Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

Preliminary Investigation of Boltzmann Machine Classifiers for Speaker Recognition

Themos Stafylakis
Patrick Kenny
Mohammed Senoussaoui
Pierre Dumouchel

Centre de recherche informatique de Montréal (CRIM) and École de technologie supérieure (ETS)
First.Last@crim.ca

Abstract

We propose a novel generative approach to speaker recognition using Boltzmann machines, a fledgeling non-Gaussian probabilistic framework that is increasingly gaining attention in several machine learning fields. We show how a modified i-vector representation of speech utterances enables the development of several Boltzmann machine architectures for speaker verification and we report some preliminary speaker recognition results obtained with one of them, which we refer to as Siamese twins. The Siamese twin architecture is designed to capture correlations between utterances spoken by a single speaker and it can be regarded as probabilistic analogue of the well known cosine distance metric. A relative improvement of 27% is reported on NIST-2010 telephone female data.

1. Introduction

Boltzmann machines are probability distributions on high dimensional binary vectors which are analogous to Gaussian Markov Random Fields in that they are fully determined by first and second order moments. A key difference however is that augmenting Boltzmann machines with hidden variables enlarges the class of distributions that can be modeled, so that in principle it is possible to model distributions of arbitrary complexity [1]. (On the other hand, marginalizing over hidden variables in a Gaussian distribution merely gives another Gaussian.) A variational Bayes expectation maximization algorithm has been developed for training Boltzmann machines which is reasonably efficient for a class of sparsely connected Boltzmann machines that includes the deep Boltzmann machines studied in [2]. The binary/Gaussian distinction is not an exclusive dichotomy: hybrid models containing both types of hidden variable can be constructed. This enables Boltzmann machines to model continuous data vectors such as acoustic observation vectors in speech recognition or i-vectors in speaker recognition.

Readers familiar with the machine learning literature will be aware that Boltzmann machines are principally used in unsupervised training of another type of generative model known as a deep belief network which serves to initialize backpropagation training of discriminative neural networks [1]. These neural networks have recently proved to be very successful in speech recognition [3, 4, 5, 6, 7, 8] so the question naturally arises whether such an approach can be made to work in speaker recognition. However that is not the question that we will attempt to address here.

State of the art speaker recognition systems use several types of generative model as feature extractors (the Universal Background Model, Joint Factor Analysis and the i-vector extractor) and as classifiers (Probabilistic Linear Discriminant Analysis and the cosine distance metric) [9, 10, 11]. (It is only with the advent of i-vectors that discriminative approaches have begun to have an impact [12, 13].) These generative models rely heavily on Gaussian assumptions (some of which are quite questionable [11]) so there is reason to believe that modeling with Boltzmann machines may eventually prove to be more powerful. In the long run we aim to devise a complete speaker recognition architecture in this way. In this paper, we will describe a first step we have taken in this direction and explain how we have used the Boltzmann machine apparatus to build a classifier for speaker verification.

2. Boltzmann Machines

A Boltzmann machine is a probability distribution on binary vectors $\boldsymbol{x}$ of the form

$$P(\boldsymbol{x}) \quad = \quad \frac{1}{Z} e^{-E(\boldsymbol{x})}$$

where the "energy function" $E(\boldsymbol{x})$ has the form

$$E(\boldsymbol{x}) \quad = \quad -\sum_{i<j} x_i w_{ij} x_j,$$

the sum extending over all pairs (i, j) such that $i < j$. The normalizing constant Z is referred to as the partition function.

Units are sometimes referred to as neurons and the weights w_{ij} as synaptic weights. We denote the weight matrix by $\boldsymbol{W}$. This is assumed to be symmetric with zero diagonal (to prevent neurons from interacting with themselves). Some of the units may be distinguished as visible and others as hidden. When it is necessary to make this distinction we will use the symbol v_i for the binary variable associated with a visible unit i and the symbol h_j for the binary variable associated with a hidden unit j.

First order terms such as $\sum_j w_j x_j$ could also be included but there is no gain in generality as this is equivalent to forcing some of the components of $\boldsymbol{x}$ to be 1. We will exclude these "bias" terms in order to keep the notation simple.

2.1. Gibbs sampling and the mean field approximation

The Gibbs sampling formulas shows that a Boltzmann machine can be regarded as a stochastic neural network (for background

on variational Bayes, expectation maximization and Gibbs sampling see [14]).

$$\begin{aligned} Q(x_i = 1|\boldsymbol{x}_{\setminus i}) &= \frac{P(x_i = 1, \boldsymbol{x}_{\setminus i})}{P(x_i = 0, \boldsymbol{x}_{\setminus i}) + P(x_i = 1, \boldsymbol{x}_{\setminus i})} \\ &= \frac{\exp\left(\sum_{j \neq i} w_{ij} x_j\right)}{1 + \exp\left(\sum_{j \neq i} w_{ij} x_j\right)} \\ &= \sigma\left(\sum_{j \neq i} w_{ij} x_j\right) \end{aligned}$$

where σ is the sigmoid (that is, S-shaped) non-linearity defined by

$$\sigma(u) = (1 + e^{-u})^{-1}.$$

In order to convert the Boltzmann machine into a deterministic neural network we can apply variational Bayes to the prior distribution $P(\boldsymbol{x})$, that is, we calculate a variational approximation

