A GENERALIZATION OF THE NOISY-OR MODEL

Jiří Vomlel

In this paper, we generalize the noisy-or model. The generalization is two-fold. First, we allow parents to be multivalued ordinal variables. Second, parents can have both positive and negative influences on their common child. Our generalization has several advantages: it requires only one parameter per parent, the child variable is still binary, and each inhibition probability depends on the value of the corresponding parent. We show that our generalized noisy-or model belongs to the family of Generalized Linear Models, namely, it is a subfamily of Generalized Linear Models with the logarithmic link function. We suggest a method for learning the models and report results of experiments on the Reuters text classification data. For most classes the generalized noisy-or models achieved better performance than the standard noisy-or. Finally, we describe how the suggested generalization can be extended to multivalued child variables so that the number of parameters of the model does not increase.

Keywords: Bayesian Networks, Noisy-or Model, Classification, Generalized Linear Models

Classification: 68T37, 68T30

1. INTRODUCTION

Conditional probability tables (CPTs) that are the basic building blocks of Bayesian networks [11, 8] have, generally, an exponential size with respect to the number of parent variables of the CPT. This has two unpleasant consequences. First, during the elicitation of model parameters one needs to estimate an exponential number of parameters. Second, in case of a high number of parent variables the exact probabilistic inference may become intractable.

On the other hand, real implementations of Bayesian networks (see e.g. [10]) often have a simple local structure of the CPTs. The noisy-or model [11] is a popular model for describing relations between variables in one CPT of a Bayesian network. Noisy-or is member of a family of models called models of independence of causal influence [6] or canonical models [4]. The advantage of these models is that the number of parameters required for their specification is linear with respect to the number of parent variables in CPTs and that they allow applications of efficient inference methods, see for example [5, 13]. In [18], Zagorecki and Druzdzel show that practical models, for which the authors have not used noisy-or (or noisy-max) models, even models learned from data, have many CPTs that can be approximated by a noisy-or (noisy-max) model. Also, the results presented in [17] suggest that many CPTs from real applications can be parametrized by a low number of parameters.
In this paper, we propose a generalization of the noisy-or model to multivalued parent variables. Our proposal differs from the noisy-max model [7] since we keep the child variable binary no matter what the number of states of the parent variables are. Also, we have only one parameter for each parent no matter what the number of states of the parent variables. Our generalization also differs from the generalization of the noisy-or model proposed by Srinivas [14] since in his model the inhibition probabilities cannot depend on the state of the parent variables if the state differs from the state of the child. We consider this to be a quite restrictive requirement for some applications.

We will show that our proposal is closely connected to Generalized Linear Models [9] with the logarithmic link function. We discuss methods one can use to learn parameters of the generalized noisy-or model from data. In the second part of the paper, we present results of numerical experiments on the well-known Reuters text classification data. We use this dataset to compare the performance of suggested generalizations of multivalued and binary noisy-or models. We have made the source code and datasets used in experiments freely available on the Web. In the final part of the paper we describe how the suggested generalization can be further extended to multivalued child variables and perform learning experiments of the suggested model from artificially generated data.

2. MULTIVALUED NOISY-OR

In this section, we propose a generalization of noisy-or for multivalued parent variables. Let $Y$ be a binary variable taking states $y \in \{0, 1\}$ and $X_i, i = 1, \ldots, n$ be multivalued discrete variables taking states $x_i \in \{0, 1, \ldots, m_i\}$, $m_i \in \mathbb{N}^+$. The local structure of both the standard (see, e.g., [4]) and the multivalued generalization of the noisy-or can be made explicit, as we show in Figure 1.

![Fig. 1. Noisy-or model with the explicit deterministic (OR) part.](image)

The CPT $P(Y|X_1, \ldots, X_n)$ is defined using CPTs $P(X'_i|X_i)$ as

\[ P(X'_i = 0|X_i = x_i) = (p_i)^{x_i} \]  

(1)  

\[ P(X'_i = 1|X_i = x_i) = 1 - (p_i)^{x_i} \]  

(2)
A generalization of the noisy-or model

where (for $i = 1, \ldots, n$) $p_i \in [0, 1]$ is the parameter that defines the probability that the positive value $x_i$ of variable $X_i$ is inhibited. In the formula, we use parenthesis to emphasize that $x_i$ is an exponent, not an upper index of $p_i$. The CPT $P(Y|X'_1, \ldots, X'_n)$ is deterministic and represents the logical OR function.

Remark. Note that the higher the value $x_i$ of $X_i$ the lower the probability of $X'_i = 0$, which is a desirable property in many applications.

The conditional probability table $P(Y|X_1, \ldots, X_n)$ is then defined as

$$P(Y = 0|X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(X'_i = 0|X_i = x_i)$$

$$= \prod_{i=1}^{n} (p_i)^{x_i}$$

(3)

$$P(Y = 1|X_1 = x_1, \ldots, X_n = x_n) = 1 - \prod_{i=1}^{n} (p_i)^{x_i}.$$  (4)

Remark. Note that if $m_i = 1$, i.e. the values $x_i$ of $X_i$ are either 0 or 1, then we get the standard noisy-or model [11, 4].

In Figure 2 dependence of the inhibitory probability $P(X' = 0|X = x)$ on the value $x$ of a variable $X$ is depicted for different values of the parameter $p$.

