
Email: {thomas.hecht, alexander.gepperth}@ensta-paristech.fr

***Abstract*—This article is concerned with the acquisition of multimodal integration capacities by learning algorithms. Humans seem to perform statistically optimal fusion, and this ability may be gradually learned from experience. In order to stress the advantage of learning approaches in contrast to hand-coded models, we propose a generative-discriminative learning architecture that avoids simplifying assumptions on prior distributions and copes with realistic relationships between observations and underlying values. We base our investigation on a simple self-organized approach, for which we show statistical near-optimality properties by explicit comparison to an equivalent Bayesian model on a realistic artificial dataset.**

I. Introduction

Autonomous agents must possess the ability to acquire knowledge from multiple sensors in an unsupervised way. Similarly, proper multi-sensor fusion is a necessity for any biological organism, and it seems that, under certain conditions, humans and other animals can perform statistically optimal multisensory integration [1]. As to how this is achieved, many questions remain : mainly, one can speculate whether there is a generic, sensor-independent fusion mechanism taking into account the basic statistical laws such as Bayes' rule at some neural level, or whether optimal fusion, where it occurs, is something that is fully learned through experience [2].

In this article we present an argument for the latter case since it stands to reason that multisensory integration in biological systems is not generally innate but learned [3], [4]. This ability seems to be gradually acquired in the course of development, and then refined and maintained throughout a lifespan which would obviously be desirable for intelligent agents to reproduce. Thus not only the question of how multisensory integration is carried out is of importance, but also of how it is acquired. As we believe that the acquisition and maintenance of fusion skills is a life-long task, we propose a neural model capable of performing stable multisensor fusion without assumptions about noise variances. Our approach uses a hybrid architecture combining an unsupervised self-organizing map (SOM) layer with a supervised linear regression layer.

This article wishes to demonstrate that a SOM can learn a viable representation of a given data distribution given by examples thus laying the basis for an optimal fusion decision taken by a subsequent linear regression module. We put emphasis of advantages of such a generative-discriminative model by showing that the discriminative ability to focus on a given-task best performance can be modulated by underlying statistics of data distribution captured by the unsupervised generative model. We compare our system final modulated decision with the well-known Bayesian fusion optimum.

II. Related Work

Also called "multisensory data fusion", multimodal integration aims at providing a robust and unified representation of the environment based on multiple sensors inputs [5]. In mammalian brains, this process seems to be implied in maximizing information gathering and reliability by the effective use of a set of available observations from different sensors [6]. It has been widely studied at different levels (i.e. from particular cortical cells to individual behaviour) and into different scientific fields (e.g. neurophysiology, neuroimagery, psychophysics or neurobiology).

There are many reasons for believing that self-organization is a fundamental mechanism used in the brain [7]. Studies dealing with self-organizing artificial neural networks and multimodal fusion are usually based on properties one can find in biology : continuous unsupervised processes, adaptation and plasticity, topological preservation of the input space relationship or dimensionality reduction. Self-organized approaches have the potential to establish a transition from high-dimensional, noisy and modality-specific sensor readings to abstract, multimodal and symbolic concepts, whereas they are considered less appropriate for reproducing statistical optimality which should be respected by any integration process.

Some recent papers used SOM or SOM-like bio-imitating architectures in order to reproduce individual behaviours or biological phenomena by imitating, more or less clumsily, hierarchical layered interconnected cortical areas, especially superior colliculus ([8] and [9] which place emphasis on the positive impact of a non-linear transfer function applied to neural maps and [10], [11] and [12] which deal with imitating SC multisensory integration). Several studies aim at designing models inspired by recent neurophysiological findings without fully copying nature architectures or processes, focusing on

Thomas Hecht gratefully acknowledges funding support by the Direction Générale de l'Armement (DGA-MRIS scholarship) and École Polytechnique.

Preprint submitted to 5th International Conference on Development and Learning and on Epigenetic Robotics. Received March 23, 2015.

well-defined applications like information retrieval [13], visual categorisation [14], data visualisation [15] or audio-visual multimodal integration [16].