$$\begin{aligned} P(\boldsymbol{x}) &\approx \prod_i Q(x_i) \\ &= \prod_i \mu_i^{x_i} \end{aligned}$$

where $\mu_i = Q(x_i = 1)$. (In the context of Boltzmann machines, variational Bayes is usually referred to as the mean field approximation.) The variational update equations are

$$\begin{aligned} \ln Q(x_i) &\equiv E_{Q(\boldsymbol{x}_{\setminus i})}\left[\ln P(x_i, \boldsymbol{x}_{\setminus i})\right] \\ &= E_{Q(\boldsymbol{x}_{\setminus i})}\left[\sum_{j \neq i} x_i w_{ij} x_j\right] \\ &= \begin{cases} \sum_{j \neq i} w_{ij} \mu_j & \text{if} \quad x_i = 1 \\ 0 & \text{if} \quad x_i = 0 \end{cases} \end{aligned}$$

so

$$\begin{aligned} Q(x_i = 1) &= \frac{\exp\left(\sum_{j \neq i} w_{ij} \mu_j\right)}{1 + \exp\left(\sum_{j \neq i} w_{ij} \mu_j\right)} \\ &= \sigma\left(\sum_{j \neq i} w_{ij} \mu_j\right) \end{aligned}$$

Since $\mu_i = Q(x_i = 1)$, the fixed point equations are

$$\mu_i = \sigma\left(\sum_{j \neq i} w_{ij} \mu_j\right).$$

These are the same as the equations for Gibbs sampling except that the x's are replaced by their mean values, the μ's. This explains the the term "mean field" and shows how the Boltzmann distribution can be approximated by a deterministic neural net.

Note that the mean field approximation is *not* a good approximation to $P(\boldsymbol{x})$ since it treats the units as statistically independent and the approximate distribution is unimodal. In practice, variational Bayes should only be applied to posterior distributions (where the values of some of the components of $\boldsymbol{x}$ are fixed), not prior distributions.

2.2. Role of hidden units

Both Boltzmann machines and Gaussian Markov random fields can model only the second order statistics of data. Perhaps the most interesting difference between the two is that including hidden variables in Boltzmann machines extends the class of distributions that can be modeled but this is not the case for Gaussian Markov random fields. If $P(\boldsymbol{v}, \boldsymbol{h})$ is a Gaussian Markov random field then the marginal distribution of $P(\boldsymbol{v})$ is just a Gaussian. On the other hand if $P(\boldsymbol{v}, \boldsymbol{h})$ is a Boltzmann distribution with energy function $E(\boldsymbol{v}, \boldsymbol{h})$ we can represent the marginal distribution $P(\boldsymbol{v})$ as an energy based model with energy function $F(\boldsymbol{v})$, that is

$$P(\boldsymbol{v}) = \frac{1}{Z} e^{-F(\boldsymbol{v})},$$

by defining the *free energy* $F(\boldsymbol{v})$ as

$$F(\boldsymbol{v}) = -\ln \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})}.$$

It is clear the the free energy cannot be represented as a quadratic form in $\boldsymbol{v}$ so introducing hidden variables extends the class of distributions that can be modeled by Boltzmann machines. It appears that by adding sufficiently many hidden variables *any* distribution on discrete binary vectors can be accommodated [1]. For a concrete example of a distribution that can be modeled by a Boltzmann machine with hidden variables but not by a Boltzmann machine, see [15].

2.3. Training

Boltzmann machines and related models are traditionally trained by stochastic gradient ascent rather than in batch mode as is usual in speech processing. To begin with, consider the case where there are no hidden units. Given a training token $\boldsymbol{x}^1$, a straightforward calculation gives the gradient of the log likelihood with respect to the model parameters:

$$\left.\frac{\partial \ln P(\boldsymbol{x})}{\partial w_{ij}}\right|_{\boldsymbol{x}=\boldsymbol{x}^1} = x_i^1 x_j^1 - \langle x_i x_j \rangle_{\text{model}}$$

where

$$\langle x_i x_j \rangle_{\text{model}} = \sum_{\boldsymbol{x}} P(\boldsymbol{x}) x_i x_j$$

Similarly, if there are N training tokens $x^1, \ldots, x^N$,

$$\frac{1}{N} \frac{\partial \ln P(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N)}{\partial w_{ij}} = \langle x_i x_j \rangle_{\text{data}} - \langle x_i x_j \rangle_{\text{model}}$$

where

$$\langle x_i x_j \rangle_{\text{data}} = \frac{1}{N}\left(x_i^1 x_j^1 + \ldots + x_i^N x_j^N\right).$$

We will refer to the correlations $\langle x_i x_j \rangle_{\text{data}}$ and $\langle x_i x_j \rangle_{\text{model}}$ as the *data statistics* and the *model statistics*. These have to agree at a critical point of the log likelihood function, just as the analogy with a zero mean Gaussian Markov random field would suggest.

