RC23314 (W0408-175) August 30, 2004
Computer Science

IBM Research Report

Kikuchi-Bayes: Factorized Models for Approximate Classification in Closed Form

Alex Jakulin*, Irina Rish, Ivan Bratko*
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

*Faculty of Computer and Information Science
University of Ljubljana
Trzaska cesta 25
SI-1001 Ljubljana, Slovenia

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Kikuchi-Bayes: Factorized Models for Approximate Classification in Closed Form

Aleks Jakulin
Faculty of Computer and Information Science
University of Ljubljana, Tržaška cesta 25
SI-1001 Ljubljana, Slovenia
jakulin@acm.org

Irina Rish
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532, USA
rish@us.ibm.com

Ivan Bratko
Faculty of Computer and Information Science
University of Ljubljana, Tržaška cesta 25
SI-1001 Ljubljana, Slovenia
ivan.bratko@fri-uni.lj.si

Abstract

We propose a simple family of classification models, based on the Kikuchi approximation to free energy. We note that the resulting product of potentials is not normalized, but for classification it is easy to perform the normalization for each instance separately. We propose a learning method based on including those initial regions that would otherwise be significantly different from those estimated directly. We observe that this algorithm outperforms other methods, such as the tree-augmented naïve Bayes, but that the inclusion of regions may increase the approximation error, even in cases when adding a region does not yield loopy dependencies.

1 Introduction

In this paper, we focus on the problem of building probabilistic classifiers from data. Since in high-dimensional domains, it is impossible to reconstruct the true global probabilistic model from a limited amount of data (and even with potentially unlimited data, the model complexity might be too high), various approximations techniques are used. A common approach is to use local models: namely, every joint model that fits the data globally should also fit the data when marginalized to any subset of attributes. We need not build an intractable globally consistent model, but instead an ensemble of submodels, each modelling a region of the attributes.

One approach to reintegrating the submodels into a single global model is to consider each submodel a constraint and seek a joint probability mass function (pmf) conforming to them [4]. For example, the maximum entropy approach is to seek the global model with the highest entropy of those consistent with the constraints, essentially assuming that no information is to be assumed by the model beyond that provided by the constraints. Approaches

such as iterative scaling are typically used in order to find a max-entropy distribution satisfying such constraints [2]. However, the disadvantage of iterative scaling is that the complete complex global model itself needs to be modelled explicitly.

The cluster variation method was originally proposed as a way of estimating the entropy of the whole complex system model, by only having information about its parts [5]. Recently, the Kikuchi approximation to free energy has been shown to be a special case of region graph approximations, and the probability distributions inferred by the algorithms of the belief propagation family are at the stationary points of the Kullback-Leibler divergence between the region-based approximation of the joint probability density function and the inferred pmf (probability mass function) []. Thereby, each submodel corresponds to a region of attributes, and can be represented with a region graph. Interestingly, for tractable region graphs, the models obtained through the chain rule, the MaxEnt approach and with the cluster variational method are identical.

In this paper, we will focus on a very simple inferential task of classification, where the query attribute is the label, and all the other attributes are the evidence. No approximate marginalization method is needed. We will not use the cluster variation method to model entropy but instead to model the global pmf directly. We will employ the interaction testing approach to determine the submodels required for good predictive performance without restricting ourselves to tractable hierarchical, graphical or decomposable probabilistic models.

There was plenty of work, but whenever they got overlapping regions, they had to merge them into a single humongous region. The novelty of our approach is that we're using Kikuchi to handle the overlap: sometimes exactly, sometimes approximately.

The second novelty is that most of the learning around NIPS is done with respect to KL-divergence, without correct regard for the risk and chance. We're using significance testing to show how it applies.

The third novelty is the focus on assuring the local fit, and rely on approximation to do its work. We do not do global model selection, because it's so arbitrary and because it is not consistent on projections.

2 Notation and Definitions

Let $\mathbf{X} = \{X_1, ..., X_n\}$ be a set of observed random variables, called *attributes*, and let $\mathbf{x} = (x_1, ..., x_n)$ be a vector of values assigned to the variables in $\mathbf{X}$. Herein, we assume discrete-valued attributes, i.e. $\mathbf{x} \in \mathcal{X} = \{\mathcal{X}_1 \times ... \times \mathcal{X}_n\}$ where each $\mathcal{X}_i$ is a set of possible values of X_i. Let C denote an unobserved random variable called the *class*, where $c \in \mathcal{C}, |\mathcal{C}| = m\}$. The set of attributes together with the class is denoted $\mathbf{Y} = \mathbf{X} \cup \{C\}$. An assignment $\mathbf{y} = (\mathbf{x}, c)$ of values to the attributes and the class is called an *instance*, or *example*. The set of all possible instances is denoted $\Omega = \mathcal{X}_1 \times ... \times \mathcal{X}_n \times \mathcal{C}$. We will use a short notation $P(\mathbf{y}) = P(\mathbf{x}, c) = P(x_1, ..., x_n, c)$ to describe the joint probability distribution $P(X_1 = x_1, ..., X_n = x_n, C = c)$. A subset of variables $R \in \mathbf{Y}$ is called a *region* (or *cluster*), and a value assignment to R is denoted $\mathbf{y}_R$.