III. Methods

A. *Hybrid model*

In this section, we detail components and notations of the proposed neural architecture. Our bio-inspired hybrid learning method, depicted in Fig. 1, is presented noisy observations $s_i, i = 0 \dots N-1$, obtained one after the other from N sensors, where the dependence of observations on the underlying "true" sensor value r is described by a probabilistic noise model (throughout the article, we will use $N = 2$ sensors to keep things simple). In order to increase the computational capacities of the architecture, observations and the "true" underlying value r are represented using a technique variously termed "population encoding" or "basis function representation", described in Sec. III-B. From the set of population-coded observations, the SOM algorithm III-C creates a combinatorial *internal representation* capturing key statistical properties of observations. Each neuron-like unit in this representation preferentially responds to certain input stimuli, the generative aspect of the SOM algorithm ensuring that common combinations cause strong responses and uncommon ones weak responses. Subsequently, a *decoder module* maps the internal representation to the population-coded "true value" r by using a multiple linear regression (MLR) algorithm (see Sec. III-D) resulting in an estimate r^*_{nn}. Learning the decoder function happens in a supervised fashion using the known underlying values r.

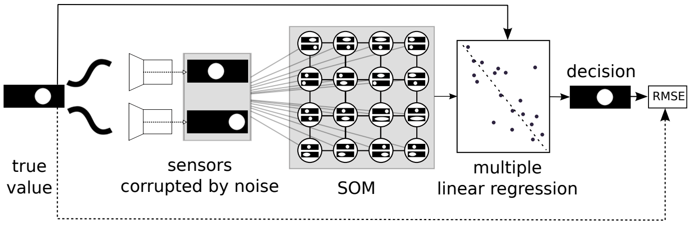

Fig. 1: Detailed illustrative view of the proposed system.

To verify that the internal representation indeed captures important statistical properties relevant to the fusion process, an alternative fusion mechanism that linearly maps sensory inputs to true values is investigated as well, producing estimates denotes r^*_{alt} . The comparison of population-coded estimates and true values is performed after a decoding step using the mean squared error (MSE) measure.

B. *Population encoding for raw sensor values*

Population coding or "basis function encoding" allows a common representational format based on biological observations [17] : this kind of discrete information coding occurs at the sensory input areas of the brain [18], [19]. A quantity is encoded as an "activation" on a two-dimensional surface, where the position and amplitude of the activation "blob" respectively code for value and confidence [20].

We use a simpler version of the population coding idea in order to enrich the representation of the raw unimodal sensor measurement by projecting scalars from a specific range into a unified two-dimensional grid of $l \times c$ of pixels : values are still encoded by placing a 2D Gaussian centered on a determined position but we do not encode the confidence part of the quantity into the block. The main reason why we do so is because the very point of our system is to identify the variability in data, whatever distribution it comes from. The encoding process translates a continuous variable from its own continuous range to a discrete position into a discrete space. Then a 2D Gaussian blob with given x- and y-axis variance is placed at this position. To prevent the center of the blob from exceeding the discrete border (e.g. because of the addition of noise to the value to encode) we usually set a margin δ which is used in the discretization process : with a and b being the continuous borders and c and d the discrete borders, the translation $[a, b] \to [c; d]$ is replaced by $[a, b] \to [c + \delta; d - \delta]$.

C. *Kohonen's Self-Organizing Map*

The Kohonen's Self-Organizing Map (SOM) algorithm is a topology-based ("nearby patterns should be mapped by nearby neurons") unsupervised clustering method which combines biological inspiration and a solid mathematical foundation. The algorithm assumes the existence of $N = p \times q$ competing units receiving m-dimensional input vectors $\mathbf{x}_i \in X$ and implements a winner-takes-most learning strategy. Each unit u_j has an associated *codebook vector* or *prototype* $\mathbf{w}_j$ which is adapted over time. For each iteration t, an input sample $\mathbf{x}^t$ is randomly picked from X and the learning algorithm determines the best-matching unit (BMU). Usually, the BMU is found by applying a distance function. In this article, we use the so-called dot-product maps [21] in which codebook vectors $\mathbf{w}_j$ are compared to the current input sample $\mathbf{x}^t$ through the *cosine similarity*. The current BMU corresponds to the unit for which $\mathbf{w}^{\star t} = argmax_j \ sim(\mathbf{x}^t, \mathbf{w}_j)$ ($\mathbf{x}^t$ and $\mathbf{w}_j$ are kept normalized to constant length). The grid then moves its prototype a certain proportion η^t of this similarity closer to $\mathbf{x}^t$. A set of adjoining codebook vectors around the BMU are also moved closer to the current training case. This set of "adjoining" units is determined by a kernel function $\phi(\mathbf{w}_a, \mathbf{w}_b, \sigma^t)$ called *neighborhood function* which tells for a codebook $\mathbf{w}_a$ vector whether it stays in the influence area of another codebook vector $\mathbf{w}_b$, with respect to an influence radius σ_t. Every $\mathbf{w}_i \in W$ is updated with the following rule : $\mathbf{w}_i^{t+1} \leftarrow \mathbf{w}_i^t + \eta^t \phi(\mathbf{w}_i^t, \mathbf{w}^\star, \sigma^t)(\mathbf{x}^t - \mathbf{w}^\star)$.