It is important to note that, contrary to the definition of causal noisy-max [4, Section 4.1.6], we have only one parameter $p_i$ for each parent $X_i$ of $Y$ no matter what is the number of states of $X_i$. This implies that our model is more restricted. But, on the other hand, the suggested simple parametrization guarantees ordinality, which is in many application a desirable property (as it is also discussed in [4]). Also, since domain experts elicit or learning algorithms estimate fewer parameters the estimates might be more reliable.

Often, in practical application of noisy-or models, we lift the requirement that if all parent variables are in state 0 then the probability of $Y = 0$ must be one. One can achieve this by the inclusion of an auxiliary parent variable $X_0$ that is assumed to be always in state 1. This auxiliary variable is called a leaky cause [4] and its inhibition probability $p_0 = p_L < 1$ is called the leaky probability. This allows the probability

$$P(Y = 0|X_1 = 0, \ldots, X_n = 0) = p_L < 1.$$ 

In this way we can model unobserved or unknown causes of $Y = 1$. When we include the leaky cause in the generalized noisy-or models the formula (3) modifies to

$$P(Y = 0|X_1 = x_1, \ldots, X_n = x_n) = p_L \prod_{i=1}^{n} (p_i)^{x_i}.$$  (5)

We conclude this section by a note on possible interpretations of multivalued noisy-or. The expression at the left hand side of formula (3) corresponds also to a model where each multivalued variable $X_i$ is replaced by $x_i$ copies of a corresponding binary variable. It means that the probability of the inhibition $p_i$ of multivalued variable $X_i$ corresponds
Fig. 2. The dependence of $P(X'_i = 0|X_i = x_i)$ on parameter $p_i = p$ and the variable value $x_i = x$.

to probability of the inhibition of binary variables repeated in model as many times as is the value of $x_i$ – with the same parameter value $p_i$. This seems appropriate, for example, for the classification of text documents discussed in Section 6. The multivalued noisy-or model corresponds to treating each word in a classified text as a separate feature (repeated as many times as it is present in the text) with a natural requirement that equivalent words must have the same inhibition probability.

Inspired by a comment of a reviewer we suggest another possible interpretation of multivalued noisy-or. For $i = 1, \ldots, n$ we can define a new parameter $q_i = p^{m_i}$ and replace variable $X_i$ by $X'_i$ so that it takes states $x'_i = \frac{x_i}{m_i}$. The value of $q_i$ is the inhibition probability of a standard noisy-or but with variables $X'_i$ taking fractional values $\frac{x_i}{m_i}$. These values might be interpreted as degrees of the presence of $X'_i$. The formula (3) modifies to:

$$P(Y = 0|X'_1 = x'_1, \ldots, X'_n = x'_n) = \prod_{i=1}^{n} (q_i)^{x'_i},$$

where $x'_i \in \{0, \frac{1}{m_i}, \ldots, \frac{m_i-1}{m_i}, 1\}$.

3. RELATION TO GENERALIZED LINEAR MODELS

In this section, we will describe the relation of the generalized noisy-or model and Generalized Linear Models [9] with the logarithmic link function (GLM-log).
Assume $p_i > 0, i = 0, 1, \ldots n$. By taking the logarithm of both sides of equation (5) we get
\[
\log P(Y = 0|X_1 = x_1, \ldots, X_n = x_n) = \log p_L + \sum_{i=1}^{n} x_i \log p_i.
\]
Define new parameters $\beta_i = \log p_i, i = 1, \ldots, n$ and $\beta_0 = p_L$. Note that since $0 < p_i \leq 1, i = 0, 1, \ldots, n$ it holds for $i = 0, 1, \ldots, n$ that
\[
-\infty < \beta_i \leq 0. \tag{6}
\]
Then we can write
\[
\log P(Y = 0|X_1 = x_1, \ldots, X_n = x_n) = \beta_0 + \sum_{i=1}^{n} x_i \beta_i = x^T \beta, \tag{7}
\]
where $x$ denotes vector $(1, x_1, \ldots, x_n)$ and $\beta$ denotes vector $(\beta_0, \beta_1, \ldots, \beta_n)$. The above formula is the formula of GLM-log of the binary variable $1 - Y$. To see this note that since the expected value $E(1 - Y|x_1, \ldots, x_n) = P(Y = 0|X_1 = x_1, \ldots, X_n = x_n)$ it holds that
\[
\log E((1 - Y)|x_1, \ldots, x_n) = x^T \beta. \tag{8}
\]
However, in (6) we require non-positive values of $\beta$. There is no such constraint in GLM-log. In this sense, the generalized noisy-or models are a subfamily of GLM-log.

4. PARENTS WITH POSITIVE AND NEGATIVE INFLUENCE

Inspired by the relation to Generalized Linear Models with the logarithmic link function we can go one step further with the generalization of noisy-or models and use a standard GLM learning method to learn values of $\beta_i, i = 0, 1, \ldots, n$. Generally, some values may be learned to be positive, which contradicts the requirement of generalized noisy-or models, see (6). However, we can give the positive values of $\beta$ a quite natural interpretation in generalized noisy-or models – they can mean that the higher values of corresponding $X_i$ imply higher inhibitory probability.

We will further generalize noisy-or models and allow parents $X_i, i = 1, \ldots, n$ of $Y$ to possibly have also a negative influence on probability of value $Y = 1$ with increasing values of $x_i$. In the generalized noisy-or model we can treat the parents with positive values of $\beta_i$ by relabeling their values $x_i \in \{0, 1, \ldots, m_i\}$ to $(m_i - x_i) \in \{m_i, \ldots, 1, 0\}$. In this way the generalized noisy-or is now capable to treat not only positive influences (presence of $X_i$ increases probability of $Y = 1$) but also negative influences (presence of $X_i$ decreases probability of $Y = 1$).