A *classifier* is a mapping $h : \mathcal{X} \rightarrow \mathcal{C}$ that assigns a class value to any given instance. Particularly, the *Bayes classifier* $h^*(\mathbf{x}) = \arg\max_c P(c|\mathbf{x}) = \arg\max_c P(\mathbf{x}, c)$ selects the most-likely value of class given the observed attributes, and is provably optimal (i.e. has the lowest error probability, or lowest *risk*, among all classifiers). However, in practice, the true underlying distribution $P(c, \mathbf{x})$ (or, respectively, $P(c|\mathbf{x})$) is not available, and is hard to estimate from a limited set of training instances, especially in case of high-dimensional vectors of attributes.

A common approach to this problem is to assume a certain simplified class of joint probability mass functions $\hat{P}(c, \mathbf{x})$ that approximate $P(c, \mathbf{x})$. For example, one of the simplest and perhaps most popular probabilistic classifiers is the *naïve Bayes* that assumes attribute independence given the class, thus approximating $P(\mathbf{x}, c)$ by $\hat{P}(\mathbf{x}, c) = \prod_i P(x_i|c)P(c)$. Other approaches include less restrictive assumptions on the structure of $\hat{P}$, such as trees (e.g., Tree-Augmented Naïve Bayes (TAN) or, more generally, Bayesian networks [3]. We will use undirected graphical models such as *Markov networks*.

Markov networks. Given a set of random variables $\mathbf{X}$, a *Markov network* (also called *undirected graphical model* or *Markov random field* for $\mathbf{X}$), is defined as a pair (G, S) where G is an undirected graph and $S = (\Phi_1, \ldots, \Phi_m)$ is a set of (positive) functions (called *potentials*), defined over each of m cliques in G, such that the joint distribution $P(\mathbf{x})$ is factorized over the set of these cliques, i.e. $P(\mathbf{x}) = (1/Z) \prod_i \Phi(\mathbf{x}_i)$ where Z is a normalization constant. [1]

3 Main Idea

Note that typical probabilistic classifiers (e.g., Naïve Bayes, TAN, Bayesian networks) build an *explicit* probabilistic model $\hat{P}(c, \mathbf{x})$ by assuming a certain structure of the probability distribution (e.g., a tree-structure in TAN, or a particular factorization of $\hat{P}(c, \mathbf{x})$ according to the Bayesian network structure). These assumptions may introduce an unnecessary bias.

In this paper, we propose an alternative approach that models $\hat{P}(c, \mathbf{x})$ *implicitly* by using a collection of marginal distributions defined over (potentially all) subsets, or clusters, of the variables (clearly, the subset size is limited to a reasonable value to make the approach tractable). Briefly, this approach can be viewed as modelling $\hat{P}(c, \mathbf{x})$ by an undirected graph (a Markov network) defined by the selected clusters, rather than by a directed graph, or a Bayesian network. While other approaches to learning Markov networks aim at constructing a network of bounded treewidth (e.g., Chow-Liu tree learning approach and its generalization by [6]), so that probabilistic inference will be easy in such networks, we are not concerned with bounding the treewidth as we only use the resulting network for computing $\hat{P}(c|\mathbf{x})$ which turns out to be an easy inference problem. We are only concerned with bounding the clique size in the original (non-triangulated) network. Of course, since our networks are not triangulated, we are unable, in general, to provide an explicit (normalized) joint distribution function; however, we can still compute $\hat{P}(c|\mathbf{x})$ from such unnormalized distributions.

There are several advantages to our approach. First, the model construction is much easier: given a bound k on the cluster size, we only need to compute a polynomial ($O(n^k)$) number of marginal probabilities[2], while finding an optimal Bayesian network structure with a bound $k > 1$ on the number of parents is known to be NP-hard [1]. Second, this approach allows to take into account any subset of (significant) k-way interactions (or even all of them), instead of limiting ourselves to interactions consistent with a certain graph structure (e.g., a particular set of families in a Bayesian network). Finally, while generic inference in Markov networks is generally hard, and often requires approximations such as belief propagation and its generalizations [7], computing $\hat{P}(C|X_1, \ldots, X_n)$ is easy because it does not require normalization in $\hat{P}(\mathbf{x}) = (1/Z) \prod_i \phi(\mathbf{x}_i)$, as shown by the following lemma.

[1] Without loss of generality we could restrict the set of all cliques to the set of all *maximal* cliques.

[2] Actually, the number of all marginals of k variables is lower: $n!/k!(n-k)!$.

Theorem 1 *Given a set of random variables $\mathbf{Y} = \mathbf{X} \cup \{C\}$, a set $\mathcal{R} = \{R | R \subseteq \mathbf{Y}\}$ of subsets (regions) of $\mathbf{Y}$, where C belongs to at least one region, and a product $\Phi(\mathbf{y}) = \Phi(\mathbf{x}, c) = \prod_{R \subset \mathcal{R}} \Phi(\mathbf{y}_R)$ of non-negative functions (potentials) defined on these regions, let $\hat{P}(\mathbf{y}) = (1/Z)\Phi(\mathbf{y})$ be the corresponding joint probability distribution over $\mathbf{Y}$, where Z is a normalization constant. Then:*

1. *Computing $\hat{P}(c|\mathbf{x})$ does not require normalization, i.e. $\hat{P}(c|\mathbf{x}) = \Phi(\mathbf{y}) / \sum_c \Phi(\mathbf{y})$;*
2. *Bayesian classifier can be computed using only a product of potentials that contain C, i.e. $h^*(\mathbf{x}, c) = \arg\max_c \prod_{\{R \subset \mathcal{R} | C \in R\}} \Phi(\mathbf{y}_R)$.*