The result of the algorithm is the mapping of the grid of codebook vectors to the underlying structure of the input samples in a low-dimensional projection. At each step t, the similarity of a unit's codebook vector to the input sample defines that unit's *activation* a_i^t.

The SOM *output activity* o_i^t is the activity seen from modules following the SOM in the architecture, here the MLR module. It is defined by applying an *output function* $\Phi(\bullet)$ to the map activations (here, Φ and its parameters are the same for all units). The importance of this output function has been studied in [22]. In this article, because we use a cosine similarity, Φ is a power function for which each unit activity and defined as :

$$o_i^t = \Phi(a_i^t, p) = \{a_i^t\}^p$$

D. Linear regression model

Applied to to real-valued target functions, the linear regression model explains a random continuous dependent target variable y by a linear combination of K explanatory random independent known variables $x_1, x_2, \dots, x_K$. This model can be formally defined by:

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_K x_K + \epsilon$$

with $\beta_0, \beta_1, \dots, \beta_K$ some parameters and ϵ the normally distributed disturbance term and $x_0 = 1$.

If the data set consists of N pairs $(\mathbf{x}_i, y_i)$ with $\mathbf{x}_i = (x_0, x_1, \dots, x_K)^T$, the model looks for the optimal weight vector $\mathbf{w}^\star = (\beta_0^\star, \beta_1^\star, \dots, \beta_K^\star)$ in terms of the squared error (SE). Because we wish to work in an on-line fashion, we choose an iterative method to minimize the cost function SE($\bullet$). The Widrow-Hoff algorithm applies stochastic gradient descent techniques to the linear regression objective function. The error is not computed for all data points any more but for each individual example. It becomes, for the i-th update of the linear model weights:

$$\mathrm{SE}_i^{\text{on-line}}(\mathbf{w}) = \frac{1}{2}(h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2$$

with $h_{\mathbf{w}}(\mathbf{x}_n) = \mathbf{w}^T \mathbf{x}_n$ the value of the hypothesis function attached to a certain $\mathbf{w}$.

E. Data generation

During experiments, sensor values, target values and relations between them belong to two "families" of data generation : in one case (data generation Type 1) the system receives two values built by adding noise directly to a single random target value; in the other case (data generation Type 2), the system generates two independent random values, which are corrupted by noise, and the target value is a linear or non-linear combination of these two values.

1) Data generation Type 1: The first "family" of input data, so-called *data generation Type 1* follows the "classic" Bayesian framework : a single "true" value r that gives rise to several noisy sensor readings s_i. We suppose that the sensor readings s_i are generated independently from each other from a unique r value by additive Gaussian noise $\tilde{\epsilon}$.

$$\begin{aligned} r &\sim p^{\mathcal{U}}_{a,b}(x) \\ s_i &= r + \tilde{\epsilon}_i \\ \tilde{\epsilon} &\sim P^{\mathcal{N}}_{\mu,\sigma}(x) \end{aligned}$$