Remark. One may get a false impression that by doing this we get family of models that are equivalent to the family of GLM-log models. To see that this is, indeed, not the case, observe that GLM-log with $\beta_0 > 0$ cannot be represent in the suggested generalization of noisy-or models.

\footnote{Exactly speaking, the generalized noisy-or models with strictly positive parameters $0 < p_i \leq 1, i = 0, 1, \ldots, n$.}
5. LEARNING PARAMETERS OF THE GENERALIZED NOISY-OR

We will distinguish two basic versions of generalized noisy-or models:

(b) a generalized noisy-or with binary parent variables, we will refer to it as \textit{generalized binary noisy-or},

(m) a generalized noisy-or with multivalued parent variables, we will refer to it as \textit{generalized multivalued noisy-or}.

For both versions the learning algorithms will be the same, they will differ only in data used for learning. In the version where all variables are binary the data will be transformed so that all values are either 0 or 1. In this paper we define the transformed value to be 0 if and only if the original value is 0, otherwise it is 1.

Using the close relation to GLM-log we can apply the standard maximum likelihood estimation methods for GLMs. A method typically used to learn GLMs is the Iteratively Reweighted Least Squares (IRLS) method \cite{IRLS}. We use this method to get (unconstrained) values of $\beta$. The learned model is a GLM-log model but, as noted in the closing remark in Section 4, it need not be a generalized noisy-or model.

In the GLM-log model each parent has either positive (+) or a negative influence (−) on $Y = 1$ (we exclude parents with no influence). Thus, we can get two versions of the generalized noisy-or models:

(+) include only parents $X_i, i = 1, \ldots, n$ of $Y$ that have positive influence on probability of $Y = 1$ in GLM-log (i.e., negative $\beta_i$),

(±) include all parents from $X_i, i = 1, \ldots, n$ of $Y$. Those parents that have negative influence on probability of $Y = 1$ in GLM-log (i.e., positive $\beta_i$) have their states renumbered reversely (as discussed in Section 4) and then included in the generalized noisy-or models with a positive influence.

For both versions we use learning methods with the constraint $\beta \leq 0$. In this way we guarantee that for all possible values of $x$ in formula (7) we get probability values (i.e. values from $[0, 1]$) In the experiments reported in Section 6 we used a quasi-Newton method with box constraints \cite{box_constraints} implemented in R \cite{R}. When searching for the maximum of the conditional log-likelihood we used the formula (15) for the gradient derived in Appendix A. To avoid infinite numbers we added a small positive value ($10^{-10}$) to the numerator and the denominator in formula (15). The algorithm was started from ten different initial values of $\beta$ generated randomly from interval $[-1, +1]$.

To summarize, as a combination of the options discussed above we get six different models:

- GLM-log applied to binarized data (b-GLM-log),
- GLM-log applied to multivalued data (m-GLM-log),

\footnote{Our motivation is the classification of text documents. In the binary version we just consider whether a word is present or not in a document – we use threshold 0.5. Generally, any threshold from the interval $\langle 0, m \rangle$ can be applied.}
• Generalized binary noisy-or with parents of positive influence\(^3\) (noisy-or+),

• Generalized binary noisy-or with parents of positive and also negative influence (noisy-or±),

• Generalized multivalued noisy-or with parents of positive influence (m-noisy-or+),

• Generalized multivalued noisy-or with parents of positive and also negative influence (m-noisy-or±).

6. EXPERIMENTS

In this section, we will describe experiments we performed with the well known Reuters-21578 collection (Distribution 1.0) of text documents. The text documents from this dataset appeared on the Reuters newswire in 1987. Personnel from Reuters Ltd. and Carnegie Group, Inc. classified the documents manually to several classes according to their topic. In the test, we used the split of documents to training and testing sets according to Apté et al. \(^1\). We performed experiments with preprocessed data for eight largest classes\(^4\). To reduce the feature space we kept in data only relevant features. Namely, for each class we excluded from data all features with their correlation to the class less than 0.3. To allow interested readers to replicate our experiments easily we have made our R code and the datasets used in experiments available on the Web\(^5\).

In the experiments we compare all versions of the generalized noisy-or classifiers and the generalized multivalued noisy-or classifier with the two versions of the GLM-log and logistic classifier, as they are all defined in Section 5.

We decided to include into the models all features that were not rejected by GLM-log and logistic classifiers, respectively, as irrelevant at the significance level 0.1. We performed the experiments also with the significance level increased to 0.3. In this way, we increased the number of features, but since for most classes there was no significant increase in the accuracy we prefer simpler models. However, it may be topic for a future research to apply exhaustive feature selection methods that would find optimal models for the families of our interest.

We present the accuracy, sensitivity, specificity, and number of selected features of individual classifiers in Table 1 for binarized data and in Table 2 for multivalued data.

We summarize the results of experiments in Tables 3 and 4. We report the accuracy using the percentage scale, it is the relative proportion of correctly classified documents either as belonging to the given class or not. We print the best achieved accuracy across both versions of data bold and framed.

From Tables 3 and 4 we can see that multivalued noisy-or is more often better than binary noisy-or. There is almost no difference between the performance of noisy-or+ and noisy-or±. In case of binary data noisy-or± is better than noisy-or+ for one class only, in case of multivalued data m-noisy-or± both classifiers perform equally well. However,

---

\(^3\)This is the standard noisy-or. However, to stress the relation to other noisy-or models discussed in this paper, we will use the abbreviation noisy-or+.