Proof. The first claim follows from $\hat{P}(c|\mathbf{x}) = \hat{P}(\mathbf{x}, c)/\hat{P}(\mathbf{x}) = (1/Z)\Phi(\mathbf{x}, c) / \sum_c (1/Z)\Phi(\mathbf{x}, c)$, since by definition $\Phi(\mathbf{y}) = \Phi(\mathbf{x}, c)$. The second claim is easily obtained from the definition of Bayesian classifier, $h^*(\mathbf{x}) = \arg\max_c \hat{P}(c|\mathbf{x})$, and the following observation:

$$\hat{P}(c|\mathbf{x}) = \frac{\Phi(\mathbf{y})}{\sum_c \Phi(\mathbf{y})} = \frac{\prod_{\{Q \subset \mathcal{R} | C \notin Q\}} \phi(\mathbf{y}_Q)}{\sum_c \Phi(\mathbf{y})} \prod_{\{R \subset \mathcal{R} | C \in R\}} \Phi(\mathbf{y}_R), \tag{1}$$

where $\frac{\prod_{\{Q \subset \mathcal{R} | C \notin Q\}} \Phi(\mathbf{y}_Q)}{\sum_c \Phi(\mathbf{y})}$ is independent of C. ■

Given a set of data, the question is how to select a factorized approximation of $P(\mathbf{y})$, i.e. how to select a set of regions and potentials over these regions. Our approach to region selection is inspired by the *cluster-variation method* (CVM) [7], also known as *Kikuchi approximation* of free energy [5]. An overview of our learning approach is given below:

Kikuchi-Bayes algorithm:

1. Given $\mathbf{Y} = \mathbf{X} \cup \{C\}$, and a bound k on region size, select an initial set of regions $\mathcal{M} = \{M | M \subseteq \mathbf{Y}\}$ using significance test described in Section 5 and estimate marginal pmfs $P_M = P(\mathbf{y}_M)$.
2. Given $\mathcal{M}$ and $\{P_M\}$, compute an extended set of regions $\mathcal{R} \supseteq \mathcal{M}$ using the *cluster-variation method* (see Section 4) and a set of marginals $P(\mathbf{y}_R)$. Approximate $P(\mathbf{y})$ by (unnormalized) product as $\Phi(\mathbf{y}) = \prod_{R \subset \mathcal{R}} P(\mathbf{y}_R)^{c_R}$ where c_R is a *counting number* for region R (see next section).
3. Classify: $c^*(\mathbf{x}) = \arg\max_c P(c|\mathbf{x}) = \arg\max_c \frac{\Phi(\mathbf{y})}{\sum_c \Phi(\mathbf{y})}$.

In the following two section we elaborate on the first two steps of the algorithm (we start with the approximation step given a set of initial regions, and then describe the initial region selection).

4 Kikuchi Approximation to Probability Distributions

Let us consider a problem of approximating a joint $P(\mathbf{Y})$ by the product of marginals over subsets of $n+1$ random variables $\mathbf{Y} = \{X_1, X_2, \ldots, X_n, C\}$.
Task: Given a set of l initial subsets of $\mathbf{Y}$ (regions) $\mathcal{M} = \{M_1, M_2, \ldots, M_l\}$, and a joint pmf for each region, $P_M = P(\mathbf{y}_M) = P(y_{M,1}, y_{M,2}, \ldots, y_{M,k})$, find an (unnormalized) approximation $\Phi_{\mathcal{M}}(\mathbf{y})$ of the intractable $P(\mathbf{y})$ using a set of $\{P(\mathbf{y}_M) | M \in \mathcal{M}\}$.
Approach: our approach to region selection is inspired by the *cluster-variation method*

(CVM) [7], also known as Kikuchi approximation of free energy [5] [3]. We apply cluster variation method [7] on the set of initial regions $\mathcal{M}$ to obtain a proper *region graph* $\{\langle R, c_R\rangle\}$ where $\mathcal{R} = \{R\}$ is a new set of regions that includes the initial set of regions, their intersections, intersections of intersections, and so on. For each region R, there is a corresponding *counting number* $\{c_R\}$, that account for region overlaps (to avoid double-counting) when using the *region-based approximation* of the *free energy* which is defined as $F_{\mathcal{R}} = U_{\mathcal{R}} - H_{\mathcal{R}}$, where $U_{\mathcal{R}}$ and $H_{\mathcal{R}}$ are the region-based approximations of the average energy and the entropy, respectively, and are given by: $U_{\mathcal{R}} = \sum_{R\in\mathcal{R}} c_R U_r(b_R)$, and $H_{\mathcal{R}} = \sum_{R\in\mathcal{R}} c_R H_R(b_R)$, where b_R is some marginal probability distribution over R, $U_R(b_R) = \sum_{\mathbf{y}_R} b_R(\mathbf{y}_R) E_R(\mathbf{y}_R)$ is the average energy, and $H_R(b_R) = \sum_{\mathbf{y}_R} b_R(\mathbf{y}_R) \ln b_R(\mathbf{y}_R)$ is the entropy of a region, respectively [7]. Region-based approximation using CVM is considered a good approximation to the (intractable) true free energy, because it accounts for the overlaps between the regions. When true $P_R = P(\mathbf{y}_R)$ are used instead of b_R (as in our case where they are obtained from 'true' empirical distribution), the region-based average energy is exact (although region-based entropy is still an approximation).