2) Data generation Type 2: A more realistic setting is where the sensory readings s_i and the underlying true value r are more tightly coupled. In this second "family" of input data, so-called *data generation type 2*, we suppose that the s_i are no longer class-conditionally independent and depend on r as well as each other. In other words, *all* of the s_i have to be considered simultaneously for inferring r. Formally, we express this by drawing the $\tilde{s}_i$ independently from a bounded uniform distribution, and then making r a function of the $\tilde{s}_i$: $r = f(\tilde{s}_1, \tilde{s}_2, \dots)$. Afterwards, the $\tilde{s}_i$ are subjected to additive Gaussian noise in order to produce the measured values s_i:

$$\begin{aligned} \tilde{s}_i &\sim p^{\mathcal{U}}_{a,b}(x) \\ r &= f(\tilde{s}_1, \tilde{s}_2, \dots) \\ \tilde{\epsilon}_i &\sim P^{\mathcal{N}}_{\mu,\sigma}(x) \\ s_i &= \tilde{s}_i + \tilde{\epsilon}_i \end{aligned}$$

F. Extending the classic Bayesian integration framework

In this section, we will present Bayesian multisensory fusion in a setting that is close to real-life operation. In particular, we will drop the common assumption of r being unbounded and introduce an interval $r \in [a, b]$ into the Bayesian calculations. In the most generic form, Bayesian sensor fusion aims to find the most probable value of r given the observations $\mathbf{s}$:

$$r^* = \text{arg max}_r p(r|\mathbf{s}) \sim \text{arg max}_r p(\mathbf{s}|r)p(r) \tag{1}$$

This amounts to a maximization problem, putting the first derivative with respect to r to 0 yields the necessary condition:

$$\begin{aligned} \partial_r p(\mathbf{s}|r)p(r) &= 0 \\ \Leftrightarrow \partial_r \left(p(\mathbf{s}|r)\right) p(r) + p(\mathbf{s}|r)\partial_s p(r) &= 0 \end{aligned} \tag{2}$$

Eqn. (2) has trivial solutions outside the interval $[a, b]$ where both $p(s)$ and $\partial_s p(r)$ vanish. However they *minimize* $p(\mathbf{s}|r)p(r)$ (inserting an appropriate r always gives a value of 0), and are thus excluded from our considerations.

If, however, a solution exists inside $[a, b]$, it must obey the simplified equation

$$\partial_s \left(p(\mathbf{s}|r)\right) = 0 \tag{3}$$

On the other hand, if eqn.(3) has a non-trivial solution outside the interval $[a, b]$ then it must be either $s = a$ or $s = b$, depending on which is closer, because the infinities in the derivatives of $p(r)$ achieve a "clamping" of obtained fusion results to the known interval $[a, b]$. This can be implemented very efficiently, without solving any equations at all, as a post-processing step of fusion.

Please bear in mind that we do not make any assumptions about the conditional distributions $p(\mathbf{s}|r)$ which may be defined by additive noise or other models, both of which we will discuss in next paragraphs.

1) Dealing with Data generation Type 2: The only tricky point consists here in computing the quantity $p(\mathbf{s}|r)$ required by eqn. (2). As the sensor measurements s_i no longer directly depend on r but on $\tilde{s}_i$, the calculation is a little more cumbersome, especially since the factorization $p(\mathbf{s}|r) = \Pi_i p(s_i|r)$ no longer holds. For a simplified setting of two sensors we obtain:

$$
\begin{aligned}
p(s_1 s_2|r) &= \int\int d\tilde{s}_1 d\tilde{s}_2 p(s_1 s_2|\tilde{s}_1\tilde{s}_2 r)p(\tilde{s}_1\tilde{s}_2|r) \\
&= \int\int d\tilde{s}_1 d\tilde{s}_2 p(s_1 s_2|\tilde{s}_1\tilde{s}_2)p(\tilde{s}_1\tilde{s}_2|r) \\
&= \int\int d\tilde{s}_1 d\tilde{s}_2 p(s_1|\tilde{s}_1)p(s_2|\tilde{s}_2)\delta(f(\tilde{s}_1,\tilde{s}_2)-r) \\
&= \int d\tilde{s}_1 p(s_1|\tilde{s}_1)p(s_2|f_{s_2}^{-1}(\tilde{s}_1,r)) \qquad (4)
\end{aligned}
$$

where the first transformation follows from the law of total probability: we insert a complete set of disjoint states $\tilde{s}_1\tilde{s}_2$. In the second line, the factor r has been removed from the conditional probability $p(s_1 s_2|\tilde{s}_1\tilde{s}_2 r)$ as it can be deduced from $\tilde{s}_1$ and $\tilde{s}_2$. Later, the conditional probability has been split as s_i depends only on $\tilde{s}_i$.

