Selecting feature based models*

Amir Jalalirad & Tjalling Tjalkens

Eindhoven University of Technology, Eindhoven, The Netherlands

2013 Information Theory and Applications Workshop

February 15, 2013

*This work is in part supported by ENIAC Joint Undertaking, grant 270707-2, EnLight.

abstract

- A binary classification problem with a feature vector of high dimensionality. *Example: Spam mail filters.*
- A Bayesian approach requires the estimation the probability of a feature vector given the class of the object.
  Due to the size of the feature vector this is an unfeasible task.
- A useful approach: split the feature space into several (conditionally) independent subspaces.
  This reduces the number of model parameters.
  Simple example: Naive Bayes filter.
- This results in a new problem: find the "best" subdivision.
- We consider a weighing approach:
  - that will perform (asymptotically) as good as the best subdivision.
  - and still have a manageable complexity.
- Use the same efficient computation structure to find the Maximum-Likelihood model.

Problem setting

- Object $\mathcal{O}$ belongs to a class c and is described by a vector of features, f^k.

$$\mathcal{O} = \left(c, f^k\right).$$

- Assume binary classes, binary features, and independent objects drawn from

$$P(\mathcal{O}) = P(c)P(f^k|c).$$

- The model classes partition the feature vector into conditionally independent parts.

$$P(f^k|c) = P(f_1, f_2|c)P(f_3|c)\cdots$$

Model class description

Let the feature vector index set $\{1, 2, \ldots, k\}$ be written as $\mathfrak{F}$.
A model $\mathcal{M}$ is described by a number of subsets $\mathfrak{s}_1, \mathfrak{s}_2, \ldots, \mathfrak{s}_g$.
These subsets have a (sub-)partitioning property defined by

$$\mathfrak{s}_i \bigcap \mathfrak{s}_j = \emptyset; \quad \text{if } i \neq j,$$

$$\bigcup_{i=1}^{g} \mathfrak{s}_i \subset \mathfrak{F}$$

Apart from the subsets a model also contains parameters $\underline{\theta}$ that describe the probabilities of the feature vector given the class.

A subset $\mathfrak{s}$ selects some features from the feature vector f^k.
If $\mathfrak{s} = \{i_1, i_2, \ldots, i_s\}$ then this selection is written as

$$f^{\mathfrak{s}} = f_{i_1}, f_{i_2}, \ldots, f_{i_s}.$$

The model $\mathcal{M} = (\mathfrak{s}_1, \mathfrak{s}_2, \ldots, \mathfrak{s}_g)$ defines the following conditional feature vector probability.

$$P(f^k|c, \mathcal{M}, \underline{\theta}) = \prod_{i=1}^{g} P(f^{\mathfrak{s}_i}|c, \underline{\theta}).$$

Assuming unknown parameters we can use e.g. a Beta prior (Dirichlet-$\frac{1}{2}$) based averaging process (Krichevsky-Trofimov) and write $P_e(f^k|c, \mathcal{M})$ and $P_e(f^{\mathfrak{s}_i}|c)$ and so on.

An example

A possible model class results if the subsets $\mathfrak{s}_i$ for each model form a complete partitioning. e.g. let $k = 3$, then the following models belong to this class.

Model	feature probability
$\mathcal{M}_1 = (\{1\}, \{2\}, \{3\})$	$P_e(f_1\|c)P_e(f_2\|c)P_e(f_3\|c),$
$\mathcal{M}_2 = (\{1, 2\}, \{3\})$	$P_e(f_1, f_2\|c)P_e(f_3\|c),$
$\mathcal{M}_3 = (\{1, 3\}, \{2\})$	$P_e(f_1, f_3\|c)P_e(f_2\|c),$
$\mathcal{M}_4 = (\{2, 3\}, \{1\})$	$P_e(f_2, f_3\|c)P_e(f_1\|c),$
$\mathcal{M}_5 = (\{1, 2, 3\})$	$P_e(f_1, f_2, f_3\|c).$

Sequence probability

- A sequence of objects $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$,

$$\mathcal{O}_i = \left(c_{(i)}, f^k_{(i)} \right).$$

- Independent object drawing, so

$$P_e(f^k_{(1)}, f^k_{(2)}, \ldots, f^k_{(n)} | c_{(1)}, c_{(2)}, \ldots, c_{(n)}) = \prod_{i=1}^{n} P_e(f^k_{(i)} | c_{(i)}).$$