Example: Consider $\mathbf{Y} = \{A, B, C\}$ and a region graph (a set of regions with counting numbers) $\{\langle 1, \{A, B\}\rangle, \langle 1, \{B, C\}\rangle, \langle 1, \{A, C\}\rangle, \langle -1, \{A\}\rangle, \langle -1, \{B\}\rangle, \langle -1, \{C\}\rangle\}$. The region-based approximate entropy is given by $\hat{H}_{\mathcal{R}} = H(A, B) + H(B, C) + H(A, C) - H(A) - H(B) - H(C)$, and the (unnormalized) approximation to $P(A, B, C)$ is given by $\Phi(A, B, C) = P(A, B)P(B, C)P(A, C)/(P(A)P(B)P(C))$. Note that $\hat{H}_{\mathcal{R}}(\mathbf{Y}) = -\sum_{a,b,c} P(a, b, c) \ln \Phi(a, b, c)$. In general, it is easy to show that:

Theorem 2 *The region-based approximate entropy can be expressed as* $\hat{H}_{\mathcal{R}} = -\sum_{\mathbf{y}} P(\mathbf{y}) \ln \Phi(\mathbf{y})$ *where* $P(\mathbf{y})$ *is the true joint pmf over* $\mathbf{Y}$ *and* $\Phi(\mathbf{y}) = \prod_{R\in\mathcal{R}} P(\mathbf{y}_R)^{c_R}$.

Proof. $\hat{H}_{\mathcal{R}} = -\sum_{R\in\mathcal{R}} c_R \sum_{\mathbf{y}_R} P(\mathbf{y}_R) \ln P(\mathbf{y}_R) = -\sum_{R\in\mathcal{R}} c_R \sum_{\mathbf{y}} P(\mathbf{y}) \ln P(\mathbf{y}_R) =$
$= -\sum_{\mathbf{y}} P(\mathbf{y}) \ln \prod_{R\in\mathcal{R}} P(\mathbf{y}_R)^{c_R} = -\sum_{\mathbf{y}} P(\mathbf{y}) \ln \Phi(\mathbf{y})$. ■

This motivates us to use cluster-variation method for constructing regions $\mathcal{R}$ from an initial set $\mathcal{M}$, and to approximate $P(\mathbf{y})$ by using a set of $P(\mathbf{y}_R)$ as follows [4]:

$$\Phi_{\mathcal{M}}(\mathbf{y}) = \prod_{R\in\mathcal{R}} P(\mathbf{y}_R)^{c_R}. \tag{2}$$

Note that the approximation in the equation 2 has a nice property: if the region graph has no cycles, $P(\mathbf{y}) = \Phi_{\mathcal{M}}(\mathbf{y})$, i.e. the approximation becomes exact [7]; of course, this is not true in general, when there are cycles in the region graph – in this case normalization constraint may not hold ($\sum_{\mathbf{y}} \Phi(\mathbf{y}) \neq 1$).

5 Selection the Initial Set of Regions

Task: Given a set of N i.i.d. instances $\mathbf{Y}^N = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, where $\mathbf{y}_i \in \Omega = \mathcal{X} \times \mathcal{C}$, determine the set of initial regions $\mathcal{M} = \{M_1, M_2, \ldots, M_l\}$ so that the $\Phi(\mathbf{y})$ obtained using estimates $P_{M_i} = P(\mathbf{y}_{M_i})$ and a region graph constructed using $\mathcal{M}$ will be a 'good' approximation of $P(\mathbf{y})$ (in a sense explained below).

[3] Note that we only use the cluster selection part of the method, and do not perform an iterative belief propagation over the selected set of regions, since classification is an easy inference problem of computing $P(c|\mathbf{x})$ given that all variables $\mathbf{X}$ are observed.

[4] Note that $P(\mathbf{y}_R)$ is computed by marginalization of some $P(\mathbf{y}_M)$ where $R \subseteq M$. Since each P_M is a marginal of some $P(\mathbf{y})$, we get unique $P(\mathbf{y}_R)$ when using different $M \in \mathcal{M}, R \subseteq M$.

Assume that we have a candidate approximation $\Phi_{\mathcal{M}}$ based on a set of initial regions $\mathcal{M}$. It is an intractable problem trying to compare $\hat{P}_{\mathcal{M}}(\mathbf{y}) = (1/Z)\Phi_{\mathcal{M}}$ and $P(\mathbf{y})$ directly if $\mathbf{y}$ is of high dimensionality. It is possible, though, to compare their marginals $\hat{P}(\mathbf{y}_R)$ and $P(\mathbf{y}_R)$ within a tractable region R. We will denote by $\mathbf{x}_R$ the projection of $R \subseteq \mathbf{Y} = \mathbf{X} \cup \{C\}$ on variables in $\mathbf{X}$. If the two are significantly different, region R should be included among the initial regions, for which $P(\mathbf{y}_R)$ is estimated directly from the data and not approximated. We question the quality of the predicted class distribution $D(P(c|\mathbf{x}_R)||\hat{P}(c|\mathbf{x}_R))$.

Significance: Since KL-divergence between these two marginals is exceedingly rarely zero, it would seem that P and $\hat{P}$ are always different. We need a different criterion: when are P and $\hat{P}$ *significantly* different. $P(\mathbf{y}_R)$ is estimated from the instance vector $\mathbf{Y}^N$ (which we denote by $P(\mathbf{y}_R|\mathbf{Y}^N)$). Now assume that $\mathbf{Y}^N$ itself is a sample from $P(\mathbf{y}_R)$. The idea of nonparametric bootstrap is random sampling with replacement from $\mathbf{Y}^N$ generates resamples that are equally likely and of the same size as $\mathbf{Y}^N$. Given a particular resample $\tilde{\mathbf{Y}}^N$, we can compute the *self-loss* $D(P(\mathbf{y}_R|\tilde{\mathbf{Y}}^N)||P(\mathbf{y}_R|\mathbf{Y}^N))$. This way, even the correct model will often end up with a particular loss on a finite sample. In fact, we can speak of a probability distribution of self-loss: $\Pr\{D(P(\mathbf{y}_R|\tilde{\mathbf{Y}}^N)||P(\mathbf{y}_R|\mathbf{Y}^N)) \leq d\}$.