\(^5\)The code and data are available at [http://www.utia.cas.cz/vomlel/generalized-noisy-or/](http://www.utia.cas.cz/vomlel/generalized-noisy-or/).
### Tab. 1. Results on binarized data

<table>
<thead>
<tr>
<th>Feature</th>
<th>b-GLM-log</th>
<th></th>
<th></th>
<th>noisy-or+</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>acc.</td>
<td>sens.</td>
<td>spec.</td>
<td>feat.</td>
<td>acc.</td>
<td>sens.</td>
</tr>
<tr>
<td>earn</td>
<td>91.09</td>
<td>97.97</td>
<td>84.36</td>
<td>10</td>
<td>92.46</td>
<td>96.49</td>
</tr>
<tr>
<td>acq</td>
<td>90.00</td>
<td>82.76</td>
<td>93.37</td>
<td>9</td>
<td>89.49</td>
<td>83.76</td>
</tr>
<tr>
<td>crude</td>
<td>97.58</td>
<td>61.98</td>
<td>99.66</td>
<td>4</td>
<td>97.58</td>
<td>61.98</td>
</tr>
<tr>
<td>money-fx</td>
<td>97.03</td>
<td>60.92</td>
<td>98.53</td>
<td>7</td>
<td>96.98</td>
<td>59.77</td>
</tr>
<tr>
<td>interest</td>
<td>96.80</td>
<td>22.22</td>
<td>99.67</td>
<td>3</td>
<td>96.67</td>
<td>17.28</td>
</tr>
<tr>
<td>trade</td>
<td>98.58</td>
<td>73.33</td>
<td>99.48</td>
<td>8</td>
<td>98.40</td>
<td>65.33</td>
</tr>
<tr>
<td>ship</td>
<td>98.77</td>
<td>50.00</td>
<td>99.58</td>
<td>3</td>
<td>98.77</td>
<td>50.00</td>
</tr>
<tr>
<td>grain</td>
<td>99.91</td>
<td>90.00</td>
<td>99.95</td>
<td>1</td>
<td>99.91</td>
<td>90.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature</th>
<th>noisy-or±</th>
<th></th>
<th></th>
<th>b-logistic</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>acc.</td>
<td>sens.</td>
<td>spec.</td>
<td>feat.</td>
<td>acc.</td>
<td>sens.</td>
</tr>
<tr>
<td>earn</td>
<td>92.46</td>
<td>96.49</td>
<td>88.52</td>
<td>10</td>
<td>94.47</td>
<td>96.49</td>
</tr>
<tr>
<td>acq</td>
<td>90.04</td>
<td>83.62</td>
<td>93.03</td>
<td>9</td>
<td>91.64</td>
<td>81.61</td>
</tr>
<tr>
<td>crude</td>
<td>97.58</td>
<td>61.98</td>
<td>99.66</td>
<td>4</td>
<td>98.58</td>
<td>80.99</td>
</tr>
<tr>
<td>money-fx</td>
<td>96.98</td>
<td>59.77</td>
<td>98.53</td>
<td>7</td>
<td>97.08</td>
<td>45.98</td>
</tr>
<tr>
<td>interest</td>
<td>96.67</td>
<td>17.28</td>
<td>99.72</td>
<td>3</td>
<td>96.80</td>
<td>22.22</td>
</tr>
<tr>
<td>trade</td>
<td>98.40</td>
<td>65.33</td>
<td>99.57</td>
<td>8</td>
<td>98.86</td>
<td>85.33</td>
</tr>
<tr>
<td>ship</td>
<td>98.77</td>
<td>50.00</td>
<td>99.58</td>
<td>3</td>
<td>99.09</td>
<td>50.00</td>
</tr>
<tr>
<td>grain</td>
<td>99.91</td>
<td>90.00</td>
<td>99.95</td>
<td>1</td>
<td>99.77</td>
<td>50.00</td>
</tr>
</tbody>
</table>

