# Conditionally Independent Component Extraction for Naive Bayes Inference

Shotaro Akaho

Electrotechnical Laboratory,
1–1–4 Umezono, Tsukuba-shi, Ibaraki 305-8568, Japan
s.akaho@aist.go.jp,
WWW home page: http://staff.aist.go.jp/s.akaho/index.html

**Abstract.** This paper extends the framework of independent component analysis (ICA) to supervised learning. The key idea is to find a conditionally independent representation of input variables for given output. The representation is useful for the naive Bayes learning which has been reported to perform as well as more sophisticated methods. The learning algorithm is derived in a similar criterion to ICA. Two dimensional entropy takes an important role, while one dimensional entropy does in ICA.

## 1 Introduction

Independent component analysis (ICA) has been successfully applied to data analysis, signal processing and modeling of brain functions, especially early vision[3]. ICA has been restricted to unsupervised learning and we extend the framework of ICA to supervised learning. As an application of this extension, we focus on naive Bayes learning in this paper. Naive Bayes learning scheme performs well especially on most classification tasks, and is often significantly more accurate than more sophisticated methods. However, for real valued prediction, the assumption of conditional independence is sensitive to the performance. This paper proposes a method to find conditionally independent components.

## 2 Naive Bayes learning

We consider the problem of predicting an output value $y$ for an input vector $\boldsymbol{x}$ with $d$ elements $x_1, \ldots, x_d$. If the probability density function $p(y \mid \boldsymbol{x})$ is known, we can choose $y$ to minimize the expected prediction error. However, $p(y \mid \boldsymbol{x})$ is not usually known and it must be estimated from a set of training samples. If the dimensionality of $\boldsymbol{x}$ is large, the learning is considered to be difficult, which is often called "curse of dimensionality". Naive Bayes learning solves this problem by assuming the conditional independence of $x_i$'s for given $y$. Under this assumption, $p(y \mid \boldsymbol{x})$ can be factorized as

$$p(y \mid \boldsymbol{x}) = \frac{p(x_1 \mid y) \cdots p(x_d \mid y)\, p(y)}{\int p(x_1 \mid y) \cdots p(x_d \mid y)\, p(y)\, dy}, \tag{1}$$

thus the estimation of $p(y \mid \boldsymbol{x})$ is reduced to the estimation of individual functions $p(x_i \mid y)$ and $p(y)$, which makes the problem much easier.

Despite its simplicity, the naive Bayes scheme performs well on most classification tasks, even if the assumption of independence is seriously violated[5]. Although this scheme has been tried to real value estimation as well, it is reported that the performance is rather sensitive to the validity of the assumption of independence than the discrete case[7, 6]. In order to reduce this sensitivity, we propose a method to find a linear transformation of $\boldsymbol{x}$ which provides a representation satisfying the independence assumption.

## 3 Conditionally independent component extraction

### 3.1 Problem

We assume that there is an unknown vector of source signals $\boldsymbol{s} = (s_1, \ldots, s_{d^*})$ whose elements are mutually independent conditionally for given $y$,

$$p(\boldsymbol{s} \mid y) = p(s_1 \mid y) \cdots p(s_{d^*} \mid y). \tag{2}$$

Observed $\boldsymbol{x}$ is considered to be a linear mixture of $\boldsymbol{s}$. For the sake of simplicity, we assume $d^* = d$. The case $d^* < d$ is dealt with as a dimension reduction problem in section 5. If $d^* = d$, the observation is generated by

$$\boldsymbol{x} = A\boldsymbol{s}, \tag{3}$$

where $A \in \mathcal{R}^{d \times d}$ is an unknown nonsingular matrix. Without knowing $\boldsymbol{s}$ and $A$, we want to recover $\boldsymbol{s}$ from $\boldsymbol{x}$ by a linear transformation

$$\boldsymbol{z} = W\boldsymbol{x}, \tag{4}$$

where $W \in \mathcal{R}^{d \times d}$ is a matrix. If $W = A^{-1}$, we obtain $\boldsymbol{z} = \boldsymbol{s}$. However, by the same discussion as ICA, $W$ can not be uniquely determined because of the ambiguity of the order and the amplitude of signals. Our goal is to find $W$ to recover $\boldsymbol{s}$ except for this ambiguity. In addition, it is necessary to assume that at most one conditional distribution $p(s_i \mid y)$ is Gaussian. Otherwise, there remains continuous freedom of rotation transformation.