The key idea of goodness-of-fit testing, as originated by K. Pearson, is that the difference between P and $\hat{P}$ is *significant* if it is unlikely that the self-loss of P would be as large as the observed approximation loss $D(P||\hat{P})$. Using the distribution of self-loss as a reference, we can estimate the p-value γ of the difference between P and $\hat{P}$:

$$\gamma = \Pr\{D(P(\mathbf{y}_R|\tilde{\mathbf{Y}}^N)||P(\mathbf{y}_R|\mathbf{Y}^N)) \geq D(P(\mathbf{y}_R|\mathbf{Y}^N)||\hat{P}(\mathbf{y}_R|\mathbf{Y}^N))\}$$

When we are working with a potential Φ rather than with the approximate pmf $\hat{P}(\mathbf{y}) = (1/Z)\Phi(\mathbf{y})$, the potential Φ needs to be normalized if the divergence is to be meaningful. In case the normalization of $\Phi(c|\mathbf{x}_R)$ is needed, it can always be performed locally without having to seek $\hat{P}$. We denote this way of computing conditional KL-divergence with $\hat{D}$:

$$\hat{D}(P(c|\mathbf{x}_R)||\Phi(c|\mathbf{x}_R)) \triangleq \sum_{\mathbf{x}_R}\sum_{c} P(\mathbf{x}_R, c)\log\frac{P(\mathbf{x}_R, c)\sum_{c'}\Phi(\mathbf{x}_R, c')}{P(\mathbf{x}_R)\Phi(\mathbf{x}_R, c)}.$$

Building Models: We start with a single initial region which only includes the label attribute $M = \{C\}$, $\mathcal{M}_1 = \{M\}$. In the first stage, we verify on all clusters of size 2 that include the label that the approximation based on this initial region alone is not significantly worse than the true model. Therefore, for each attribute X, we compare the self-loss of $P(c|x)$ with the loss of $\hat{P}_{\mathcal{M}_1}(c|x) = P(C)$. If the loss of $\hat{P}$ is significant, the region $\{X, C\}$ is included in the set of candidates. However, we do not make use of these candidates until the end of the stage, as this would imply the undesirable relevance of the order of testing individual attributes. In the second phase, we add the set of candidates into the set of initial regions, obtaining $\mathcal{M}_2$, possibly removing the regions that are mere subsets of the new candidates. For all pairs of attributes X_i, X_j, we examine the goodness of fit of $\hat{P}_{\mathcal{M}_2}(c|x_i, x_j)$ to $P(c|x_i, x_j)$.

In general, at level k, for all k-tuples of attributes $\mathbf{X}_k \in \mathcal{P}(\mathbf{X}), |\mathbf{X}_k| = k$, the significance of including the region $R = \{C\} \cup \mathbf{X}_k$ is validated by comparing the the true $P(c|\mathbf{x}_k)$ with the approximation based on the region graph constructed from the set of initial regions from the set of at levels lower than k that intersect with R: $\{\mathcal{S} \cap \mathcal{R};\ \mathcal{S} \in \mathcal{M}, \mathcal{S} \cap \mathcal{R} \neq \emptyset\}$. All those regions R where the approximation is significantly different are added to the set of initial regions. We refer to this algorithm as Kikuchi-Bayes level k.

	Classification error						Average negative log-likelihood					
	NB	K1	TN	TK	K2	win	NB	K1	TN	TK'	K2	win
NB		0.3	-0.6	-0.4	-0.2	**29.5**		0.4	0.1	0.1	0.5	**9.1**
K1	-0.3		-1.0	-0.7	-0.6	**47.7**	-0.4		-0.3	-0.3	0.1	**29.5**
TN	0.6	1.0		0.2	0.4	**20.5**	-0.1	0.3		0.1	0.4	**27.3**
TK	0.4	0.7	-0.2		0.2	**15.9**	-0.1	0.3	-0.1		0.3	**13.6**
K2	0.2	0.6	-0.4	-0.2		**29.5**	-0.5	-0.1	-0.4	-0.3		**34.1**
lose ↑	**22.7**	**20.5**	**25.0**	**20.5**	**31.8**		**18.2**	**15.9**	**34.1**	**11.4**	**22.7**	

Table 1: The percentage of the domains in which a particular algorithm achieved the best or the worst result according to a particular criterion appears in bold. The expected difference in rank between each pair of methods appears within each matrix. For example, the expected gain in rank of K1 versus NB measured by classification accuracy is 0.3.

6 Experiments

We have compared the following algorithms on 44 classification domains, most were from the UCIKDD repository: a) Kikuchi-Bayes level 1 with the p-value cutoff γ at 0.1 (K1) can be seen as an approach to feature selection for the naïve Bayesian classifier. b) Kikuchi-Bayes level 2 with $\gamma = 0.1$ (K2). c) Kikuchi-Bayes level 2 with $\gamma = 0.1$ which does not include a candidate region if it would cause a cycle (TK). d) Kikuchi-Bayes level 2 with $\gamma = 1.0$, which includes all possible clusters of two attributes and the label (K2'). e) The naïve Bayesian classifier (NB), which corresponds to K1 with $\gamma = 1.0$ and consequently no cluster selection. f)The tree-augmented naïve Bayesian classifier [3] (TN).