- Shorthand notation for this sequence probability

$$P_e(f^k_{(1\ldots n)} | c_{(1\ldots n)}).$$

Four model classes

We can consider ordered and unordered partitions.
A partition is ordered if the subsets contain only consecutive feature indices, so

$$\mathfrak{s}_{=} \{a, a+1, a+2, \ldots, a+b\}.$$

In an unordered partition the subsets can contain any combination of feature indices.
For both cases the partitioning can be complete (full partitioning) or incomplete (sub-partitioning).

$$\bigcup_{i=1}^{g} \mathfrak{s}_i \begin{cases} = \mathfrak{F}; & \text{full partitioning} \\ \subset \mathfrak{F}; & \text{sub-partitioning} \end{cases}$$

Class I: ordered features and full partitioning

Here I wish to compute

$$P_e(f^k_{(1\ldots n)}|c_{(1\ldots n)}) = \sum_{i=1}^{|\mathfrak{M}|} P(\mathcal{M}_i) P_e(f^k_{(1\ldots n)}|c_{(1\ldots n)}, \mathcal{M}_i),$$

where the model subsets contain consecutive indices and form a (full) partition of $\mathfrak{F}$.

If $\mathfrak{s}_i$ is a subset in the model, then we call $P_e(f^{\mathfrak{s}_i}_{(1\ldots n)}|c_{(1\ldots n)})$ the corresponding basic probability. Obviously there are

$$\frac{1}{2}k(k+1)$$

basic probabilities which we must compute using an update rule, e.g.

$$P_e(f^{\mathfrak{s}_i}_{(1\ldots n)}|c_{(1\ldots n)}) = P_e(f^{\mathfrak{s}_i}_{(1\ldots n-1)}|c_{(1\ldots n-1)})\frac{\#(f^{\mathfrak{s}_i}_{(n)}|c_{(n)}) + \frac{1}{2}}{\#(c_{(n)}) + 2^{|\mathfrak{s}_i|} - 1}.$$

(see Krichevsky-Trofimov or the use of the Dirichlet-$\frac{1}{2}$ prior.)

Brute force computation cost

- How much work is it to compute the mixture, assuming that the *basic* probabilities are known?
- There are $\binom{k-1}{g-1}$ models with g groups.
  For g groups we must cut $g-1$ wires out of $k-1$ wires in a string of k beads.
- The number of models is

$$\sum_{g=1}^{k-1} \binom{k-1}{g-1} = 2^{k-1} - 1.$$

- A model with g groups requires $g-1$ multiplications of basic probabilities. So the total number of multiplications is

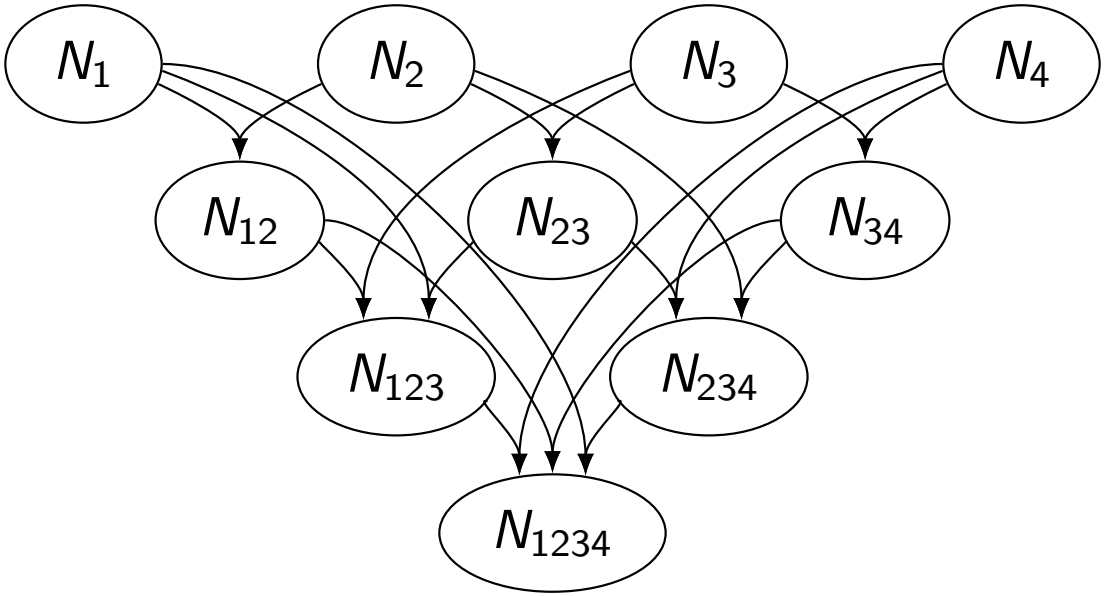
$$\sum_{g=1}^{k-1} \binom{k-1}{g-1} (g-1) = (k-1)\left(2^{k-2} - 1\right).$$

