PROJECTED BELIEF NETWORKS WITH DISCRIMINATIVE ALIGNMENT FOR CLASSIFYING MARINE MAMMALS

Paul M Baggenstoss

Fraunhofer FKIE, Fraunhoferstr 20,
53343 Wachtberg, Germany

ABSTRACT

Combining generative and discriminative classifiers has the potential to attain large performance improvements because they operate in fundamentally different ways, and tend to make independent errors. However this potential is difficult to realize because generative classifiers tend to perform poorly. In this paper, we create a generative classifier using a projected belief network (PBN) in conjunction with discriminative alignment (DA) that rivals a CNN and achieves significant performance improvements when combined. New classification experiments using marine mammal vocalizations are shown in which the error rate is cut in half by combining with a conventional CNN.

1. INTRODUCTION

1.1. Motivation of PBN-DA and Paper Contributions

Because generative (GEN) and discriminative (DISC) classifiers operate in fundamentally different ways, they tend to make independent errors and offer the potential to obtain greatly improved performance when combined. However, despite the advantages of GEN classifiers (data synthesis, robustness, detecting out-of-set events), they tend to perform poorly, making this potential difficult to achieve. A projected belief network (PBN) is a single network that can act as both a GEN and DISC classifier at the same time, and in conjunction with discriminative alignment (DA) allows the best of both GEN and DISC approaches to be realized. In this paper, we present new experimental results using PBN with DA (PBN-DA) in combination with a CNN for classfying marine mammal calls, cutting the error rate in half. We also provide new arguments and motivations for PBN and DA including new visualization experiments to illustrate DA.

1.2. Challenges in Designing Generative Classifiers

Creating a GEN classifer that rivals a DISC one is a major challenge. Model fitting is limited by the dimensionality curse, whereby model complexity and data requirements necessary for accurate fitting increase exponentially with data dimension [1]. However data size and model complexity are limited for practical reasons, forcing one to discard information in order to limit model complexity. Unfortunately, GEN classifiers tend to keep information that describes the data, but discard discriminative information.

Let $\mathbf{x}$ be the input data, and $\mathbf{y} \in \{\mathbf{y}_1, \mathbf{y}_2 \ldots\}$ be the class labels. Consider the following two ways to construct GEN classifiers: The **joint approach** first estimates $p(\mathbf{x}, \mathbf{y})$, then calculates $p(\mathbf{y}|\mathbf{x})$ using Bayes rule, whereas the **class-specific** approach separately estimates the class-conditional densities $p(\mathbf{x}|\mathbf{y}_1),\ p(\mathbf{x}|\mathbf{y}_2), \ldots$.

The class-specific approach has disadvantages: separately estimating each class distribution is computationally expensive and introduces imbalances due to separate random initialization and different tradeoffs in model overfitting/underfitting, causing classification errors. In contrast, the joint approach estimates $p(\mathbf{x}, \mathbf{y})$ at once, eliminating errors caused by imbalances. An example joint approach is the deep belief network (DBN) [2], which at the time had state of the art performance. The universal background model (UBM) is another attempt to reduce imbalances in speaker classification by starting with a single model trained on all the data, then adapting it for each class [3].

But, the joint approach also has disadvantages: it cannot be easily extended to new classes, and it is just one “expert”, whereas the class-specific approach is a team of experts that can model each class with more care and depth, so can be much more selective. For example, the convolutional kernels of a network trained separately on each class can work as basis functions to represent the data the corresponding class.

1.3. Benefit and Main Idea of PBN-DA

Using PBN-DA, one can create a better class-specific GEN classifier by reducing the effect of model imbalances and retaining discriminative information [4–6]. In DA, the shifting of the decision boundaries due to model imbalance is reduced by forcing the likelihood function (LF) to have a high slope in the directions orthogonal to the decision boundaries, so that model imbalances have less influence (see visualization in Section 3.2). This training also persuades the generative model to retain discriminative information.