### Tab. 2. Results on multivalued data

<table>
<thead>
<tr>
<th>Feature</th>
<th>m-GLM-log</th>
<th></th>
<th></th>
<th>m-noisy-or+</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>acc.</td>
<td>sens.</td>
<td>spec.</td>
<td>feat.</td>
<td>acc.</td>
<td>sens.</td>
</tr>
<tr>
<td>earn</td>
<td>93.60</td>
<td>95.48</td>
<td>91.77</td>
<td>8</td>
<td>93.60</td>
<td>95.48</td>
</tr>
<tr>
<td>acq</td>
<td>85.53</td>
<td>67.67</td>
<td>96.78</td>
<td>6</td>
<td>85.20</td>
<td>59.20</td>
</tr>
<tr>
<td>crude</td>
<td>98.13</td>
<td>74.38</td>
<td>99.52</td>
<td>4</td>
<td>98.54</td>
<td>82.64</td>
</tr>
<tr>
<td>money-fx</td>
<td>96.98</td>
<td>44.83</td>
<td>99.14</td>
<td>5</td>
<td>97.03</td>
<td>47.13</td>
</tr>
<tr>
<td>interest</td>
<td>97.62</td>
<td>49.38</td>
<td>99.48</td>
<td>2</td>
<td>97.85</td>
<td>49.38</td>
</tr>
<tr>
<td>trade</td>
<td>98.36</td>
<td>76.00</td>
<td>99.15</td>
<td>5</td>
<td>98.36</td>
<td>72.00</td>
</tr>
<tr>
<td>grain</td>
<td>99.91</td>
<td>90.00</td>
<td>99.95</td>
<td>1</td>
<td>99.91</td>
<td>90.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature</th>
<th>m-noisy-or±</th>
<th></th>
<th></th>
<th>m-logistic</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>acc.</td>
<td>sens.</td>
<td>spec.</td>
<td>feat.</td>
<td>acc.</td>
<td>sens.</td>
</tr>
<tr>
<td>earn</td>
<td>93.60</td>
<td>95.48</td>
<td>91.77</td>
<td>8</td>
<td>94.34</td>
<td>94.92</td>
</tr>
<tr>
<td>acq</td>
<td>85.20</td>
<td>59.20</td>
<td>97.32</td>
<td>6</td>
<td>89.90</td>
<td>74.57</td>
</tr>
<tr>
<td>crude</td>
<td>98.54</td>
<td>82.64</td>
<td>99.47</td>
<td>4</td>
<td>97.99</td>
<td>71.07</td>
</tr>
<tr>
<td>money-fx</td>
<td>97.03</td>
<td>47.13</td>
<td>99.10</td>
<td>5</td>
<td>96.67</td>
<td>33.33</td>
</tr>
<tr>
<td>interest</td>
<td>97.85</td>
<td>49.38</td>
<td>99.72</td>
<td>2</td>
<td>97.35</td>
<td>35.80</td>
</tr>
<tr>
<td>trade</td>
<td>98.36</td>
<td>72.00</td>
<td>99.29</td>
<td>5</td>
<td>98.81</td>
<td>80.00</td>
</tr>
<tr>
<td>ship</td>
<td>99.13</td>
<td>66.67</td>
<td>99.67</td>
<td>4</td>
<td>99.22</td>
<td>63.89</td>
</tr>
<tr>
<td>grain</td>
<td>99.91</td>
<td>90.00</td>
<td>99.95</td>
<td>1</td>
<td>99.86</td>
<td>70.00</td>
</tr>
</tbody>
</table>
Tab. 3. Comparisons of the accuracy of the GLM-log and generalized noisy-or classifiers on binarized data.

<table>
<thead>
<tr>
<th>nr. doc</th>
<th>b-GLM-log</th>
<th>noisy-or+</th>
<th>noisy-or±</th>
<th>b-logistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn</td>
<td>1083</td>
<td>91.09</td>
<td>92.46</td>
<td>92.46</td>
</tr>
<tr>
<td>acq</td>
<td>696</td>
<td>90.00</td>
<td>89.49</td>
<td>90.04</td>
</tr>
<tr>
<td>crude</td>
<td>121</td>
<td>97.58</td>
<td>97.58</td>
<td>97.58</td>
</tr>
<tr>
<td>money-fx</td>
<td>87</td>
<td>97.03</td>
<td>96.98</td>
<td>96.98</td>
</tr>
<tr>
<td>interest</td>
<td>81</td>
<td>96.80</td>
<td>96.67</td>
<td>96.67</td>
</tr>
<tr>
<td>trade</td>
<td>75</td>
<td>98.58</td>
<td>98.40</td>
<td>98.40</td>
</tr>
<tr>
<td>ship</td>
<td>36</td>
<td>98.77</td>
<td>98.77</td>
<td>98.77</td>
</tr>
<tr>
<td>grain</td>
<td>10</td>
<td><strong>99.91</strong></td>
<td><strong>99.91</strong></td>
<td><strong>99.91</strong></td>
</tr>
</tbody>
</table>

Tab. 4. Comparisons of the accuracy of the GLM-log and the generalized noisy-or classifiers on multivalued data.

<table>
<thead>
<tr>
<th>nr. doc</th>
<th>m-GLM-log</th>
<th>m-noisy-or+</th>
<th>m-noisy-or±</th>
<th>m-logistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn</td>
<td>1083</td>
<td>93.60</td>
<td>93.60</td>
<td>93.60</td>
</tr>
<tr>
<td>acq</td>
<td>696</td>
<td>87.53</td>
<td>85.20</td>
<td>85.20</td>
</tr>
<tr>
<td>crude</td>
<td>121</td>
<td>98.13</td>
<td>98.54</td>
<td>98.54</td>
</tr>
<tr>
<td>money-fx</td>
<td>87</td>
<td>96.98</td>
<td>97.03</td>
<td>97.03</td>
</tr>
<tr>
<td>interest</td>
<td>81</td>
<td>97.62</td>
<td><strong>97.85</strong></td>
<td><strong>97.85</strong></td>
</tr>
<tr>
<td>trade</td>
<td>75</td>
<td>98.36</td>
<td>98.36</td>
<td>98.36</td>
</tr>
<tr>
<td>ship</td>
<td>36</td>
<td>99.13</td>
<td>99.13</td>
<td>99.13</td>
</tr>
<tr>
<td>grain</td>
<td>10</td>
<td><strong>99.91</strong></td>
<td><strong>99.91</strong></td>
<td><strong>99.91</strong></td>
</tr>
</tbody>
</table>
m-noisy-or± is never worse than m-noisy-or+ which is a desirable property of a good generalization of noisy-or.

From the tested models the best performing model is the logistic regression model. Its accuracy is the best for five classes in the binary case and for one class in the multivariate case. m-noisy-or± has the best performance for two classes.

Remark. We did not perform comparisons with several other classifiers that were already applied to the Reuters text collection since our goal was not to find the best performing classifier for this dataset. We used this dataset to illustrate that the suggested generalizations of noisy-or are sensible and improve the performance of the standard noisy-or.