In practice, training is implemented sequentially and the model is updated after each training token or mini-batch is presented. Specifically, if a training token $\boldsymbol{x}^1$ is presented, the model is updated according to

$$w_{ij} \leftarrow w_{ij} + \alpha \left.\frac{\partial \ln P(\boldsymbol{x})}{\partial w_{ij}}\right|_{\boldsymbol{x}=\boldsymbol{x}^1}$$

where α is a learning rate.

2.3.1. Contrastive divergence

The model statistics $\langle x_i x_j \rangle_{\text{model}}$ are typically estimated by *contrastive divergence* (CD) which in the general formulation given by Bengio and Dellaleau [16] can be applied using any MCMC algorithm to simulate the model (not just Gibbs sampling). Starting at the given training token $\boldsymbol{x}^1$, run the Markov chain for n steps: $\boldsymbol{x}^1 \rightarrow \ldots \rightarrow \boldsymbol{x}^{n+1}$. If n is sufficiently large then $\boldsymbol{x}^{n+1}$ will be approximately distributed according to the model distribution $P(\boldsymbol{x})$ so

$$E\left[x_i^{n+1} x_j^{n+1}\right] \approx \langle x_i x_j \rangle_{\text{model}}$$

and we can approximate

$$\left.\frac{\partial \ln P(\boldsymbol{x})}{\partial w_{ij}}\right|_{\boldsymbol{x}=\boldsymbol{x}^1} \approx x_i^1 x_j^1 - x_i^{n+1} x_j^{n+1}.$$

This approximation is known as CD-n. Surprisingly CD-1 works well in practice. CD-1 acts so as to depress the energy surface at the real datum $\boldsymbol{x}^1$ and increase the energy at a nearby fictitious datum $\boldsymbol{x}^2$. It can be argued that the net effect is to fit the energy surface closely to the real data points.

2.3.2. Persistent contrastive divergence

A set of samples (or "particles") $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N$ drawn from the model distribution is maintained and updated whenever the model is updated. (N Markov chains are run in parallel and, on every update, several steps of Gibbs sampling are performed in each chain.) A small learning rate ensures that the samples are always drawn from the model distribution even though the model is constantly being updated. The model statistics are derived by averaging over the particles:

$$\langle x_i x_j \rangle_{\text{model}} = \frac{1}{N} \sum_n x_i^n x_j^n$$

Persistent CD generally works better than CD-1.

2.3.3. Training with hidden units

To train Boltzmann machines with hidden units, the variational Bayes version of the EM algorithm is used [2]. Given a data vector $\boldsymbol{v}$, the variational lower bound is

$$\langle \ln P(\boldsymbol{x}) \rangle_{\text{data}} + H$$

where $\boldsymbol{x} = (\boldsymbol{v}, \boldsymbol{h})$ and $\langle \cdot \rangle_{\text{data}}$ refers to the expectation calculated with the posterior of $\boldsymbol{h}$ given $\boldsymbol{v}$ and H is the entropy of this posterior. We seek to maximize this with respect to the model parameters (so that the entropy term can be ignored).

Differentiating with respect to w_{ij},

$$\frac{\partial}{\partial w_{ij}} \langle \ln P(\boldsymbol{x}) \rangle_{\text{data}} = \langle x_i x_j \rangle_{\text{data}} - \langle x_i x_j \rangle_{\text{model}}.$$

The fact that not all of the units are visible makes no difference to the way the model statistics are handled and the only difference is in the treatment of the data statistics. In the case of a restricted Boltzmann machine (described in Section 3.1 below), the posterior $Q(\boldsymbol{h}|\boldsymbol{v})$ can be evaluated exactly and calculating the data statistics is straightforward. In the general case, either Gibbs sampling or variational Bayes is used to calculate the posterior expectation needed to evaluate the data statistics. Variational Bayes is used in practice as Gibbs sampling is too slow.

Variational Bayes could be computationally expensive as it may be necessary to cycle through all of the variables many times to achieve convergence. A standard trick which alleviates this problem is to initialize the variational Bayes updates for a given training token using the variational approximation calculated the last time the token was visited in the course of training.

2.4. Variational lower bound

The variational lower bound can be evaluated if the partition function Z is known. For each unit i, visible or hidden, set

$$\mu_i = Q(x_i = 1|\boldsymbol{v})$$

so that the expectation of x_i calculated with the variational posterior just μ_i and

$$\begin{aligned} \langle x_i x_j \rangle &= \langle x_i \rangle \langle x_j \rangle \\ &= \mu_i \mu_j \end{aligned}$$

if $i \neq j$. (If the variational posterior factorizes fully, x_i and x_j are independent in the posterior.) Then the variational lower bound is given by

$$\begin{aligned} &\langle \ln P(\boldsymbol{x}) \rangle + H \\ &= -\langle E(\boldsymbol{x}) \rangle - \ln Z + H \\ &= \sum_{i<j} \mu_i w_{ij} \mu_j - \ln Z \\ &\quad - \sum_i \left(\mu_i \ln \mu_i + (1-\mu_i) \ln(1-\mu_i)\right). \end{aligned}$$

If the partition function is not known, this calculation gives a lower bound on the free energy (which is actually enough for our purposes).