### 3.2 Cost function

In order to evaluate the degree of the conditional independence, we define the cost function to be minimized by the Kullback-Leibler divergence between the left and the right hand side of (2), which is averaged over the whole $y$,

$$L(W) = \mathrm{E}_y \left[ K[p(\boldsymbol{z} \mid y) \| \prod_{i=1}^{d} p(z_i \mid y)] \right] = K[p(\boldsymbol{z}, y) \| p(y) \prod_{i=1}^{d} p(z_i \mid y)], \tag{5}$$

where $K[p\|q)] = \int p \log(p/q)$.

In terms of entropy, the cost function can be written in the form,

$$L(W) = -H(\boldsymbol{z}, y) + \sum_{i=1}^{d} H(z_i, y) + (1-d)H(y). \tag{6}$$

Since $H(\boldsymbol{z}, y) = H(\boldsymbol{x}, y) + \log|W|$, we have

$$L(W) = -\log|W| + \sum_{i=1}^{d} H(z_i, y) + \text{const.} \tag{7}$$

where $|W|$ denotes the determinant of $W$,

### 3.3 Learning algorithm

The cost function is similar to that of ICA based on the mutual information minimization[9], except the entropy term of one variable in ICA is replaced by the joint entropy of $z_i$ and $y$. We can derive the gradient descent learning algorithm for the cost function, CICA (conditionally independent component analysis) algorithm.

**CICA algorithm** (general form)

$$\Delta W \propto -\frac{\partial L(W)}{\partial W} W^T W = \left(\mathrm{I}_d - \mathrm{E}_{y,\boldsymbol{z}}[\boldsymbol{\varphi}(\boldsymbol{z}, y)\boldsymbol{z}^T]\right) W, \tag{8}$$

where $\mathrm{I}_d$ is a unit matrix of order $d$, and $\boldsymbol{\varphi}$ is a vector with elements

$$\varphi_i(\boldsymbol{z}, y) = -\frac{\partial \log p(z_i, y)}{\partial z_i} = \frac{-\partial p(z_i, y)/\partial z_i}{p(z_i, y)}, \tag{9}$$

In the above algorithm, the gradient is calculated in the sense of natural gradient which makes the algorithm simpler and faster.

There are various possibilities how the probability distribution $p(z_i, y)$ is estimated from training samples. We use kernel density estimation described in the next section both for CICA and for the naive Bayes learning. In the above algorithm, the average $\mathrm{E}_{y,\boldsymbol{z}}[u]$ can be considered in three ways: the first one is that $u$ is numerically integrated by the estimated $p(z_i, y)$, whose cost is the most expensive and might not be practical. The second one, which we use, is batch learning where $u$ is averaged over training samples. The third one is on-line learning where $\mathrm{E}_{y,\boldsymbol{z}}$ is eliminated, which still ensures the convergence in the sense of 'stochastic approximation'.

## 4 Kernel density estimation

Suppose we have $n$ training samples $(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$. Let $\hat{W}$ denote an estimation of the demixing matrix $W$, and $\boldsymbol{z}^{(j)}$ denote the transformed value of $\boldsymbol{x}^{(j)}$ by $\hat{W}$, $\boldsymbol{z}^{(j)} = \hat{W}\boldsymbol{x}^{(j)}$.

Kernel density estimation is a nonparametric method[8], which is written for the joint distribution of $z_i$ and $y$ as

$$\hat{p}(z_i, y) = \frac{1}{nh_i h_y} \sum_{j=1}^{n} K\left(\frac{z_i - z_i^{(j)}}{h_i}\right) K\left(\frac{y - y^{(j)}}{h_y}\right), \tag{10}$$

where $K$ is a kernel function, and $h_i$ and $h_y$ are parameters defining kernel widths for $z_i$ and $y$.