7. EXAMPLES OF LEARNED MODELS

In this section we use the class interest to illustrate the benefits of treating the features as multivalued. In the first example we present the binary noisy-or model and in the second the multivalued noisy-or model. In the third example we present the binary GLM model and in the fourth the multivalued GLM model. In the fifth example we present the binary GLM model and in the fourth the multivalued GLM model. All models contain only features significant at level 0.1.

Example 1 (b-noisy-or± model for the interest class). In Figure 3 we present the structure of the noisy-or model for the interest class (in the examples we do not make the deterministic part explicit). All variables are binary, taking values 0 or 1. The leaky
definition as

\[
P(Class.interest = 0 | Rate = r, Lend = \ell , Prime = p) = (p_1)^r (p_2)^\ell (p_3)^p p_0 ,
\]

where \( r \in \{0,1\} \) is the state of feature Rate, \( \ell \in \{0,1\} \) is the state of feature Lend, and \( p \in \{0,1\} \) is the state of feature Prime. The values of parameters \( p_1, p_2, p_3 \) were estimated to be

\[
\begin{align*}
p_1 &= \exp(\beta_1) = \exp(-0.355480596) \doteq 0.7008365 \\
p_2 &= \exp(\beta_2) = \exp(-0.237486879) \doteq 0.7886072 \\
p_3 &= \exp(\beta_3) = \exp(-0.397284307) \doteq 0.6721429
\end{align*}
\]

Fig. 3. Noisy-or model for the interest class.
and the leaky parameter $p_0 = \exp(\beta_0)$ was estimated to be

$$p_0 = \exp(\beta_0) = \exp(-0.004363394) \approx 0.9956461 .$$

This model has the accuracy 96.67%. Models for noisy-or+ and noisy-or± are the same since no features with significant negative influence were found.

**Example 2** (m-noisy-or± model for the interest class). In Figure 4, we present the structure of the multivalued noisy-or model for the interest class. The variable $Rate$ takes

![rate prime leaky cause class.interest](image)

Fig. 4. m-noisy-or model for the interest class.

values from the set $\{0, 1, \ldots, 34\}$, variable $Prime$ takes values from the set $\{0, 1, \ldots, 17\}$. The leaky cause has fixed state 1. The conditional probability for the class interest is defined as

$$P(Class.interest = 0|Rate = r, Prime = p) = (p_1)^r (p_2)^p p_0 ,$$

where $r \in \{0, 1, \ldots, 34\}$ is the state of feature $Rate$ and $p \in \{0, 1, \ldots, 17\}$ is the state of feature $Prime$. The values of parameters $p_1$ and $p_2$ were estimated to be

$$p_1 = \exp(\beta_1) = \exp(-0.174466109) \approx 0.8399053$$
$$p_2 = \exp(\beta_2) = \exp(-0.326624088) \approx 0.7213549$$

and the leaky parameter $p_0 = \exp(\beta_0)$ was estimated to be

$$p_0 = \exp(\beta_0) = \exp(-0.004392911) \approx 0.9956167 .$$

Models for m-noisy-or+ and m-noisy-or± are the same since no features with significant negative influence were found.

Despite the fact that this model has less features than the noisy-or model from Example 1 its accuracy is higher since $97.85\% > 96.67\%$. This example illustrates how we can benefit from multivalued features.

**Example 3** (b-GLM-log model for the interest class). GLM-log with the binary features is specified by following formula:

$$\log E(1 - Class.interest|Rate = r, Lend = \ell, Prime = p) = \beta_0 + \beta_1 r + \beta_2 \ell + \beta_3 p .$$
Parameters of the model with the binary features are:

\[
\begin{align*}
\beta_0 &= 0.00114928 \\
\beta_1 &= -0.38059771 \\
\beta_2 &= -0.41347338 \\
\beta_3 &= -0.52342439
\end{align*}
\]

We cannot transform this model to any equivalent model in the family of generalized noisy-or models since the value of \(\exp(\beta_0) = 1.0011499\) is more than one. The accuracy is 96.80%.

**Example 4** (m-GLM-log model for the interest class). GLM-log with the multivalued features is specified by following formula:

\[
\log E(1 - \text{Class.interest}|Rate = r, Fed = f, Prime = p) = \beta_0 + \beta_1 r + \beta_2 p .
\]

Parameters of the model with the multivalued features are:

\[
\begin{align*}
\beta_0 &= -0.002860378 \\
\beta_1 &= -0.173693342 \\
\beta_2 &= -0.408505856
\end{align*}
\]

We can transform this model to an equivalent model in the family of generalized noisy-or models since all values of \(\beta\) are from the interval \((-\infty, 0)\). However, since the learning methods are different the parameters of m-noisy-or differ slightly from parameters of the m-GLM-log model.

The accuracy is 97.62%. Note that, despite m-GLM-log has less features than b-GLM-log, its accuracy is higher.