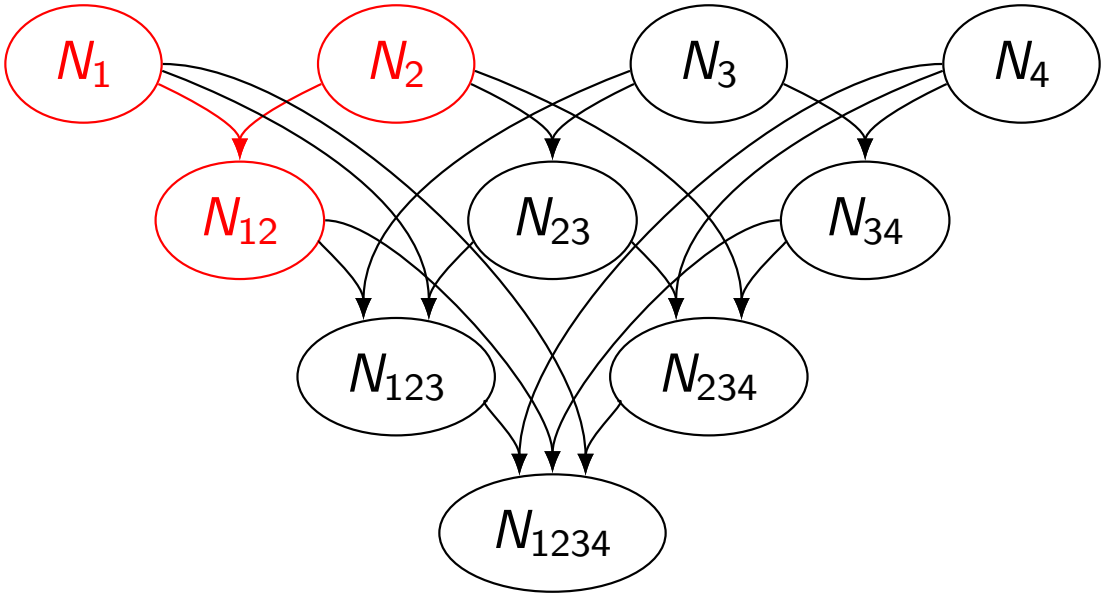
Network method

- We shall make use of the distributive law of algebra.
- node $N(\mathfrak{s})$ that equals a combination of all ordered and full partition sub-model probabilities. e.g. let $\mathfrak{s} = \{1, 2, 3\}$ then