The function $f_{s_2}^{-1}(s_1,r)$ is the function obtained by solving $f(s_1,s_2,\dots)=r$ for s_2. The optimal fused value of r in the interval $[a,b]$ is obtained as before by maximizing eqn. (4). As the resulting expression is in general intractable analytically, we resort to numerical methods to solve it for s. It should be stressed that eqn. (4) can be derived for any number of sensors although its numerical solution will get more and more costly.

IV. Experiments

For all the experiments described in this section, the case two virtual sensors is considered. We assume that they both observe the same real phenomenon in the environment, which is represented by a single real number. We demonstrate the abilities of our generative-discriminative system by comparing its performances with optimal Bayesian fusion which is not learned but uses prior knowledge of noise type and properties [23] : the Bayesian optimal fusion weights are statistically relevant but, on the other hand, specifically need prior estimate of sensor variances. Our common representation space is able to represent the statistical properties of all signals sufficiently well to allow for correct fusion.

A. Set up

Whatever the type of data generation, each experiment we conduct in this section varies the variance of each artificial sensor and measures how efficient fusion can be for a given combination. For simplicity, we use two simulated sensors with five possible standard deviations : for the k-th sensor, $\sigma_k \in [0.01, 0.0575, 0.105, 0.1525, 0.2]$.

Each target and noisy sensors values are respectively spatially encoded in shape of a Gaussian blob located in a 1×200 bunch of pixels (this bunch represents the discrete range $[0;100]$) with an integrated margin $\delta = 3$ as described in Sec. III-B.

The common representation space takes the shape of a 20×20 grid. The 400 corresponding codebook vectors are initialized uniformly at random and then updated following Kohonen's learning algorithm with a Gaussian neighborhood function and iteration-indexed decreasing neighborhood radius and learning rate (both with exponential decay), as described in Sec. III-C.

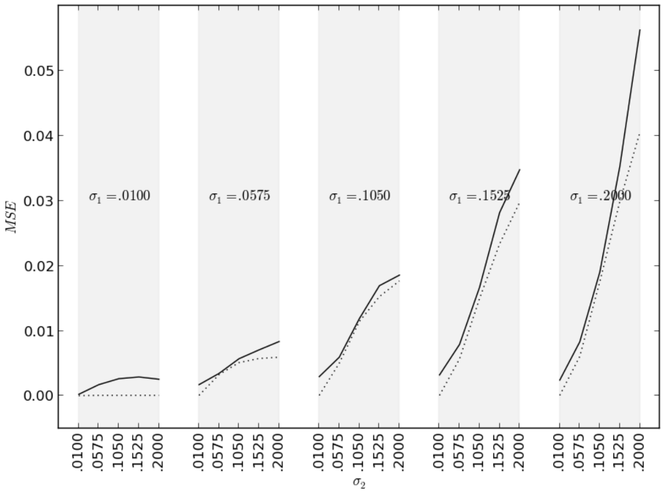

Fig. 2: Comparison of mean squared errors between our architecture (solid line –) and Bayesian (dotted line $\cdots$) fusion decisions with **data generation type 1** and **additive Gaussian noise**

For a given data generation type, for each of the 25 standard deviations couples and each of the 3 noise models, a SOM is trained during 100000 iterations on available spatially encoded training samples : at each step the SOM tries to map an input vector $\boldsymbol{\xi}^t$ which is the concatenation of noisy s_1^t and s_2^t, with $1 \leq t \leq 100000$.

After the first 100000 first steps, a MLR iteratively brings face to face the SOM output (with $p = 25$ for the power output function) with the spatially encoded target value. This interval of 100000 steps is not mandatory but allows a faster convergence of the MLR because it receives output from a quasi-stable learned organisation in the SOM. The MLR uses a learning rate $\alpha = 1 \times 10^{-4}$.

At the end, the MLR prediction abilities are test on 10000 never-seen sensor and target values with mean-squared error and compared with Bayesian (or extended Bayesian) predictions and MLR predictions based directly on encoded sensors couples.