All continuous attributes were discretized beforehand with the Fayyad-Irani discretization procedure. All the missing attribute values were handled as special values. The Laplacean prior was employed for estimating the probability density function from the data within each region. The generalization error of the algorithms was tested using 5-fold cross-validation replicated twice. To evaluate the calibration of probabilistic predictions for each test instance $\langle \mathbf{x}, c \rangle$, the negative log-likelihood is computed as $\log \sum_{c'} \Phi(\mathbf{x}, c') - \log \Phi(\mathbf{x}, c)$, and is averaged over all the instances. The average ranks of the methods across the data sets are:

	Classification error						Average negative log-likelihood					
	NB	K1	TN	TK	K2	K2'	NB	K1	TN	TK	K2	K2'
rank	2.93	2.65	3.52	3.30	3.15	5.45	3.38	2.98	3.19	3.13	2.85	5.48

The rank of a method depends strongly on the criterion used for comparing the classifiers. Overall, Kikuchi-Bayes level 1 has best performance in classification, and Kikuchi-Bayes level 2 wins in probability estimation. There are some distinct differences: the naïve Bayes excels in classification even if it is a rather bad probability estimator. On the other hand, Kikuchi-Bayes level 2 achieves good probability estimation performance, but this does not always reflect in superior classification performance. The heuristic that we used for determining which regions to include in the model was the Kullback-Leibler divergence, not the classification error. It is therefore unsurprising that the improvement was greater with respect to the log-likelihood criterion. The morale is that one should employ that heuristic that maximizes the preferred criterion of final evaluation. For example, if the final evaluation of the methods employs classification accuracy and cross-validation, then the learning should not employ the KL-divergence with the bootstrap, but cross-validation with some measure of class ordering.

The most striking aspect of the results is the frequently inferior performance of the full unpruned Kikuchi-Bayes level 2 model without cluster selection. However, based on the detailed results excluding it in Table 1, we cannot claim the inherent superiority of any other method. It is therefore sensible to examine the reasons for success or failure on a few domains where the differences between methods are most accentuated. We will disregard

the generalization error and focus on the bias, the approximation error assessed on the training set itself, illustrating that the addition of clusters may deteriorate the performance.

Although the full Kikuchi-Bayes level 2 algorithm generally yielded dismal performance, it excelled on domains like 'tic-tac-toe', which resembles the task of graphical pattern recognition, and on the synthetic benchmark 'monk-2'. We plotted the performance of K1, K2 and de-cycled K2 (TK) at differing levels of the significance testing parameter γ. Figure 1 illustrates 'tic-tac-toe', where K2' wins due to usefulness of almost each pair of attributes: the restrictions of K1 and TK prevent the utilization of these pairs.

In spite of significant clusters, there is deterioration to addition of clusters in 'voting'. After the initial gains (based on two cliques of 3 and one clique of 4 attributes), additional clusters result in larger cliques and a large approximation errors, in spite of significant clusters. Unfortunately, significance only indicates the reliability of the probability estimate, not the reduction in estimation error, and the local learning neglects the additional approximation error that arises from cycles.

The tree-based model is better than K2 at any setting of γ, but adding edges into the tree does not improve the performance monotonically. It is incorrect to view γ as a domain-dependent tuning parameter: γ does affect how many clusters will get included, but it has meaning of its own that should remain unrelated to the issue of approximation error. Modelling approximation error in Kikuchi-Bayes remains an open issue.

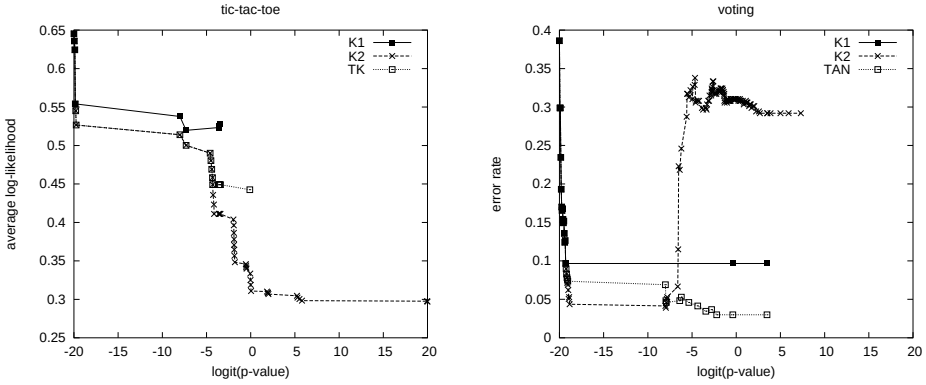

Figure 1: We graphically illustrate the dependence of classification performance on the type of the model and the significance testing threshold γ. The horizontal scale indicates the logit-transformed value of the threshold $\log(\gamma/(1-\gamma))$ used as a parameter for Kikuchi-Bayes learning.

References

[1] D.M. Chickering, D. Geiger, and D. Heckerman. Learning Bayesian networks: Search methods and experimental results. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, page 112, 1995.

[2] J. N. Darroch and D. Ratcliff. Generalised iterative scaling and maximum likelihood. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

[3] N. Friedman, D. Geiger, and Goldszmidt M. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[4] C. T. Ireland and S. Kullback. Contingency tables with given marginals. *Biometrika*, 55(1):179–188, 1968.

[5] R. Kikuchi. A theory of cooperative phenomena. *Physical Review*, 81(6):988–1003, 1951.