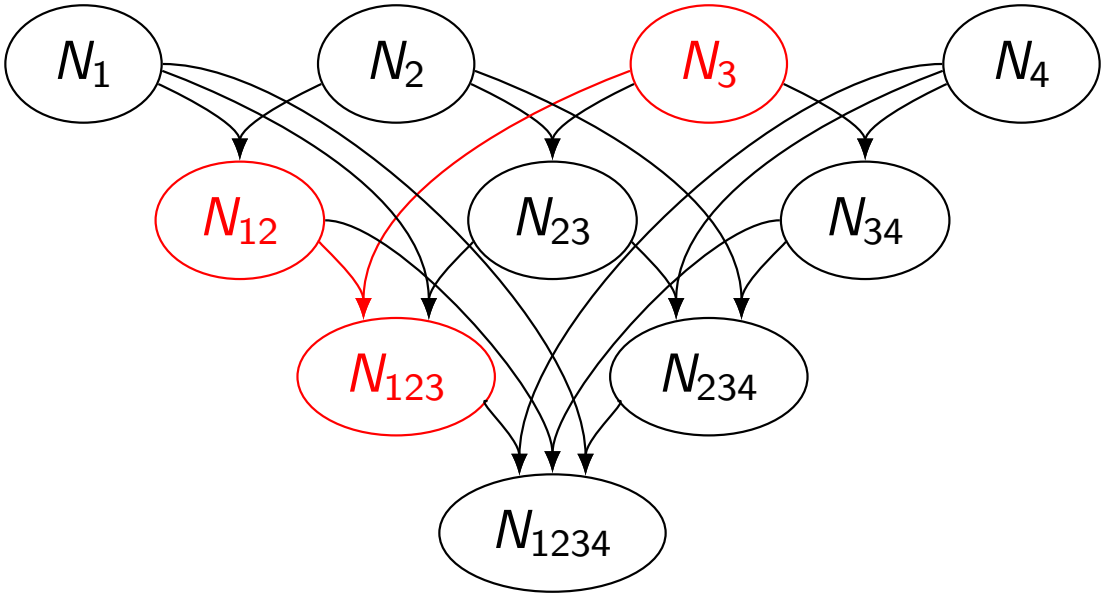
$$\begin{aligned} N(\{1,2,3\}) = \alpha_1 P_e(f_{(1\ldots n)}^{\{1,2,3\}} | c_{(1\ldots n)}) + \alpha_2 P_e(f_{(1\ldots n)}^{\{1,2\}} | c_{(1\ldots n)}) P_e(f_{(1\ldots n)}^{\{3\}} | c_{(1\ldots n)}) \\ + \alpha_3 P_e(f_{(1\ldots n)}^{\{1\}} | c_{(1\ldots n)}) P_e(f_{(1\ldots n)}^{\{2,3\}} | c_{(1\ldots n)}) \\ + \alpha_4 P_e(f_{(1\ldots n)}^{\{1\}} | c_{(1\ldots n)}) P_e(f_{(1\ldots n)}^{\{2\}} | c_{(1\ldots n)}) P_e(f_{(1\ldots n)}^{\{3\}} | c_{(1\ldots n)}). \end{aligned}$$

  The α's are to be selected in a appropriate or convenient way.
- Short-hand N_{123} for $N(\{1,2,3\})$ and P_{123} for $P_e(f_{(1\ldots n)}^{\{1,2,3\}} | c_{(1\ldots n)})$ and so on.