This work was supported jointly by the Office of Naval Research Global and the Defense Advanced Research Projects Agency under Research Grant - N62909-21-1-2024

To achieve DA, a single network is trained with a joint cost function consisting of (a) a generative LF trained on just one data class, and (b) a DISC classifier cost (i.e. cross-entropy) trained to discriminate the given class from all other classes. For a single network to have such a joint cost function, it must operate in both the forward and backward direction, (a **two-directional network**). The forward direction operates as a standard feed-forward classifier network and the backward direction operates as a stochastic generative network with a tractable LF. We seek a two-directional network with the *exact inverse* property : initial variables passed backward through the generative path should produce input data that re-creates the same initial variables when passed forward. This property results in a tighter connection between the two directions, insuring that the decision boundaries of the classifier affect the shape of the generative likelihood function. The projected belief network (PBN) is a two-directional network with all these properties [5, 7–10].

A bank of PBNs is a *class-specific* GEN classifier type (as defined in Section 1.2). Each PBN is trained to act as a generative model for one class, and at the same time as classifier to discriminate between the given class and all other classes. This makes a given PBN an "expert" on just one class, knowing not only what describes the given class, but also what distinguishes it from the other classes.

1.4. Prior Work

Existing two-directional networks include restricted Boltzmann machine (RBM), constructed using back-to-back perceptron layers. The same network parameters are used in both directions [11]. Multi-layer stacked RBMs and can be jointly trained using the up-down algorithm [2]. However the RBM does not have a tractable LF and does not have the exact inverse property. The same can be said of auto-encoders constructed with tied reconstruction and analysis weights [12]. Normalizing flows (NF) [13], and the extension to dimension-changing networks called SurVAE [14] are two-directional networks with exact inverse property, but are largely a re-invention of probability density function (PDF) projection [15], the basis of PBN, which came over 20 years earlier [16, 17].

2. REVIEW OF PDF PROJECTION AND PBN

We provide now a short review of PBN and its theoretical basis in PDF projection, which is covered in depth in existing work [16, 18–20]. Consider a fixed and differentiable dimension-reducing transformation, $\mathbf{z} = T(\mathbf{x})$, where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^N$, and $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^M$, where $M < N$. We assume furthermore that the matrix of partial derivatives $\mathbf{D}_{i,j} = \left[\frac{\partial z_i}{\partial x_j}\right]$ has full rank. Assuming a known or assumed feature distribution $g(\mathbf{z})$, one can construct a PDF on the input data with support $\mathcal{X}$ given by

$$G(\mathbf{x}) = \frac{p_{0,x}(\mathbf{x})}{p_{0,x}(\mathbf{z})} g(\mathbf{z}), \quad \mathbf{z} = T(\mathbf{x}), \tag{1}$$

where $p_{0,x}(\mathbf{x})$ is a prior distribution and $p_{0,x}(\mathbf{z})$ is its mapping through $T(\mathbf{x})$ [1]. Note that (1) is a function of only $\mathbf{x}$ since $\mathbf{z}$ is deterministically determined from $\mathbf{x}$. It can be shown [16] that $G(\mathbf{x})$ is a PDF (integrates to 1) and is a member of the set of PDFs that map to $g(\mathbf{z})$ through $T(\mathbf{x})$. If $p_{0,x}(\mathbf{x})$ is selected for maximum entropy (MaxEnt), then $G(\mathbf{x})$ is unique for a given transformation, data range $\mathcal{X}$, and a given $g(\mathbf{z})$ (where "g" represents the "given" feature distribution) [18, 19]. To train the transformation, one maximizes the mean of $\log G(\mathbf{x})$ over a set of training data, and this results in a transformation that extracts sufficient statistics and maximizes information [20]. We say that $G(\mathbf{x})$ is the "projection" of $g(\mathbf{z})$ back to the input data range $\mathcal{X}$, i.e. a *back-projection.* To generate data from $G(\mathbf{x})$ in (1), one draws a sample $\mathbf{z}$ from $g(\mathbf{z})$, then draws a sample $\mathbf{x}$ from the set $\mathcal{M}(\mathbf{z})$, where

$$\mathcal{M}(\mathbf{z}) = \{\mathbf{x} \in \mathcal{X} | T(\mathbf{x}) = \mathbf{z}\}, \tag{2}$$

probabilistically weighted by the prior distribution $p_{0,x}(\mathbf{x})$, i.e. we sample from $p_{0,x}(\mathbf{x})$ restricted to $\mathcal{M}(\mathbf{z})$.