[6] N. Srebro. Maximum likelihood bounded Tree-Width markov networks. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 504–511, 2001.

[7] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR2004-040, MERL, 2004.

A Appendix (may be ignored at the discretion of the program committee)

$\mathcal{R}_0 \leftarrow \emptyset$ {The set of initial regions, without redundancies.}
for all $\mathcal{S} \in \mathcal{M}$ **do** {for each initial region}
  if $\forall \mathcal{S}' \in \mathcal{R}_0 : \mathcal{S} \not\subseteq \mathcal{S}'$ **then**
    $\mathcal{R}_0 \leftarrow \mathcal{R}_0 \cup \{\mathcal{S}\}$ {$\mathcal{S}$ is not redundant.}
  end if
end for
$\mathcal{R} \leftarrow \{\langle \mathcal{S}, 1\rangle;\ \mathcal{S} \notin \mathcal{R}_0\}$ {The output region graph with counting numbers.}
$k \leftarrow 1$
while $|\mathcal{R}_{k-1}| > 1$ **do** {there are feasible subsets}
  $\mathcal{R}_k \leftarrow \emptyset$
  for all $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{R}_{k-1}$, $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$, $\mathcal{S}_1 \cap \mathcal{S}_2 \ni \mathcal{R}_k$ **do** {all overlapping pairs of regions}
    $c \leftarrow 1$ {the counting number}
    for all $\langle \mathcal{S}', c' \rangle \in \mathcal{R}$, $\mathcal{S}_1 \cap \mathcal{S}_2 \subseteq \mathcal{S}'$ **do**
      $c \leftarrow c - c'$ {consider the counting numbers of all submodels containing the intersection}
    end for
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{\langle \mathcal{S}_1 \cap \mathcal{S}_2, c\rangle\}$
    $\mathcal{R}_k \leftarrow \mathcal{R}_k \cup \{\mathcal{S}_1 \cap \mathcal{S}_2\}$
  end for
end while

Algorithm 1: Cluster variation method for constructing the set of submodels using the initial set of regions $\mathcal{M} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\}$.

	Classification error						Average negative log-likelihood					
domain	NB	K1	TN	TK	K2	K2'	NB	K1	TN	TK	K2	K2'
anneal	18.4	**17.8**	47.8	21.5	60.4	100.0	0.64	**0.60**	1.49	0.61	2.67	11.51
audiology	89.2	**55.3**	89.4	88.5	76.8	100.0	5.35	**2.20**	5.03	5.18	3.47	11.51
australian	14.5	**14.5**	14.9	14.9	15.2	44.7	0.45	0.45	0.39	**0.36**	0.38	1.32
balance-scale	**8.8**	**8.8**	17.8	17.0	**8.8**	32.4	**0.52**	**0.52**	0.59	0.58	**0.52**	0.70
breast-LJ	28.0	**27.8**	28.8	28.3	29.0	43.7	0.62	0.61	**0.57**	0.59	0.61	0.86
breast-wisc	**2.5**	**2.5**	4.8	3.1	48.8	96.3	0.20	0.20	0.14	**0.11**	3.07	9.08
bupa	35.5	39.4	33.2	33.3	34.9	**32.2**	0.63	0.67	**0.61**	0.61	0.62	0.61
car	14.0	14.1	14.8	14.4	**6.0**	94.6	0.33	0.33	0.50	0.49	**0.26**	5.99
cmc	47.8	47.8	46.4	46.9	**45.7**	58.7	1.00	1.00	0.95	0.95	**0.95**	1.38
crx	13.7	**13.5**	14.4	14.0	14.9	39.1	0.45	0.44	0.40	**0.37**	0.41	1.35
ecoli	44.2	**35.7**	64.0	61.3	**35.7**	99.7	1.24	**1.03**	2.06	1.65	**1.03**	10.00
german	25.7	25.6	27.7	27.8	**24.9**	40.9	**0.53**	0.53	0.59	0.58	0.55	1.18
glass	**28.7**	30.8	31.1	31.1	31.1	99.5	**0.83**	0.84	0.92	0.92	0.84	10.31
hayes-roth	17.8	**15.6**	32.8	34.4	**15.6**	41.9	0.56	**0.55**	0.71	0.72	**0.55**	0.96
heart	**44.4**	44.5	47.3	47.4	44.9	84.1	1.22	1.20	**1.19**	1.31	1.23	5.83
hepatitis	**14.8**	15.5	18.4	18.4	16.5	84.8	0.61	0.48	**0.46**	0.70	0.53	6.53
horse-colic	65.0	**52.6**	96.1	91.7	74.0	99.2	3.68	**2.51**	8.17	7.31	3.79	11.40
ionosphere	9.7	9.7	10.0	**9.3**	68.7	77.1	0.77	0.77	**0.55**	0.58	6.73	10.13
iris	5.7	5.7	**5.0**	5.3	5.7	99.3	0.16	**0.16**	0.22	0.17	**0.16**	4.01
krkp	12.2	12.0	7.9	**6.2**	6.9	21.3	0.29	0.29	0.19	**0.16**	0.22	0.93
lenses	**12.5**	31.2	29.2	45.8	25.0	95.8	0.62	0.63	0.91	0.96	**0.60**	1.65
lung-cancer	46.9	**43.8**	53.1	57.8	64.1	92.2	2.23	**1.27**	2.54	3.35	5.19	10.36
lymphography	42.9	**35.5**	94.9	67.6	43.2	100.0	1.79	**1.03**	3.99	2.70	1.51	11.51
monk1	25.4	25.4	**0.0**	**0.0**	**0.0**	16.7	0.50	0.49	0.15	0.12	**0.10**	0.36
monk2	37.9	34.3	34.9	33.3	30.8	**26.5**	0.65	0.65	0.62	0.60	0.55	**0.52**
monk3	3.6	3.6	1.7	1.4	**1.1**	4.4	0.20	0.20	0.11	0.10	**0.09**	0.16
mushroom	4.7	4.7	**0.0**	0.0	3.9	4.3	0.14	0.14	**0.00**	0.00	0.39	0.43
o-ring-erosion	**13.0**	17.4	15.2	**13.0**	15.2	15.2	0.59	0.60	0.51	**0.50**	0.59	0.53
pima	21.9	21.9	**21.9**	22.9	22.7	25.9	0.49	0.49	**0.46**	0.48	0.48	0.52
post-operative	36.4	**28.4**	39.8	39.8	30.1	47.7	0.72	**0.61**	0.78	0.75	0.69	1.02
primary-tumor	84.7	**72.1**	88.5	88.3	72.9	100.0	3.85	**2.64**	4.49	4.26	2.67	11.38
promoters	**9.9**	**9.9**	21.2	23.6	30.7	66.0	0.28	**0.21**	0.78	0.85	1.39	7.20
segment	**8.9**	**8.9**	84.2	65.6	100.0	100.0	**0.45**	0.45	4.09	3.23	11.50	11.51
shuttle-control	6.7	6.7	**2.6**	3.0	3.6	43.1	0.17	0.19	0.27	0.18	**0.14**	0.58
soybean-large	**9.8**	**9.8**	51.1	29.3	**9.8**	96.3	0.73	0.73	1.75	1.21	**0.73**	11.51
soybean-small	**0.0**	**0.0**	1.1	3.2	**0.0**	100.0	0.00	0.00	0.02	0.09	**0.00**	11.51
tic-tac-toe	30.1	29.1	22.0	26.4	25.6	**20.6**	0.55	0.55	0.49	0.53	0.50	**0.41**
titanic	22.1	22.1	**21.1**	**21.1**	21.2	21.2	0.52	0.52	**0.48**	**0.48**	0.48	0.48
vehicle	39.1	39.1	32.0	**31.0**	82.0	83.3	1.80	1.80	**0.71**	0.92	8.94	9.06
voting	10.0	9.9	**5.5**	6.4	79.1	84.6	0.59	0.59	**0.19**	0.21	7.77	9.38
wdbc	4.2	4.2	5.8	**3.2**	91.9	94.2	0.22	0.22	**0.16**	0.17	10.00	10.74
wine	**1.1**	**1.1**	19.9	3.7	**1.1**	100.0	0.03	0.03	0.58	0.10	**0.03**	11.51
yeast-class	**0.3**	**0.3**	2.7	0.5	**0.3**	66.9	0.01	0.01	0.14	0.03	**0.00**	11.51
zoo	8.4	**5.4**	12.4	12.9	5.9	100.0	0.16	**0.14**	0.36	0.30	0.14	11.51
avg rank	2.93	2.65	3.52	3.30	3.15	5.45	3.38	2.98	3.19	3.13	2.85	5.48