3. The Markov property

The most interesting situation is when the matrix $\boldsymbol{W}$ is sparse. Let $\boldsymbol{x}_{\setminus\{i,j\}}$ denote the set of variables other than x_i and x_j. If $\boldsymbol{x}_{\setminus\{i,j\}}$ is given and $w_{ij} = 0$ can write the energy function as

$$E(\boldsymbol{x}) = ax_i + bx_j + c$$

where a, b and c depend on $\boldsymbol{x}_{\setminus\{i,j\}}$ but not on x_i or x_j. Thus x_i and x_j are conditionally independent if $\boldsymbol{x}_{\setminus\{i,j\}}$ is given, provided that $w_{ij} = 0$.

A Boltzmann machine is represented by a graphical model with *undirected* edges joining all pairs of units i, j for which w_{ij} is *not* zero. The conditional independence property can be generalized as follows. Suppose we are given a partition of the units into three subsets A, B and C with the property that there is no edge between units in A and C. Then $\boldsymbol{x}_A$ and $\boldsymbol{x}_C$ are conditionally independent if $\boldsymbol{x}_B$ is given where $\boldsymbol{x}_A = \{x_i : i \in A\}$ etc. This is called the Markov property: the "future" (C) is independent of the "past" (A) if the "present" (B) is given.

3.1. Restricted Boltzmann machines (RBMs)

In this case there is a hidden layer and a visible layer with no hidden-to-hidden or visible-to-visible connections. The vectors $\boldsymbol{h}, \boldsymbol{v}$ are of dimension $J \times 1$ and $I \times 1$ and $\boldsymbol{W}$ is of dimension $I \times J$. The energy function is

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{v}^T \boldsymbol{W} \boldsymbol{h}.$$

We denote the ith row of $\boldsymbol{W}$ by $\boldsymbol{W}_{i\cdot}$ and the jth column by $\boldsymbol{W}_{\cdot j}$. By the Markov property, $Q(\boldsymbol{h}|\boldsymbol{v})$ and $P(\boldsymbol{v}|\boldsymbol{h})$ both factorize

$$\begin{aligned} Q(\boldsymbol{h}|\boldsymbol{v}) &= \prod_j Q(h_j|\boldsymbol{v}) \\ P(\boldsymbol{v}|\boldsymbol{h}) &= \prod_i P(v_i|\boldsymbol{h}). \end{aligned}$$

It turns out that each of the factors here as well as the free energy can be evaluated in closed form by a deft application of the distributive law of arithmetic [1]. Thus there is no need for variational Bayes and Gibbs sampling can be implemented efficiently by alternating between the hidden and visible levels. This is known as *block Gibbs sampling*.

The free energy works out to be

$$-\ln \prod_{j=1}^{J} \left(1 + \exp\left(\boldsymbol{v}^T \boldsymbol{W}_{\cdot j}\right)\right).$$

As for the posteriors,

$$Q(h_j = 1|\boldsymbol{v}) = \sigma(\boldsymbol{v}^T \boldsymbol{W}_{\cdot j}).$$

Similarly

$$P(v_i = 1|\boldsymbol{h}) = \sigma(\boldsymbol{W}_{i\cdot} \boldsymbol{h}).$$

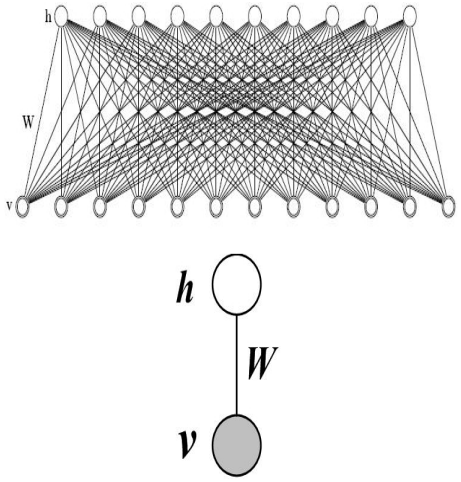

Figure 1: Restricted Boltzmann machine

The marginal distribution $P(\boldsymbol{v})$ does not factorize (if it did the model would be trivial) and, by symmetry, the same is true of the prior $P(\boldsymbol{h})$. (Thus the hidden variables are *not* independent in the prior, contrary to what experience with directed graphical models might suggest.)

3.2. Gaussian-Bernoulli restricted Boltzmann machines

The visible vectors are assumed to be real valued, the hidden vectors binary valued and the energy function is given by

$$E(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{2}(\boldsymbol{v} - \boldsymbol{b})^T(\boldsymbol{v} - \boldsymbol{b}) - \boldsymbol{c}^T \boldsymbol{h} - \boldsymbol{v}^T \boldsymbol{W} \boldsymbol{h} \quad (\boldsymbol{v} \in \mathbb{R}^I)$$

The conditional distribution of $P(\boldsymbol{v}|\boldsymbol{h})$ is Gaussian with mean $\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}$ and identity covariance matrix. As for the posterior $Q(\boldsymbol{h}|\boldsymbol{v})$, it factorizes as

$$Q(\boldsymbol{h}|\boldsymbol{v}) = \prod_j Q(h_j|\boldsymbol{v})$$

where

$$Q(h_j = 1|\boldsymbol{v}) = \sigma(c_j + \boldsymbol{v}^T \boldsymbol{W}_{\cdot j}).$$