A typical choice for $K$ is the Gaussian kernel $K(u) = \exp(-u^2)/\sqrt{2\pi}$, and we use it in our experiments. Various methods to determine the kernel widths have been proposed. However, since performances of those methods depend on applications, we choose simple one, $h_i = \sigma_i n^{-1/6}$, $h_y = \sigma_y n^{-1/6}$, where $\sigma_i^2$ and $\sigma_y^2$ are sample variances of $z_i$ and $y$. Although this kernel widths are derived under the assumption that $z_i$ and $y$ are jointly Gaussian, it does not affect seriously even if the assumption is not correct.

In the case of Gaussian kernel, $\varphi_i$ in (9) can be written as

$$\varphi_i(\boldsymbol{z}, y) = \frac{\sum_{j=1}^{n} (z_i - z_i^{(j)}) K_i(z_i - z_i^{(j)}) K_y(y - y^{(j)})}{h_i^2 \sum_{j=1}^{n} K_i(z_i - z_i^{(j)}) K_y(y - y^{(j)})}, \tag{11}$$

where $K_i(u) = K(u/h_i)$, $K_y(u) = K(u/h_y)$.

It is necessary to estimate $p(y)$ and $p(z_i \mid y)$ for the naive Bayes, they can be estimated by $\hat{p}(y) = K_y(y - y^{(j)})/(nh_y)$ and $\hat{p}(z_i \mid y) = \hat{p}(z_i, y)/\hat{p}(y)$. Although a kernel width of order $n^{-1/5}$ achieves better performance for one dimensional density estimation, we use the same kernel width for the consistency that the marginal distribution of $\hat{p}(z_i, y)$ is equal to $\hat{p}(y)$.

## 5 Dimension reduction

In this section, we consider the case that the dimensionality of observations is larger than that of sources, i.e. $d > d^*$. In order to apply the CICA algorithm, we reduce the dimensionality of $\boldsymbol{x}$ by linear transformation, $\boldsymbol{u} = B\boldsymbol{x}$, where $B$ is a $d^* \times d$ matrix and $\boldsymbol{u}$ is a transformed value. How should we choose $B$? There are two requirements for $\boldsymbol{u}$ : one is that $\boldsymbol{u}$ is preferred to preserve as much information on $y$ as possible. The other is that applying the CICA algorithm to $\boldsymbol{u}$ can achieve the conditional independence as well as possible. In this paper, we deal with mainly the former one, since the latter may need some change to the CICA algorithm which is left as a future work.

Although it is difficult to find the map that preserves the most information on $y$ in a general setting, the canonical correlation analysis (CCA) provides the map that we want if a pair of multivariates are jointly Gaussian. Since we have only one dimensional output variable $y$, CCA gives a map onto just one dimensional space. Therefore, we use CCA which is nonlinearly extended by basis functions: let $c_1(y), \ldots, c_k(y)$ be $k(\geq d^*)$ functions of $y$ which are linearly independent. We

apply CCA between $\boldsymbol{x}$ and $\boldsymbol{c}(y) = (c_1(y), \ldots, c_k(y))^T$. We obtain $\boldsymbol{u}$ by choosing $d$ dimensional canonical subspace.

In order to design a good set of functions $\boldsymbol{c}(y)$, we have to check how transformed variable $\boldsymbol{u}$ fulfills the two requirements described above.

## 6 Simple experiments

We show a simple experiment result for artificial data to ascertain the algorithm to work. In this experiment, we does not investigate the case of dimension reduction.

Training data and test data are generated as follows: first output value $y$ is uniformly distributed from $u[-3, 3]$ and two dimensional source signals are generated by $s_1 = y + \varepsilon_1$, $s_2 = y^2 + \varepsilon_2$, where $\varepsilon_1$ and $\varepsilon_2$ are independently generated from $u[-1, 1]$. Observation signals are generated by mixing source signals, $x_1 = s_1 + 0.5s_2$ and $x_2 = 0.5s_1 + s_2$.