B. Results

For data generated by ”data generation type 1”, with additive Gaussian noise for bounded target value, results show that the proposed architecture reaches classic and extended Bayesian frameworks results (Fig. 2). Because ”data generation type 1” deals with linear combinations of sensor values weighted by sensors noise variance, a single MLR module can achieve such performances, on condition that there are enough available training observations (we verified that MSE predictions are not significantly different from Bayesian predictions). The added SOM module with a non-linear output function does not damage the performances of the MLR module. Since results are near-optimal, we do not show the effect of a fusion decision modulation here.

For data generated by ”data generation type 2” (Fig. 3a and 3b), with strongly coupled sensors with additive Gaussian

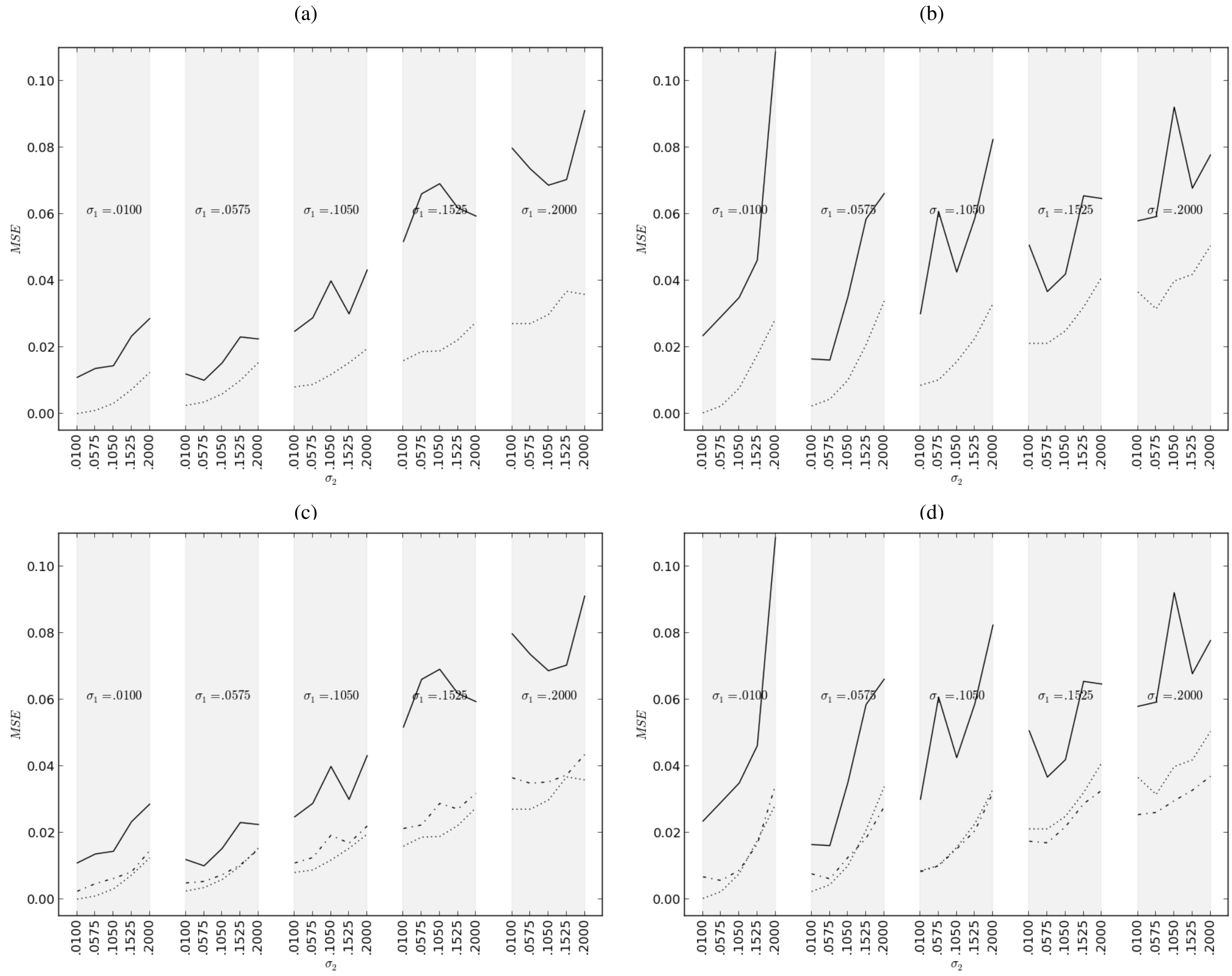

Fig. 3: Comparison of mean squared errors between our architecture (solid line –) and Bayesian (dotted line $\cdots$) fusion decisions with **data generation type 2** and **additive Gaussian noise**. **(a)** : targets values are linear combinations of sensors values ; **(b)** : targets values are non-linear combinations of sensors values ; **(c)**, respectively **(d)** : same as (a), respectively (b), and dash-dot lines (-.-) represent cases where a fusion decision is taken only if BMU scores are greater than the median of BMU scores of the trained grid.

noise, results show that the proposed architecture works far better than just on MLR module (note that, in these cases, MSE performances are out of range in presented figures). Target value and sensors values are not class-conditionally independent and there is no direct linear link between the latter and the former anymore. The learning phase provides a self-organized space which preserves encoded noisy sensors couples topological relationships and allows reduced representation of observations. The MLR module then receives a non-linear projection of the map activation – comparable to a kernel trick over cosine similarities – which allows a better prediction of the real target value.

Because the generative part of our system tries to represent data distribution statistics, it can calculate how probable an observation of encoded sensors values actually is. We apply the following simple heuristic to exclude outliers which do not permit sensible fusion results : the regression fusion decision is taken into account only if the current BMU score is greater than the median of past BMU scores. More complex heuristics could be used as thresholds to improve performance. But this kind of filter explicitly shows that the SOM module autonomously catches intrinsic variances of noisy sensors values, and that we can use this information to remove outliers. This very point is a huge advantage of the proposed architecture and improves results especially in case of realistic relationships between sensors values and true values (Fig. 3c and 3d).

V. Discussion