Chain-rule. For cascaded transformations, we apply the chain rule by recursively applying (1). Consider a cascade of two transformations, $\mathbf{y} = T_1(\mathbf{x})$, and $\mathbf{z} = T_2(\mathbf{y})$. Applying (1) to the first transformation, we have $G(\mathbf{x}) = \frac{p_{0,x}(\mathbf{x})}{p_{0,x}(\mathbf{y})} g(\mathbf{y})$. Applying to the second, we have $G(\mathbf{y}) = \frac{p_{0,y}(\mathbf{y})}{p_{0,y}(\mathbf{z})} g(\mathbf{z})$. We then just substitute $G(\mathbf{y})$ for $g(\mathbf{y})$, resulting in

$$G(\mathbf{x}) = \frac{p_{0,x}(\mathbf{x})}{p_{0,x}(\mathbf{y})} G(\mathbf{y}) = \frac{p_{0,x}(\mathbf{x})}{p_{0,x}(\mathbf{y})} \frac{p_{0,y}(\mathbf{y})}{p_{0,y}(\mathbf{z})} g(\mathbf{z}), \tag{3}$$

which can be extended to any number of stages. To compute $\log G(\mathbf{x})$, one just accumulates the contibutions of each layer. Data generation is also cascaded, and is initiated by drawing a sample $\mathbf{z}$ from $g(\mathbf{z})$. When the chain rule is applied to a feed-forward neural network (FFNN) layer-by-layer, this results in the projected belief network (PBN) [8].

3. DISCRIMINATIVE ALIGNMENT (DA)

3.1. Cost Function

DA is achieved by training each class-dependent PBN to minimize the joint cost function $C_m(\mathbf{x}_i) = \alpha d_m(\mathbf{x}_i) - \delta[c(i) - m] \log G(\mathbf{x}_i)$, where m is the class index $1 \leq m \leq M$, $G(\mathbf{x})$ is the generative LF (3), $d_m(\mathbf{x})$ is the classifier cost function for the binary classification between class m and all other

[1] In our notation, the argument of the distribution defines its range of support, and the variable in the subscript defines the original range where the distribution was defined. Thus, $p_{0,x}(\mathbf{z})$ is a distribution with support on $\mathcal{Z}$, but is a mapping of a distribution that was defined on $\mathcal{X}$.

classes joined together, data index i ranges over all the training data (i.e. data from all M classes), $c(i)$ is the data class label for data sample i, $\delta[c-m]$ is the indicator function equal to 1 when $c = m$ and zero otherwise, and α is a constant. By pre-multiplying with $\delta[c(i) - m]$, $G(\mathbf{x}_i)$ is trained on only data from class m , but $d_m(\mathbf{x}_i)$ is trained on all classes.

3.2. Visualization of DA

To visualize DA, we trained a simple PBN network on two-dimensional data and two data classes. In Figure 1, on the top two rows, from left to right, we see data from both classes, an intensity plot of the PBN LF after training on one class, and the corresponding likelihood contours. As we would expect, the LF has a peak at the location of the data on which it was trained. However, as can be seen by the contour plots, there is not much selectivity against the other data class.

We then re-trained the two PBNs using DA. In rows 3 and 4 of the figure, we see the results. Now the contour plots show high selectivity against the other data class (i.e. high slope in the direction that separates the two data classes). The contours have been "discriminatively aligned". Now, when classifying class 1 versus class 2 using a straight Bayesian likelihood classifier, the classification results will resemble the properties of the discriminative classifier.

