ON MIXING MODELS

Tjalling Tjalkens

Eindhoven University of Technology, Dept EE,
P.O.Box 513, 5600 MB Eindhoven, THE NETHERLANDS, t.j.tjalkens@tue.nl

1. INTRODUCTION

The Context-Tree-Weighting algorithm is an example of an computationally efficient way to compute a Bayesian mixture of context-tree models. In this talk I will present several classes of models for which efficient mixing procedures exist. After briefly touching the binary CTW method I will discuss an efficient way to determine the MAP context tree model given an observed data sequence. Then I discuss the way to extend this method to non-binary alphabets. The decision tree problem requires a different model class and I will briefly review a possible method to find these trees using mixtures. Finally I will discuss a problem of Bayesian classification using partially dependent features. This problem cannot be formulated efficiently in a context tree setting and I will discuss another efficient mixture approach.

2. TREE MODELS

A Context-Tree model, which is essentially identical to a FSMX source introduced by Rissanen in [1], defines conditional symbol probabilities

$$P(x_1|x_{-k}^0), \tag{1}$$

where the conditioning strings form a proper and complete set of variable length strings $\mathcal{S}$. This defines sequence probabilities, given a sufficiently long past, as

$$P(x^T|x_{1-k_1}^0, \mathcal{S}) = \prod_{i=1}^{T} P(x_i|x_{i-k_i}^{i-1}). \tag{2}$$

Here k_i is the variable length of the current conditioning string in $\mathcal{S}$.

In the case where the x_i's are binary symbols the Context-Tree Weighting method [2] computes a Bayesian mixture probability of $P(x^T|x_*^0)$ with a special prior distribution over al models $\mathcal{S}$,

$$P(x^T|x_*^0) = \sum_{\mathcal{S}\in\mathcal{S}_D} P(\mathcal{S})P(x^T|x_{1-k_1}^0, \mathcal{S}). \tag{3}$$

The basic expression for this mixture computation is

$$P_w^s(x^T) = \frac{P_e(x^T) + P_w^{0s}(x^T)P_w^{1s}(x^T)}{2}. \tag{4}$$

Here s is a node in the context-tree and $P_e(x^T)$ is the Beta-mixture probability as presented in [3].

3. LARGER ALPHABET SIZES

Often, when compressing textual data and other large alphabet data, not all possible symbols will occur in the data. Let $\mathcal{A}$ be the large alphabet and let $\mathcal{A}[x^T]$ be the set of letters occurring in x^T, so $\mathcal{A}[x^T] \subset \mathcal{A}$. In [4] the performance of probability assignments of the form

$$P(x^T) = \sum_{j=|\mathcal{A}[x^T]|}^{|\mathcal{A}|} w_j \binom{|\mathcal{A}| - |\mathcal{A}[x^T]|}{j - |\mathcal{A}[x^T]|} P_e(x^T) \tag{5}$$

was studied for several choices of w_j and shown to achieve the Krichevski-Trofimov [3] lower bound.

Another approach was taken in [5] where a binary decomposition of the alphabet $\mathcal{A}$ was used together with an adapted binary Beta-mixture probability assignment. This method appears to work better than the subset approach of [4] even though the asymptotic performance is worse.

A computationally and storage efficient method to compute the Bayesian tree model mixture was presented in [6].

4. MAP MODEL SELECTION

Nohre [7] and Volf [8] independently presented a context maximizing method to determine the maximum aposteriori probability tree model from a large set of possible tree models, using the same model prior $P(\mathcal{S})$ as in the CTW mixture approach of Formula 3. The basic expression for the maximizing approach is

$$P_m^s(x^T) = \frac{1}{2}\max\{P_e(x^T), P_m^{0s}(x^T)P_m^{1s}(x^T)\}. \tag{6}$$

In [9] a computationally and storage efficient method was presented to compute the maximizing probability and to determine the MAP tree model.