For a typical Bayesian approach, the distribution of the true stimulus, the noise models and the variances of the problem must be known beforehand in order to perform inference. In our approach, such parameters are implicitly estimated from data using a generative learning algorithm (SOM). Then, in virtually all related modeling approaches on multisensory fusion, it is assumed that the distribution of the underlying ”true” stimulus r, $p(r)$ is uniform and unbounded: $p(r) \sim \mathcal{U}(\infty, \infty)$, and that observations s_i for each sensor are obtained from r by adding Gaussian noise with a variance that is known for each sensor. Individual observations are therefore class-conditionally independent. In this article, we treat Bayesian multisensory integration in a way much closer to real experiments. We introduce a finite interval $[a, b]$ to which r is constrained which introduces additional complexity into the Bayesian formulas. Lastly, the models according to which observations s_i are generated from the true stimulus r are considerably more complex, and above all individual observations need no longer be class-conditionally independent.

When a dependant linear link between target value and sensors values exists (”data generation type 1”), our system is able to reach good performances – especially thanks to the MLR module – which are comparable to Bayesian optima. When this link is more realistic and potentially non-linear (”data generation type 2”), the MLR module needs to rely on the non-linear SOM output to achieve performance close to extended Bayesian framework.

We can significantly improve our system results by filtering fusion decisions with available quality measures based on BMU scores. Without this fusion decision modulation, the MLR module always combines sensors values, as the classic Bayesian fusion formula does. Without knowing explicit information about data distributions or noise variances, the proposed architecture can perform near-optimal performances on realistic fusion tasks.

VI. Future work

The hybrid architecture presented here still needs to have ground truths available for the discriminative part of it. A step towards more autonomy lies in using one cue as ground truth : the system could try to predict a sensor cue based on other sensors. Such a system also implies information feedback from the discriminative module to the generative one.

The median threshold used to detect outliers is currently offline and global : it consistently rejects half of the decisions making performance go better in average by combining BMU scores over all units of the SOM and must be computed between train and test phases. We need to investigate on-line and adaptive thresholding techniques and rely on topological localisation to refine this selection.

Finally, we want our approach to deal with incremental learning tasks and life-long learning, hopefully on ”real world” robotics datasets.

VII. Acknowledgements

We gratefully thank Mathieu Lefort for technical assistance and reviewing this article.

References

[1] G. A. Calvert and T. Thesen, “Multisensory integration: methodological approaches and emerging principles in the human brain,” *Journal of Physiology-Paris*, vol. 98, no. 1, pp. 191–205, 2004.