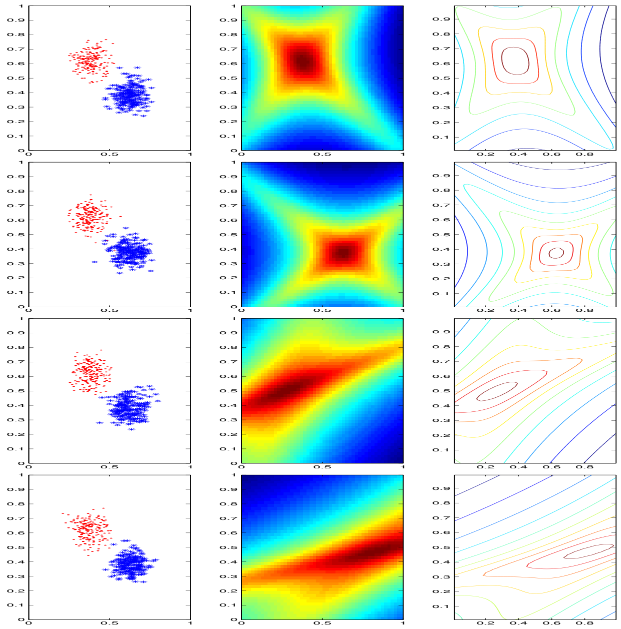

Fig. 1. From top to bottom: PBN trained on class 1 (red), PBN trained on class 2 (blue), PBN trained on class 1 with DA, PBN trained on class 2 with DA. Left column: input data, center: LF surface, right: contour lines of LF surface.

3.3. PBN-DA-HMM

PBN-DA-HMM is an extension of PBN-DA, in which a hidden Markov model (HMM) is integrated into the PBN [6]. After training a PBN using DA, the network is split and the HMM is used in place of the second half. The principle can be understood using the example in Section 2. The HMM takes the place of $G(\mathbf{y})$ in equation (3), and the second half of the network is discarded. As a result of the DA training, the features seen by the HMM carry discriminative information.

4. EXPERIMENTAL RESULTS

4.1. Computational Load and Limitation of PBN

The high computational load of PBN can be mitigated using proper network design [5], but despite this, increases linearly with the number of classes, making PBN only suitable for modest-sized classification experiments. When there is a high cost of error, such as in military and public safety applications, PBN is cost effective since it can drastically reduce the error rate.

4.2. Past Experiments

PBN with DA (with and without HMM) has been proven in numerous experiments in comparison to and in combination with a conventional CNN. Table 1 lists previously published results, including results from this paper. The error reduction factor shows how PBN-DA compares with CNN in terms of relative number of errors. A reduction factor of 1.0 means equal performance to CNN, in itself very meaningful for the reasons indicated in Section 1.2. A factor less than 1.0 indicates an improvement over CNN. When a linear combination of the output statistics is used (PBN-DA+CNN), an improvement is always seen. In two of the data sets, a straight class-specific generative PBN classifier (without DA) was also tested, demonstrating the advantage of DA.

4.3. Acoustic Trends Blue Fin Data Set

4.3.1. Data Description

The Australian Acoustic Trends Blue Fin data set [22] consists of acoustic recordings from various hydrophones, along with a set of annotations of marine mammal vocalizations that describe the bounding boxes of each vocalization (start and end time, as well as start and end frequency). We collected 200 examples of from each of six call types, denoted by "BM-Ant-A ", "BM.Ant-B ", "BM.Ant-Z ", "Bm.D ", "Bp-20Hz", "Bp-Downsweep". An example annotation is shown in Figure 2 from class "Bm-D". Although we rejected annotations where the signal of interest was not visible to the eye, the data set contains many samples that are weak and overlap with interfering noise and calls. A time-window of 3072 samples at 250 Hz sample rate (12 seconds) was extracted for each selected annotation. For the current experiments, the time-series were converted to log-band energy features with the

						Reduction Factor		
Ref	Data Set	Class	Dim.	Samp.	HMM	PBN	PBN-DA	PBN-DA+CNN
[4]	Subset of Google Keywords	2	900	500	N	1.48	1.17	0.84
[4]	Subset of MNIST (handwritten char.)	3	196	500	N	2.2	1.22	0.89
[5]	Office Sounds (acoust. events)	6	24318	102	N	-	1.0	0.55
[6]	Subset of ESC-50 (acoust. events)	8	29952	40	Y	-	0.94	0.51
[21]	Subset of ESC-50 (acoust. events)	23	29952	40	Y	-	1.01	0.51
-	Acoustic Trends (marine mammal)	6	960	200	Y	-	**0.84**	**0.54**

Table 1. PBN-DA experiments. We list the relevant reference, the number of data classes, the data dimension, the number of training samples per class, if HMM was used, and the error reduction factor with respect to CNN for PBN, PBN-DA , and for PBN-DA in combination with CNN. The last entry is for current paper in which PBN-DA alone performed better than CNN.