We applied the CICA algorithm to 20 training samples. After each step of the CICA, we evaluated the prediction error by using 100 test samples. Prediction is taken by the posterior average of $y$, $\hat{y} = \int y\hat{p}(y \mid \boldsymbol{x}; \hat{W})\, dy$, which achieves least mean square error, where $\hat{p}$ is an estimated predictive distribution by the CICA and the naive Bayes learning.

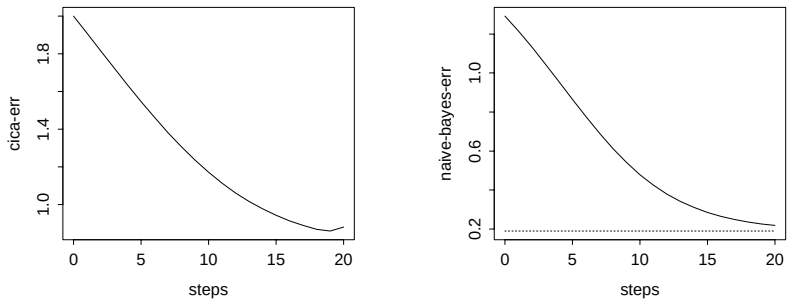


**Fig. 1.** Error curves for the CICA and the naive Bayes

Figure 1 shows the convergence of the algorithm. The error for the CICA (the left figure) is evaluated by

$$\sum_{i=1}^{d}(\sum_{j=1}^{d}\frac{|P_{ij}|}{\max_k |P_{ik}|} - 1) + \sum_{j=1}^{d}(\sum_{i=1}^{d}\frac{|P_{ij}|}{\max_k |P_{kj}|} - 1), \tag{12}$$

which represents the recovery rate of source signals, where $P = \hat{W}A$. The right figure shows the mean square error of the naive Bayes learning in each step of the CICA. The lower line is the minimum error when the source signals are known.

This experiment shows that although the naive Bayes is rather sensitive to the assumption of conditional independence, the CICA algorithm can successfully recover the conditional independence.

## 7 Concluding remarks

In this paper, we have proposed an extended framework of ICA to supervised learning scheme and applied to the naive Bayes learning. Applying the algorithm to real world data remains as a future work.

The method is useful not only for the naive Bayes learning, but also for appropriate representation of graphical models[4]. Conditionally independent representation might simplify the structure of the graphical models and the table of conditional probabilities.

The CICA is similar to the multimodal ICA (MICA) proposed by the author[1], which extends the canonical correlation analysis. The assumption of MICA is that the pairs of source signals are jointly independent which is different from the assumption in this paper. However, it is technically similar because the cost functions of both algorithms include two dimensional entropy.

ICA can be considered as a model of lower level systems in the brain such as early vision. Becker et al[2] has proposed the higher level model where the system receives signals from several kinds of lower level systems. the CICA and the MICA can be a candidate of models for such a higher level systems.

## References


1. Akaho, S., Kiuchi, Y., Umeyama, S.: MICA: Multimodal independent component analysis. In *Proc. of IJCNN* (1999) 927–932
2. Becker, S.: Mutual Information Maximization: Models of Cortical Self-Organization. *Network: Computation in Neural Systems*, **7** (1996)
3. Bell, A. J., Sejnowski, T.J.: The 'independent components' of natural scenes are edge filters. *Vision Research*, **37** (1997) 3327–3338
4. Cowell, R.G., Dawid, A.P., Lauritzen, S.L., Spiegelhalter, D.J.: *Probabilistic Networks and Expert Systems*. Springer (1999)
5. Domingos, P. and Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, **29** (1997) 103–130
6. Frank, E., Leonard, T., Holmes, G., Witten, I.H.: Naive Bayes for regression. *Machine Learning*, **41** (2000) 5–25
7. Friedman, J.: On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, **1** (1997) 55–77
8. Simonoff, J.S.: *Smoothing Methods in Statistics*. Springer-Verlag (1998)
9. Yang H., Amari, S.: Adaptive online learning algorithms for blind separation: Maximum entropy and minimum mutual information. *Neural Computation*, **9** (1997) 1457–1482