The marginal $P(\boldsymbol{h})$, that is,

$$\int P(\boldsymbol{v}, \boldsymbol{h}) \mathrm{d}\boldsymbol{v}$$

cannot be evaluated explicitly but if we write

$$P(\boldsymbol{v}) = \sum_{\boldsymbol{h}} P(\boldsymbol{h}) P(\boldsymbol{v}|\boldsymbol{h})$$

we can interpret $P(\boldsymbol{v})$ as a Gaussian mixture with a prodigious number of components, where the mixture weights are evaluated in a complicated way and each component has the identity matrix as covariance matrix. Of course, the assumption of common identity covariance matrix is unreasonable unless the data has been appropriately preprocessed (e.g. by whitening it so that the mean of the data is zero and the global covariance matrix is the identity matrix.) Allowing for the possibility of a full covariance matrix for each mixture component $\boldsymbol{h}$ is more difficult [6].

This type of model could serve as a sort of a UBM for speaker recognition but we will not attempt to explore this possibility here. In speech recognition, it is used to binarize observation vectors (typically an acoustic observation vector consists of cepstral coefficients extracted from a block of 11 successive frames) [3, 4, 5, 6, 7, 8] and we will use it to binarize i-vectors so that binary Boltzmann machines can be used in subsequent processing.

In its simplest incarnation the idea here is that a data vector $\boldsymbol{v}$ could be mapped to the binary vector $\boldsymbol{h}$ which maximizes $Q(\boldsymbol{h}|\boldsymbol{v})$. In practice, a mean field approximation is generally used, so that a vector of Bernoulli probabilities (that is, the probabilities $Q(h_j = 1|\boldsymbol{v})$) rather than a binary vector is produced instead [2]. This representation is suitable for subsequent processing involving variational Bayes computations with binary Boltzmann machines.

3.3. Sparsely connected Boltzmann machines

We use the term *layer* to refer to a set of units none of which are connected another, so that the corresponding submatrix of the weight matrix is zero. (For example, an RBM consists of a hidden layer and a visible layer.) We say that a Boltzmann machine is *sparsely connected* (or simply sparse) if the units can be partitioned into a small number of layers. We will denote the variables corresponding to the layers by $\boldsymbol{h}^0, \ldots, \boldsymbol{h}^L$ where $\boldsymbol{h}^0$ corresponds to the visible layer.

The weight matrix looks like

$$\begin{pmatrix} \mathbf{0} & & & \\ & \mathbf{0} & & \\ & & \ddots & \\ & & & \mathbf{0} \end{pmatrix}$$

There is one $\mathbf{0}$ matrix for each layer and no restrictions on the off-diagonal blocks. In the case of a restricted Boltzmann machine, there are just 2 blocks.

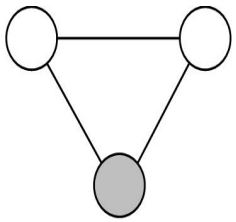

Figure 2: Sparse Boltzmann machine with one visible and two hidden layers

If $\boldsymbol{W}^{kl}$ denotes the matrix of weights on the branches joining units in layer l to those in layer k, the energy function is given by

$$E(\boldsymbol{h}) \quad = \quad -\sum_{k<l} \boldsymbol{h}^{kT} \boldsymbol{W}^{kl} \boldsymbol{h}^{l}$$

where k, l range from 0 to L.

For any pair of layers k, l, the *conditional* joint distribution $P(\boldsymbol{h}^k, \boldsymbol{h}^l | \boldsymbol{h}^{\setminus k,l})$ is a restricted Boltzmann machine (whose weight matrix is a modified version of $\boldsymbol{W}^{kl}$); the *unconditional* joint distribution is not. In fact,

$$P(\boldsymbol{h}^k, \boldsymbol{h}^l) \quad = \quad \sum_{\boldsymbol{h}^{\setminus k,l}} P(\boldsymbol{h}^k, \boldsymbol{h}^l, \boldsymbol{h}^{\setminus k,l})$$

which shows that the unconditional joint distribution can be regarded as a Boltzmann machine with hidden variables and we know that the class of Boltzmann machines with hidden variables is larger than the class of Boltzmann machines. In particular, it follows that

$$\left\langle h_i^k h_j^l \right\rangle_{\text{model}} \neq \left\langle h_i^k h_j^l \right\rangle_{\text{RBM}}$$

in general, where RBM refers to the restricted Boltzmann machine defined by the matrix $\boldsymbol{W}^{kl}$. In training a sparse BM, one might be tempted to estimate the weight matrices $\boldsymbol{W}^{kl}$ individually using RBM training but that would not be a correct procedure.