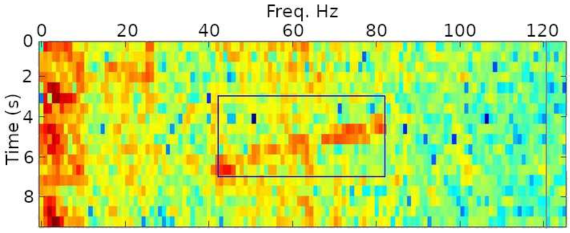

Fig. 2. Example annotation from Acoustic Trends Blue Fin Data Set of type “Bm-D”. Bounding box derived from expert annotation.

following parameters: FFT size 384 with 128-sample shift (2/3 overlap), 40 linear-spaced hanning-weighted frequency bands, resulting in a 40 × 24 feature map per event. Extracted .wav files and feature values for the selected annotations have been made available online [23].

4.4. Networks

For the PBNs, a 5-layer network was used, consisting of two convolutional layers, followed by dense layers of 128, 32, and 6 nodes. The first layer had 8 7 × 13 kernels with 2 × 4 downsampling, resulting in 8 output maps of 12 × 10 (dimensions always given in time×freq). The second layer had 48 4 × 10 kernels with 2 × 1 downsampling, resulting in 48 output maps of 5 × 1 (dimensions always given in time×freq). Linear activation was used at the output of the convolutional layers, and truncated Gaussian (TG) activation function was used at the output of dense layers. TG is the maximum entropy activation function for the truncated Gaussian prior in PBNs [7]. For the benchmark CNN, the same network structure was used, max-pooling was used instead of downsampling, TG activation functions were used at the outputs of all layers, and dropout regularization was used.

To implement PBN-DA-HMM, we tapped the output of the second convolutional layer, making a 5 × 48 feature map, where the first dimension was time. We used 4-state HMM, where each state used a Gaussian mixture (GMM) of 3 components. A value of 0.12 was added to the diagonal elements of the GMM covariance matrices.

4.5. Training

Data was split into four 150/50 random data folds. Six separate PBNs were trained (one on each data class) using discriminative alignment. For PBN and CNN, we used data random augmentation of +/ − 3 samples (max) circular time shifts and +/ − 1 sample (max) frequency shift.

4.6. Results

Results are shown in Figure 3 for PBN-DA (in red) and PBN-DA-HMM (in blue) as a function of linear combination factor (when adding the output statistic of the benchmark CNN), and averaged over the data folds. Shown is total errors out of a total of 300 for each data fold. PBN-DA alone performed well, but not as well as the benchmark CNN, and resulted in significant error reduction when combined with the CNN. PBN-DA-HMM performed exceedingly well, better than the benchmark CNN, and resulted in a very significant 2:1 error reduction.

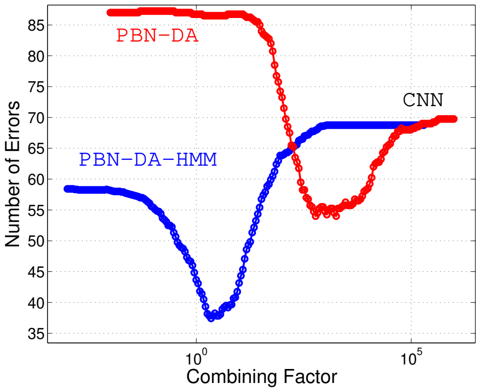

Fig. 3. Results of PBN-DA vs. PBN-DA-HMM in combination with CNN for Australian Blue Fin data. Shown is number of errors out of 300 averaged over the four data folds.

Data as well as a software toolkit to implement the experiments has been made available online [24].