**Example 5** (b-logistic model for the interest class). Logistic regression model with the binary features is specified by following formula:

\[
\logit(P(\text{Class.interest} = 0|Rate = r, Bank = b, Lend = \ell, Prime = p)) = \beta_0 + \beta_1 r + \beta_2 b + \beta_3 \ell + \beta_4 p .
\]

The parameters of the model are:

\[
\begin{align*}
\beta_0 &= 5.739984 \\
\beta_1 &= -3.907559 \\
\beta_2 &= -1.511631 \\
\beta_3 &= -1.185854 \\
\beta_4 &= -2.353052
\end{align*}
\]

and its accuracy is 96.80%.
A generalization of the noisy-or model

Example 6 (m-logistic model for the interest class). Logistic regression model with the multivalued features is specified by following formula:

$$\text{logit}(P(Class.\text{interest} = 0|Rate = r, Lend = \ell, Prime = p)) = \beta_0 + \beta_1 r + \beta_2 \ell + \beta_3 p.$$ 

The parameters of the model with the multivalued features are:

$$\begin{align*}
\beta_0 &= 4.4795102 \\
\beta_1 &= -1.0021844 \\
\beta_2 &= -0.6589065 \\
\beta_3 &= -2.1107686
\end{align*}$$

and its accuracy is 97.35%. Despite m-logistic has less features than b-logistic, its accuracy is higher.

8. ONE STEP FURTHER: A MULTIVALUED GRADED CHILD VARIABLE

A natural generalization of the noisy-or model to multivalued parents and a multivalued child is the noisy-max. See [4] for a discussion of different versions of noisy-max. Assume that $Y$ takes states $y \in \{0, 1, \ldots, m\}$, where $m = \max\{m_1, \ldots, m_n\}$. The generalizations discussed in [4] require $m \sum_{i=1}^{n} m_i$ model parameters. However, following the idea presented in this paper, we can also generalize the proposed generalized noisy-or to multivalued child variables so that the number of model parameters is still $n$ or $n + 1$ if we consider also the leaky cause.

We can generalize formulas (5) and (4) for a multivalued variable $Y$ as follows. Let

$$P(Y \leq y|X_1 = x_1, \ldots, X_n = x_n) = \begin{cases} 0 & \text{if } y < 0 \\ (p_L)^R(m-y) \prod_{i=1}^{n} (p_i)^R(x_i-y) & \text{if } y = 0, 1, \ldots, m, \end{cases}$$

where $R$ is the ramp function:

$$R(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0. \end{cases}$$

and the values of CPTs can be obtained for $y = 0, 1, \ldots, m$ as

$$P(Y = y|X_1 = x_1, \ldots, X_n = x_n) = P(Y \leq y|X_1 = x_1, \ldots, X_n = x_n) - P(Y \leq y - 1|X_1 = x_1, \ldots, X_n = x_n)$$

$$= \prod_{i=0}^{n} (p_i)^R(x_i-y) - H(y) \prod_{i=0}^{n} (p_i)^R(x_i-y+1),$$

where $x_0 = m$, $p_0 = p_L$, and $H$ is the step function:

$$H(y) = \begin{cases} 0 & \text{if } y \leq 0 \\ 1 & \text{if } y > 0. \end{cases}$$
Let \((y, x)\) denote a data vector with \(x = (x_0, x_1, \ldots, x_n)\) and \(x_0\) being fixed to value \(m\). Assume both \(x_i\) and \(y\) taking values from the set \(\{0, 1, \ldots, m\}\). Denote the vector of probability parameters of the model as \(\beta = (\log(p_0), \log(p_1), \ldots, \log(p_n))\) with \(p_0 = p_L\). Then we can write formula (11) as

\[
P(y|x) = \prod_{i=0}^{n} \exp(\beta_i R(x_i - y)) - H(y) \prod_{i=0}^{n} \exp(\beta_i R(x_i - y + 1))
\]

\[
= \exp \left( \beta^T R(x - y) \right) - H(y) \exp \left( \beta^T R(x - y + 1) \right), \tag{13}
\]

where \(y = (y, \ldots, y)\).

Remark. Note that if \(y = m\) then \(R(m - y) = 0\) and for \(i = 1, \ldots, n\) it holds that \(R(x_i - y) = 0\). This implies that for all \(x = (x_1, \ldots, x_n)\) it holds that

\[
P(Y \leq m|X_1 = x_1, \ldots, X_n = x_n) = 1,
\]

which guarantees that

\[
\sum_{y=1}^{m} P(Y = y|X_1 = x_1, \ldots, X_n = x_n) = 1.
\]

If we set \(m = 1\) then we get the generalized noisy-or from the previous sections of this paper. If \(m = 1\) and \(m_i = 1, i = 1, \ldots, m_i\) then we get the standard noisy-or. Our generalization from this section is a special case of the graded noisy-max model proposed by Díez in \[3, Definition 2\]. The main difference is that our model requires only \(n + 1\) parameters (including the leaky cause \(p_L\)). We will refer to our model as to simple graded noisy-max.

### Learning of simple graded noisy-max

We learn the simple graded noisy-max model using a constraint learning method with the constraint \(\beta \leq 0\). This constraint guarantees that for all possible values of \(x\) in formula (13) we get probability values. To maximize the conditional log-likelihood we use a quasi-Newton method with box constraints \[2\] implemented in R \[12\]. The method uses the formula (17) for the gradient derived in Appendix B. To avoid infinite numbers we added a small positive value \((10^{-10})\) to the numerator and the denominator in formula (17). The algorithm was initialized at ten different values of \(\beta\) generated randomly from interval \([-1, +1]\).