[2] D. Alais and D. Burr, “The ventriloquist effect results from near-optimal bimodal integration,” *Current biology*, vol. 14, no. 3, pp. 257–262, 2004.

[3] A. King, “Development of multisensory spatial integration,” 2004.

[4] M. Gori, M. Del Viva, G. Sandini, and D. C. Burr, “Young children do not integrate visual and haptic form information,” *Current Biology*, vol. 18, no. 9, pp. 694–698, 2008.

[5] M. O. Ernst and H. H. Bülthoff, “Merging the senses into a robust percept,” *Trends in cognitive sciences*, vol. 8, no. 4, pp. 162–169, 2004.

[6] D. E. Angelaki, Y. Gu, and G. C. DeAngelis, “Multisensory integration: psychophysics, neurophysiology, and computation,” *Current opinion in neurobiology*, vol. 19, no. 4, pp. 452–458, 2009.

[7] J. S. Kelso, *Dynamic patterns: The self-organization of brain and behavior*. MIT press, 1997.

[8] J. G. Martin, M. A. Meredith, and K. Ahmad, “Modeling multisensory enhancement with self-organizing maps,” *Frontiers in computational neuroscience*, vol. 3, 2009.

[9] T. J. Anastasio and P. E. Patton, “A two-stage unsupervised learning algorithm reproduces multisensory enhancement in a neural network model of the corticotectal system,” *The Journal of neuroscience*, vol. 23, no. 17, pp. 6713–6727, 2003.

[10] A. Pavlou and M. Casey, “Simulating the effects of cortical feedback in the superior colliculus with topographic maps,” in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1–8, IEEE, 2010.

[11] J. Mayor and K. Plunkett, “A neurocomputational account of taxonomic responding and fast mapping in early word learning.,” *Psychological review*, vol. 117, no. 1, p. 1, 2010.

[12] J. Bauer, C. Weber, and S. Wermter, “A som-based model for multisensory integration in the superior colliculus,” in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1–8, IEEE, 2012.

[13] A. Georgakis, H. Li, and M. Gordan, “An ensemble of SOM networks for document organization and retrieval,” in *Int. Conf. on Adaptive Knowledge Representation and Reasoning (AKRR’05)*, p. 6, 2005.

[14] V. Gliozzi, J. Mayor, J.-F. Hu, and K. Plunkett, “The impact of labels on visual categorisation: A neural network model,” 2008.

[15] B. Baruque and E. Corchado, “A bio-inspired fusion method for data visualization,” in *Hybrid Artificial Intelligence Systems*, pp. 501–509, Springer, 2010.

[16] T. Jantvik, L. Gustafsson, and A. P. Papliński, “A self-organized artificial neural network architecture for sensory integration with applications to letter-phoneme integration,” *Neural computation*, vol. 23, no. 8, pp. 2101–2139, 2011.

[17] A. Gepperth, B. Dittes, and M. G. Ortiz, “The contribution of context information: a case study of object recognition in an intelligent car,” *Neurocomputing*, vol. 94, pp. 77–86, 2012.

[18] A. Pouget, S. Deneve, and J.-R. Duhamel, “A computational perspective on the neural basis of multisensory spatial representations,” *Nature Reviews Neuroscience*, vol. 3, no. 9, pp. 741–747, 2002.

[19] A. Pouget, P. Dayan, and R. S. Zemel, “Inference and computation with population codes,” *Annual review of neuroscience*, vol. 26, no. 1, pp. 381–410, 2003.

[20] M. G. Ortiz, B. Dittes, J. Fritsch, and A. Gepperth, “Autonomous generation of internal representations for associative learning,” in *Artificial Neural Networks–ICANN 2010*, pp. 247–256, Springer, 2010.

[21] T. Kohonen, “Essentials of the self-organizing map,” *Neural Networks*, vol. 37, pp. 52–65, 2013.

[22] T. Hecht, M. Lefort, and A. Gepperth, “Using self-organizing maps for regression: the importance of the output function,” in *European Symposium On Artificial Neural Networks (ESANN)*, 2015.

[23] M. O. Ernst, “A bayesian view on multimodal cue integration,” *Human body perception from the inside out*, pp. 105–131, 2006.