5. CONCLUSIONS

In this paper, we have presented new arguments and a visualization experiment that explain why DA improves generative classifiers. We reviewed past experiments and provided a new classification experiment using marime mammal calls, showing that in combination with a CNN, PBN-DA-HMM cut the error rate in half.

6. REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning*. Springer, 1999.

[2] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," in *Neural Computation 2006*, 2006.

[3] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[4] P. M. Baggenstoss, "Discriminative alignment of projected belief networks," *IEEE Signal Processing Letters*, Sep 2021.

[5] ——, "Using the projected belief network at high dimensions," *Proceedings of EUSIPCO 2022, Belgrade*, 2022.

[6] P. M. Baggenstoss and K. Wilkinghoff, "Novel generative classifier for acoustic events (accepted)," *Proceedings of EUSIPCO 2023, Helsinki*, 2023.

[7] P. M. Baggenstoss, "A neural network based on first principles," in *ICASSP 2020, Barcelona (virtual)*, Barcelona, Spain, Sep 2020.

[8] ——, "On the duality between belief networks and feed-forward neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2018.

[9] ——, "Applications of projected belief networks (pbn)," in *Proceedings of EUSIPCO 2019*, La Coruña, Spain, Sep 2019.

[10] ——, "The projected belief network classifier: both generative and discriminative," *Proceedings of EUSIPCO, Amsterdam*, 2020.

[11] M. Welling, M. Rosen-Zvi, and G. Hinton, "Exponential family harmoniums with an application to information retrieval," *Advances in neural information processing systems*, 2004.

[12] P. Li and P. Nguyen, "On random deep weight-tied autoencoders: Exact asymptotic analysis, phase transitions, and implications to training," *ICLR*, 2019.

[13] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3964–3979, 2021.

[14] D. Nielsen, P. Jaini, E. Hoogeboom, O. Winther, and M. Welling, "Survae flows: Surjections to bridge the gap between vaes and flows," in *NIPS 2020 (Virtual)*, 2020.

[15] P. M. Baggenstoss and F. Govaers, "A comparison of PDF projection with normalizing flows and SurVAE," *arXiv*, 2023.

[16] P. M. Baggenstoss, "The PDF projection theorem and the class-specific method," *IEEE Trans Signal Processing*, pp. 672–685, March 2003.

[17] ——, "A theoretically optimum approach to classification using class-specific features." *Proceedings of ICPR, Barcelona*, 2000.

[18] ——, "Beyond moments: Extending the maximum entropy principle to feature distribution constraints," *Entropy*, vol. 20, no. 9, 2018. [Online]. Available: http://www.mdpi.com/1099-4300/20/9/650

[19] ——, "Maximum entropy PDF design using feature density constraints: Applications in signal processing," *IEEE Trans. Signal Processing*, vol. 63, no. 11, Jun. 2015.

[20] P. M. Baggenstoss and S. Kay, "Nonlinear dimension reduction by pdf estimation," *IEEE Transactions on Signal Processing*, 2022.

[21] P. M. Baggenstoss, K. Wilkinghof, F. Govaers, and F. Kurth, "Projected belief networks with discriminative alignment for acoustic event classification: Rivaling state of the art cnns," *arXiv*, 2024.

[22] B. Miller, K. Stafford, I. Van Opzeeland, D. Harris, F. Samaran, A. šIrović, S. Buchan, K. Findlay, N. Balcazar, S. Nieukirk, E. Leroy, M. Aulich, F. Shabangu, R. Dziak, W. Lee, and J. Hong, "An annotated library of underwater acoustic recordings for testing and training automated algorithms for detecting antarctic blue and fin whale sounds," in *Australian Antarctic Data Centre*, 2020. [Online]. Available: https://data.aad.gov.au/metadata/records/AcousticTrends_BlueFinLibrary

[23] P. Baggenstoss, "Selected events from acoustic trends blue fin data set," accessed: 2023-10-28. [Online]. Available: http://class-specific.com/au6

[24] ——, "PBN Toolkit," accessed: 2023-10-28. [Online]. Available: http://class-specific.com/pbntk