However sparse Boltzmann machines are easier to train than general Boltzmann machines because both Gibbs sampling and variational Bayes can be implemented very efficiently by cycling among layers (rather than cycling among individual units), in exactly the same way as Gibbs sampling can be carried out in RBMs by alternating between the hidden and visible layer. Note that, for each layer l, the posterior $Q(\boldsymbol{h}^l | \boldsymbol{h}^{\setminus l})$ is factorial since

$$\begin{aligned} \ln Q(\boldsymbol{h}^l | \boldsymbol{h}^{\setminus l}) \quad &= \quad \ln P(\boldsymbol{h}) - \ln P(\boldsymbol{h}^{\setminus l}) \\ &\equiv \quad \sum_{\boldsymbol{h}^{\setminus l}} E(\boldsymbol{h}^l, \boldsymbol{h}^{\setminus l}) \\ &= \quad \boldsymbol{a}^T \boldsymbol{h}^l \end{aligned}$$

where $\boldsymbol{a}$ depends on $\boldsymbol{h}^{\setminus l}$ but not on $\boldsymbol{h}^l$. Thus we can write

$$Q(\boldsymbol{h}^l | \boldsymbol{h}^{\setminus l}) \quad = \quad \prod_i Q(h_i^l | \boldsymbol{h}^{\setminus l})$$

and implement Gibbs sampling at layer l by sampling the various units independently of each other.

Likewise for variational Bayes, assuming that the variational posteriors factorizes over layers, that is

$$Q(\boldsymbol{h} | \boldsymbol{h}^0) \quad = \quad \prod_{l=1}^{L} Q(\boldsymbol{h}^l | \boldsymbol{h}^0),$$

is enough to ensure a full factorization. This is because the variational update formula for a layer l is

$$\begin{aligned} Q(\boldsymbol{h}^l | \boldsymbol{h}^0) \quad &\equiv \quad E_{\boldsymbol{h}^{\setminus l}} \left[\ln P(\boldsymbol{h}) \right] \\ &\equiv \quad E_{\boldsymbol{h}^{\setminus l}} \left[-E(\boldsymbol{h}) \right] \\ &= \quad \sum_{k<l} \boldsymbol{h}^{kT} \boldsymbol{W}^{kl} \boldsymbol{h}^{l} \\ &= \quad \boldsymbol{a}^T \boldsymbol{h}^l \end{aligned}$$

where $\boldsymbol{a}$ depends on $\boldsymbol{h}^{\setminus l}$ but not on $\boldsymbol{h}^l$. Thus we can implement variational Bayes at layer l by updating the variational posteriors for the various units independently.

4. Architectures for speaker verification

Assume that we have a representation of whole utterances (of arbitrary duration) by binary vectors (of fixed length), analagous to the i-vector representation [10]. Stated in its most general form, the speaker verification problem can be formulated as one of determining whether two collections of utterances were uttered by the same speaker or by different speakers. If the collections are denoted by E (for enrollment) and T (for test), the likelihood ratio for the verification trial is

$$\frac{P(E,T)}{P(E)P(T)} \tag{1}$$

where each term is the joint probability of a collection of utterances which are *not* independent. For example, $P(E,T)$ is the joint probability of all of the utterances in the enrollment and test sets, calculated under the hypothesis that they were uttered by a single speaker. Thus the problem confronting us is to use the sparse Boltzmann machine apparatus to construct the joint distribution on an arbitrary set of utterances uttered by a single speaker. The construction ought to be such that the joint distribution is invariant under permutations of the utterances. We will sketch several ways of doing this.

4.1. Siamese twins

Suppose that we build a sparse Boltzmann machine (of whatever topology) to represent the marginal distribution of individual utterances. We can construct a model for pairs of utterances by taking two copies of the original Boltzmann machine and gluing them together by adding branches joining units in one copy to units in the other. Requiring that the additional weight matrix be symmetric ensures that the distribution on pairs of utterances is symmetric. The construction extends straightforwardly to handling N-tuples of utterances (replicate the glue for every pair of utterances). The twin model is a sparse Boltzmann machine if the singleton model is (but it is not a deep Boltzmann machine in the sense of [2]). This model can be thought of as learning an analogue of the cosine distance metric [10].

So the numerator and the denominator in the verification likelihood ratio can be approximated by the variational free energy calculation for sparse Boltzmann machines. Note that the partition functions are not needed (although it would be desirable to have them to see how well calibrated the likelihood ratios are). In practice, verification decisions are made by comparing the likelihood ratio is with a decision threshold whose

value could be determined in theory from the parameters of a NIST-like detection cost function but which is usually determined empirically in practice. As long as this determination is done empirically, the partition functions can be absorbed into the decision threshold and their values need not be known.

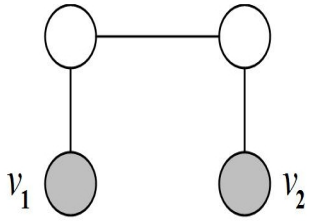

Figure 3: Example of the Siamese twin construction

4.2. Speaker factors and channel factors

An alternative approach to constructing a joint distribution on pairs of utterances by a speaker would be to take any distribution on pairs of visible vectors $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ and modify it so as to obtain a symmetric distribution. For example we could start with a RBM and tie the weight matrices as in Fig. 4. The units in the hidden layer would account for both utterances in the pair so that they would play a role analogous to speaker factors in Joint Factor Analysis or Probabilistic Linear Discriminant Analysis [9, 11]. We will refer to units like this as *tied units*. This model can obviously be extended to handle multiple utterances in a permutation-invariant way but it would not be very powerful and something analogous to channel factors would seem to be needed, as in Fig. 5.