5. OTHER MODEL CLASSES

First of all it is essential to realize that the conditioning symbols in Formula 1 can be replaced by arbitrary variables, not necssarily previous symbols. In [10] more general tree models were considered. Especially the Class III model class of that paper becomes quite important, e.g. for decision trees. In this tree class the branches are allowed to select arbitrary variables from the conditioning variables (also called features). So, in the original tree model class, called Class-IV in [10], a model with three features, say f_1, f_2, and f_3, could for example be defined by the probabilities

$$P(x|f_1=0,f_2=0), P(x|f_1=0,f_2=1),$$
$$P(x|f_1=1,f_2=0,f_3=0), P(x|f_1=1,f_2=0,f_3=1),$$
$$P(x|f_1=1,f_2=1,f_3=0), P(x|f_1=1,f_2=1,f_3=1).$$

In the case that the following holds,

$$P(x|f_1=1,f_2=0,f_3=0)=P(x|f_1=1,f_2=1,f_3=0)$$
$$P(x|f_1=1,f_2=0,f_3=1)=P(x|f_1=1,f_2=1,f_3=1)$$

Class III allows a simpler description, namely

$$P(x|f_1=0,f_2=0),\quad P(x|f_1=0,f_2=1),$$
$$P(x|f_1=1,f_3=0),\quad P(x|f_1=1,f_3=1).$$

Finally I want to mention a non-tree models that can be used in Bayesian classification problems. Consider an object $\mathcal{O}$ that belongs to a class c from a finite set of possible classes and that is described by a feature vector f^k of length k. We consider the conditional feature vector probability $P(f^k|c)$. The "naive Bayes" filter assumes conditionally independent features so

$$P(f^k|c)=\prod_{i=1}^{k}P(f_i|c). \tag{7}$$

In [11] an extention of this filter is studied. There it is assumed that the feature vector can be partitioned into a number of mutually dependent features while the partition elements are mutually independent. e.g. let f^7 be partitioned into $\{f_1^3, f_4, f_5^6, f_7\}$. This defines the probaility

$$P(f^k|c)=\ P(f_1^3|c)P(f_4|c)P(f_5^6|c)P(f_7|c). \tag{8}$$

The class is the set of all models derived from all possible partitions. [11] discusses computationally efficient ways to calculate a Bayesian mixture over all possible models for several class definitions.

6. CONCLUSION

Bayesian mixtures are a powerfull approach to optimal compression, classification, model selection, and so on. Their disadvantage often is the computational complexity of determining the mixture. The various context-tree mixture algorithms and the feature vector mixture method are examples where an independant local behaviour of the model class allows a more efficient computation by a recursive partitioning of the computations.

7. REFERENCES

[1] J. Rissanen, "Complexity of strings in the class of Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 526–532, July 1986.

[2] F.M.J. Willems, Yu.M. Shtarkov, and Tj.J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 653–664, May 1995.

[3] R.E. Krichevsky and V.K. Trofimov, "The performance of universal encoding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 199–207, Mar. 1981.

[4] Yu.M. Shtarkov, Tj.J. Tjalkens, and F.M.J. Willems, "Multi-alphabet universal coding of memoryless sources," *Probl. of Inform. Transm.*, vol. 31, no. 2, pp. 114–127, April-June 1995.

[5] Tj.J. Tjalkens, F.M.J. Willems, and Yu.M. Shtarkov, "Multi-alphabet universal coding using a binary decomposition context tree weighting algorithm," in *Proc. 15th Symp. on Inform. Theory in the Benelux*, Louvain-la-Neuve, Belgium, May 1994, pp. 259–265.

[6] F.M.J. Willems and Tj.J. Tjalkens, "Complexity reduction of the context-tree weighting algorithm: a study for KPN Research," EIDMA Report RS.97.01, Eindhoven University of Technology, 1997.

[7] R. Nohre, *Some topics in descriptive complexity*, Ph.D. thesis, Linkoping University, Sweden, 1994.

[8] P. Volf and F.M.J. Willems, "Context maximizing: Finding MDL decision trees," in *Proc. 15th Symp. on Inform. Theory in the Benelux*, Louvain-la-Neuve, Belgium, May 1994, pp. 192–200.

[9] F.M.J. Willems, Tj.J. Tjalkens, and T. Ignatenko, "Context-tree weighting and maximizing: Processing betas," in *Proc. 1th ITA*, San Diego, California, U.S.A., Feb. 2006.

[10] F.M.J. Willems, Yu.M. Shtarkov, and Tj.J. Tjalkens, "Context weighting for general finite-context sources," *IEEE Trans. Inform. Theory*, vol. IT-42, no. 5, pp. 1514–1520, Sep. 1996.

[11] Tj.J. Tjalkens, "Four model classes for efficient Bayesian selection," in *Proc. 29th Symp. on Inform. Theory in the Benelux*, Leuven, Belgium, May 2008, pp. 121–128.