Table 2: NB - naïve bayes, K1 - kikuchi level 1, TN - tree-augmented naïve bayes, TK - kikuchi level 2 without loops, K2 - kikuchi level 2, K2' - kikuchi level 2 without significance testing. Outrageous error rates (100%) indicate numerical instabilities caused by repeated multiplication of near-zero probabilities (making a probability distribution impossible to normalize), which were handled as misclassifications.

$k \leftarrow 1$ {Size of the candidate regions}
$\mathcal{M} \leftarrow \{C\}$ {Set of initial regions}
while $k \leq K$ **do**
  $\mathcal{M}' \leftarrow \emptyset$ {Candidate regions of cardinality k}
  for all $\mathbf{X}_k \in \mathcal{P}(\mathbf{X}), |\mathbf{X}_k| = k$ **do** {for each candidate}
    $\mathcal{F} \leftarrow \{C\} \cup \mathbf{X}_k$ {The consistency focus always includes the label.}
    $\mathcal{R}' \leftarrow CVA(\{\mathcal{S} \cap \mathcal{F};\ \mathcal{S} \in \mathcal{M}, \mathcal{S} \cap \mathcal{F} \neq \emptyset\})$ {The local region graph within the focus.}
    $\Phi'(C, \mathbf{X}_k) = \prod_{\langle \mathbf{X}_R, c_R \rangle \in \mathcal{R}'} P(\mathbf{X}_R)^{c_R}$
    if $\Pr\{D(P'(C|\mathbf{X}_k)||P(C|\mathbf{X}_k)) > \hat{D}(P(C|\mathbf{X}_k)||\Phi'(C|\mathbf{X}_k))\} < \gamma$ **then**
      $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{\mathcal{F}\}$ {A significant improvement, include the candidate region.}
    end if
  end for
  $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{M}'$
  $k \leftarrow k + 1$
end while
$\mathcal{R} \leftarrow CVA(\mathcal{M})$

Algorithm 2: General framework of the Kikuchi-Bayes level K algorithm for learning a K-consistent initial region structure from the data for predicting the label C, given a set of attributes $\mathbf{X}$ and their potential set $\mathcal{P}(\mathbf{X})$. P' is a model estimated on a bootstrap resamples of the original data (explained in Sect. 5), and the p-value γ was used as the significance threshold.