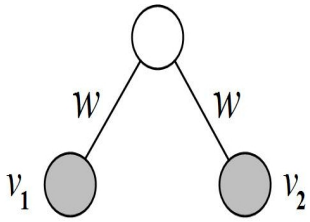

Figure 4: Tied hidden variables

The model in Fig 5 is an RBM (although it may not look like one). As such the free energies needed to evaluate the verification likelihood ratio (up to the partition functions) can be evaluated exactly. The computational burden of evaluating the free energy of a set of N utterances is proportional to N rather than to N^2 as in the case of the twin model.

4.3. Third order Boltzmann machines

Another possibility to be explored eventually would be the use of third order or gated Boltzmann machines which allow for the possibility of two sets of hidden variables $\{x_j\}$ and $\{y_k\}$ and three-way interactions involving products $v_i x_j y_k$ where the v's are the visible variables. If the x's were tied and the y's untied, the x's could play the role of speaker factors and the y's could play the role of channel factors. (The variational Bayes EM training algorithm extends to straightforwardly to third order Boltzmann machines although steps need to be taken to control the number of free parameters to be estimated [17].)

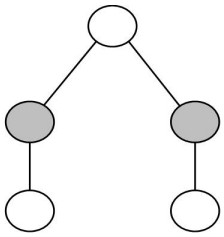

Figure 5: An RBM with tied hidden variables (on top) and untied hidden variables (on the bottom)

4.4. Preprocessing

We began this section by assuming that a representation of whole utterances by binary vectors was somehow available. We believe that finding representations of this type will be a very interesting avenue of research but for the time being we have to make to with a modified version of the i-vector representation. Using CD-1, we trained a Gaussian-Bernoulli RBM on 800 dimensional length-normalized i-vectors. Then, as explained in Section 3.2, we used this Gaussian-Bernoulli RBM to map each i-vector to an array of Bernoulli probabilities so that it can be processed with binary Boltzmann machines of the types that we have described.

5. Experimental Results

We experimented with the Siamese twin model on the female portion if the NIST 2010 extended speaker recognition evaluation data set (normal vocal effort telephone speech).[1]

5.1. Data sets

We used a standard front end consisting of 20 dimensional Gaussianized cepstral coefficients together with their first and second derivatives and trained a a 2048 component, full covariance universal background model using male and female data from the NIST 2004 and 2005 speaker recognition evaluations. We used an 800 dimensional gender-independent i-vector extractor trained using the Switchboard and Fisher corpora, and some Mixer data (namely the 2004, 2005 and 2006 speaker recognition evaluation data together with the interview development data from the 2008 evaluation). The female portion of the Switchboard and Mixer data sets (about 20,000 utterances) was used to train the Gaussian-Bernoulli RBM and Siamese twin Boltzmann machine used to make speaker verification decisions.

5.2. Implementation of the Siamese twin model

The topology that we used for the Siamese twin model is slightly more complicated than that depicted in Fig. 3 — we glued together the the visible layers as well as the hidden layers. The singleton model — that is, the model for the marginal distribution of individual utterances — is just an RBM and it is especially easy to train as variational Bayes is not needed. (Posteriors can be calculated exactly as explained in Section 3.1.) The additional parameters that need to be estimated in the twin model are the weights on the branches joining the two copies of the singleton model. For these, the full variational Bayes training algorithm is needed. We found that 25–30 training epochs

[1] We plan to include results for male speakers in the final version if the paper is accepted.

were sufficient.

Likelihood ratios for verification trials were evaluated according to (1) using variational free energies (Section 3.3) as proxies for log likelihoods in a similar spirit to [11]. (Of course, since the singleton model is a restricted Boltzmann machine, these free energies can be evaluated exactly in the case of the denominator as explained in Section 3.1.) It may seem surprising that the (intractable) partition functions are not needed here. The reason is that the likelihood ratio (1) only needs to be compared to an empirically determined decision threshold and the values of the partition functions can be absorbed into this threshold.

Error rates evaluated with the 2008 and 2010 detection cost functions for different configurations of the twin model are reported in Table 1 and the corresponding DET curves are in Fig. 6.

The i-vectors have first been projected to an LDA bases, that reduced their dimensionality from 800 to 200. We then applied within-class covariance normalization (WCCN) and finally length normalization, i.e. projection onto the unit sphere, [10].