Since we did not have real data for a learning experiment we created data artificially. We used a simple graded noisy-max model with three parents each taking states from \(\{0, 1, 2\}\). The child variable was also ternary and taking states from \(\{0, 1, 2\}\). During the data generation process we repeated following two steps. First, we randomly generated a parent configuration \(x\) (with uniform probability) and then we randomly generated value \(y\) with probabilities defined by formula (13) for the given \(x\). We performed experiments with different numbers of generated data vectors. We used the learning algorithm described above to learn the parameters \(\beta\) of the simple graded noisy-max.
In Figure 5 we present the results of experiments. The left hand side plot describes the dependence of the total sum of the Kullback-Leibler divergence of the conditional probability tables of the true model and the model learned from the generated data on the size of the training dataset. We compare two models: the simple graded noisy-max (the full line) and the full conditional probability table (CPT) computed from relative frequencies of data vectors in generated data (the dashed line). Note the logarithmic scale of the vertical axis. On the right hand side we present the dependence of Euclidean distance of parameter vectors $\beta$ of the true and the learned models on the size of the training dataset.

Fig. 5. Dependence of the quality of the learned simple graded noisy-max models on the size of the training dataset.

From the plots we can see that training datasets with a relatively small size ($\sim 1000$) are sufficient for learning the parameters of the simple graded noisy-max. The parameters of the full CPT are much harder to be estimated well. In the experiment, a general CPT required 54 parameters to be learned while to specify the simple graded noisy-max we learned only 4 parameters. This explains the difference in the learning speed.

9. CONCLUSIONS

In this paper we proposed generalizations of the popular noisy-or model to multivalued parent variables and allowed parent variables with both positive and negative influence on child variable. We showed that our generalizations are subfamilies of the family of Generalized Linear Models with the logarithmic link function. We used iteratively reweighted least squares method to learn models from the GLM-log and logistic regression families. For the restricted subfamilies corresponding to generalized noisy-or
models we used a quasi-Newton method with box constraints. In the experiments with
the Reuters text collection the generalized noisy-or models performed equally well or
better than standard noisy-or. They represent handy generalizations of noisy-or for real
applications with multivalued variables and/or with parent variables being a mixture of
variables having either positive or negative influence on their child variable.

ACKNOWLEDGEMENT

I am grateful to Remco Bouckaert from The University of Auckland, New Zealand
for his suggestion to consider generalizations of noisy-or classifier [15] to multivalued
variables and to anonymous reviewers for their comments and suggestions that helped
to improve this paper. This work was supported by the Czech Science Foundation
through the project 13-20012S. A preliminary version of this paper appeared in the
proceedings of the 16th Czech-Japan Seminar on Data Analysis and Decision Making
under Uncertainty [16].

REFERENCES

Morgan Kaufmann, 1993.
2007.
1989.
[10] R. A. Miller, F. E. Fasarie, and J. D. Myers. Quick medical reference (QMR) for diagnostic
A generalization of the noisy-or model


A. THE CONDITIONAL LOG-LIKELIHOOD AND ITS GRADIENT FOR THE BINARY RESPONSE

Let \((y, x)\) denote a data vector from a training dataset \(D\), where \(x = (x_0, x_1, \ldots, x_n)\) with \(x_0\) being fixed to value 1. Assume \(x_i\) taking values from the set \(\{0, 1, \ldots, m\}\) and \(y\) from the set \(\{0, 1\}\). Under the generalized noisy-or model the probability of \(y\) given \(x\) is defined as:

\[
\log P(y = 0|x) = \beta^T x
\]

\[
\log P(y = 1|x) = \log(1 - \exp(\beta^T x)) .
\]

The conditional log-likelihood of data given this model is

\[
\ell(\beta) = \sum_{(y, x) \in D} (1 - y)\beta^T x + y \log(1 - \exp(\beta^T x)) . \tag{14}
\]

The gradient of the conditional log-likelihood is the vector of partial derivatives with respect to \(\beta\):

\[
\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{(y, x) \in D} x - yx \frac{\exp(\beta^T x)}{1 - \exp(\beta^T x)}
\]

\[
= \sum_{(y, x) \in D} \frac{(1 - y)x - (1 - y)x \exp(x^T \beta) - yx \exp(\beta^T x)}{1 - \exp(\beta^T x)}
\]

\[
= \sum_{(y, x) \in D} x \frac{(1 - y) - \exp(\beta^T x)}{1 - \exp(\beta^T x)} . \tag{15}
\]
B. THE CONDITIONAL LOG-LIKELIHOOD AND ITS GRADIENT FOR THE GRADED RESPONSE

The logarithm of the probability of observing a datavector \((y, x)\) from \(D\) under the simple graded noisy-max (defined in Section 8) is

\[
P(y|x) = \exp \left( \beta^T R(x - y) \right) - H(y) \exp \left( \beta^T R(x - y + 1) \right)
\]

The conditional log-likelihood of data \(D\) given the model is

\[
\ell(\beta) = \sum_{(y, x) \in D} \log \left( \exp \left( \beta^T R(x - y) \right) - H(y) \exp \left( \beta^T R(x - y + 1) \right) \right)
\]

(16)

The gradient of the conditional log-likelihood is the vector of partial derivatives with respect to \(\beta\):

\[
\frac{\partial \ell(\beta)}{\partial \beta} = \sum_{(y, x) \in D} \frac{\exp \left( \beta^T R(x - y) \right) R(x - y) - H(y) \exp \left( \beta^T R(x - y + 1) \right) R(x - y + 1)}{\exp \left( \beta^T R(x - y) \right) - H(y) \exp \left( \beta^T R(x - y + 1) \right)}
\]

(17)

(Received ????)

Jiří Vomlel, Institute of Information Theory and Automation of the AS CR, Pod vodárenskou věží 4, 18208 Praha 8, Czech Republic
Faculty of Management, University of Economics, Prague, Jarošovská 1117/II, 37701 Jindřichův Hradec, Czech Republic
e-mail: vomlel@utia.cas.cz