Table 1: *Error rates on the extended NIST 2010 speaker recognition data (female telephone speech) obtained with different configurations of the twin model. BM-200-100 indicates 200 hidden units in the Gaussian-Bernoulli RBM and 100 hidden units in the singleton RBM*

	EER	2008 NDCF	2010 NDCF
cosine distance	3.45 %	0.309	0.449
BM-250-250	2.68 %	0.27	0.45
BM-250-200	2.51 %	0.27	0.41
BM-200-150	3.13 %	0.28	0.46
BM-200-200	2.98 %	0.28	0.47

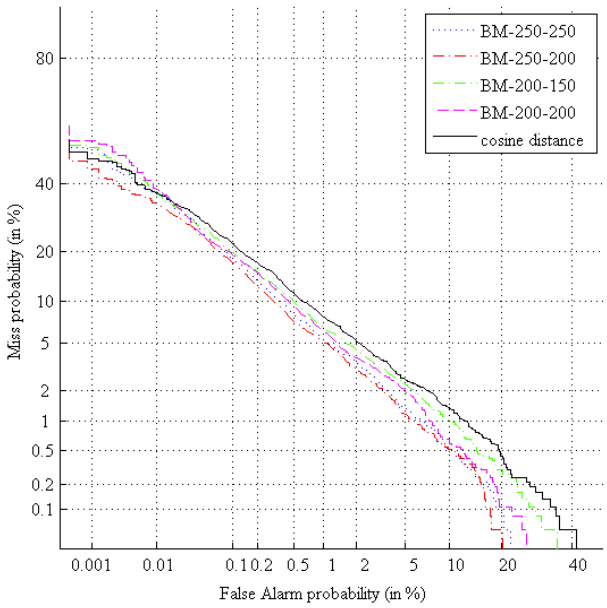

Figure 6: Det curves for various configurations of the twin model

The results are demonstrated in Table 1. Clearly, the twin model outperforms the cosine distance with relative improvement equal to 27%, indicating that the model is capable of learning a better distance, in terms of the official scoring metric.

6. Conclusion

We have conducted a preliminary investigation into the use of Boltzmann machines as generative models for collections of utterances spoken by a single speaker which has enabled us to build a rudimentary classifier for speaker verification. Clearly, a lot of work remains to be done to achieve performance comparable to other i-vector based speaker recognition systems but our long term goals are actually more ambitious.

A typical state of the art speaker recognition system uses two generative models for feature extraction (namely a UBM to extract Baum-Welch statistics and an i-vector extractor) in addition to a generative model (such as Probabilistic Linear Discriminant Analysis) to make verification decisions. All of these generative models are potentially replaceable by Boltzmann machines and it is possible to envisage a complete speaker recognition architecture built in this way.

Research in Boltzmann machines has already produced a very interesting candidate for replacing the UBM, namely the Gaussian-Bernoulli Restricted Boltzmann machine applied in the front end as in speech recognition applications [3, 4, 5, 6, 7, 8]. This appears to be capable of much more fine-grained modeling than conventional Gaussian mixture UBMs having a few thousand components and it is capable of modeling much longer intervals of speech (11 – 15 frames are typical).

Perhaps the most interesting avenue of research will be to find new ways of producing binary valued i-vector like features. At this writing, we can only speculate on what form these features might eventually assume but a plausible scenario would be to devise a binary Boltzmann machine to model whole utterances. The visible variables could be binarized acoustic observations produced by a Gaussian-Bernoulli RBM and utterances could be characterized by tied hidden variables. (Thus these variables would differ from one utterance to another but would be tied across all of the acoustic observations in a given utterance.) Tying could be enforced using a model similar to that illustrated in Fig. 5 or a third order Boltzmann machine.

Acknowledgements

Thanks to Niko Brummer and Najim Dehak for stimulating discussions and especially to Ruslan Salakhutdinov for making his Boltzmann machine code base available to us.

7. References

[1] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[2] R. Salakhutdinov and G. Hinton, "An efficient learning procedure for deep boltzmann machines," in *CSAIL Technical report, MIT-CSAIL-TR-2010-37*, Aug. 2010.

[3] A. Mohammed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Audio, Speech, and Language Processing*, 2010.

[4] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing (Special Issue on Deep Learning for Speech and Language Processing)*, Jan. 2012.

[5] A. Mohammed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, and M. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Proc. ICASSP 2011*, 2011.

[6] G. Dahl and G. Hinton, "Phone recognition with the mean-covariance restricted boltzmann machine," in *Advances in Neural Information Processing 23*, 2010.

[7] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," in *Advances in Neural Information Processing 23* .

[8] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Eurospeech 2011*, 2011.

[9] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 16, no. 5, pp. 980–988, July 2008. [Online]. Available: http://www.crim.ca/perso/patrick.kenny

[10] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[11] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Proc. Odyssey 2010: The speaker and Language Recognition Workshop*, Brno, Czech Rebublic, June 2010.

[12] S. Cumani, N. Brummer, L. Burget, and P. Laface, "Fast discriminative speaker verification in the i-vector space," in *Proceedings ICASSP*, 2011, pp. 4852–4855.

[13] L. Burget, O. Plchot, S. Cumani, O. O. Glembek, P. Matejka, and N. Brummer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *Proceedings ICASSP*, 2011, pp. 4832–4835.

[14] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer Science+Business Media, LLC, 2006.

[15] D. MacKay, *Information theory, inference and learning algorithms*. New York, NY: Cambridge University Press, 2003.

[16] Y. Bengio and O. Delalleau, "Justifying and generalizing contrastive divergence," *Neural Computation*, vol. 21, no. 6, pp. 1601–1621, 2009.

[17] R. Memisevic and G. Hinton, "Learning to represent spatial transformations with factored higher-order Boltzmann machines," *Neural Computation*, vol. 22, pp. 1473–1492.