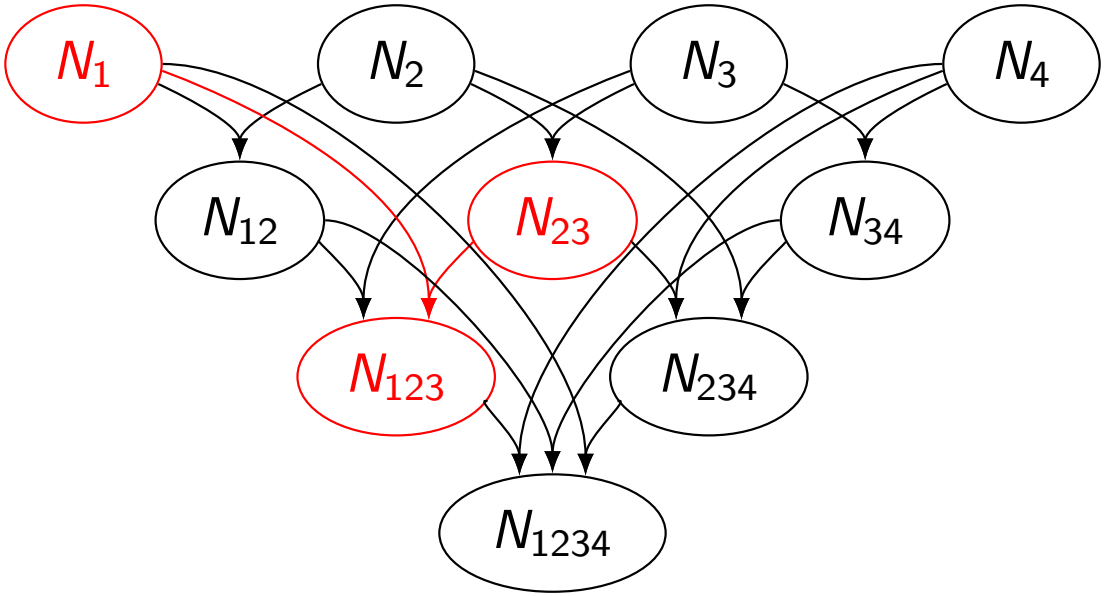

N_1
N_2
N_3
N_4
N_{12}
N_{23}
N_{34}
N_{123}
N_{234}
N_{1234}



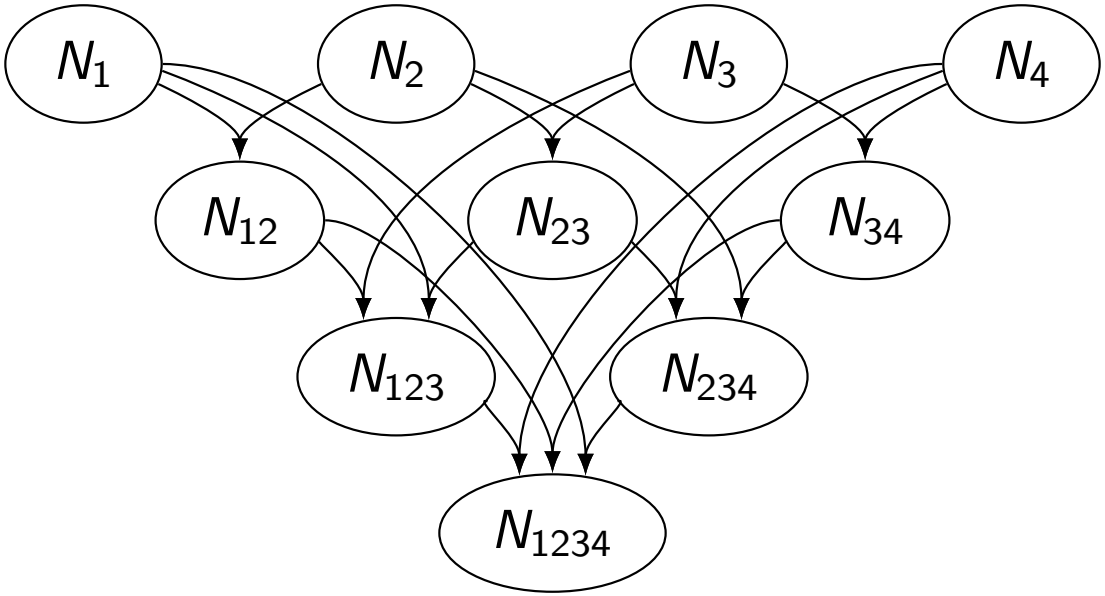
$$N_1 = P_1; \quad N_2 = P_2.$$
$$N_{12} = P_{12} + N_1 \cdot N_2$$
$$= P_{12} + P_1 P_2$$



$$N_{12} = P_{12} + P_1 P_2; \quad N_3 = P_3.$$
$$N_{123} = P_{123} + N_{12} \cdot N_3 + N_1 \cdot N_{23}$$
$$= P_{123} + P_{12} P_3 + P_1 P_2 P_3 + N_1 \cdot N_{23}.$$



$$N_1 = P_1; \quad N_{23} = P_{23} + P_2 P_3.$$
$$N_{123} = P_{123} + N_{12} \cdot N_3 + N_1 \cdot N_{23}$$
$$= P_{123} + N_{12} \cdot N_3 + P_1 P_{23} + P_1 P_2 P_3.$$



$$N_{123} = P_{123} + P_{12}P_3 + P_1P_{23} + 2P_1P_2P_3.$$

$$N(\mathfrak{F}) = N_{1234} = P_{1234} + P_1P_{234} + P_{12}P_{34} + P_{123}P_4 + 2P_1P_2P_{34} + 2P_1P_{23}P_4 + 2P_{12}P_3P_4 + 5P_1P_2P_3P_4.$$

Recall: k is the length of the feature vector; g is the number of subsets $\mathfrak{s}$ in a model $\mathcal{M}$.
I am interested in

- $T_1(k)$: The total number of terms in $N(\mathfrak{F})$.
  This is the normalization factor needed to turn $N(\mathfrak{F})$ into a probability.
- $M_1(g)$ the multiplicity of a model with g subsets in $N(\mathfrak{F})$.
  Together $\frac{M_1(g)}{T_1(k)}$ define the model prior.
- $W_1(k)$ the number of additions and multiplications needed to compute $N(\mathfrak{F})$.

$T_1(k)$

$T_1(k)$ is described by the recursion

$$T_1(k) = 1 + \sum_{i=1}^{k-1} T_1(i)\, T_1(k-i); \quad T_1(1) = 1.$$

This results in

$$T_1(k) = \sum_{i=0}^{k-1} C_i \binom{k-1}{i}.$$

Here C_i is the i^{th} Catalan number, $C_i = \frac{1}{i+1}\binom{2i}{i}$.

$$T_1 : 1 \quad 2 \quad 5 \quad 15 \quad 51 \cdots$$

$M_1(g)$

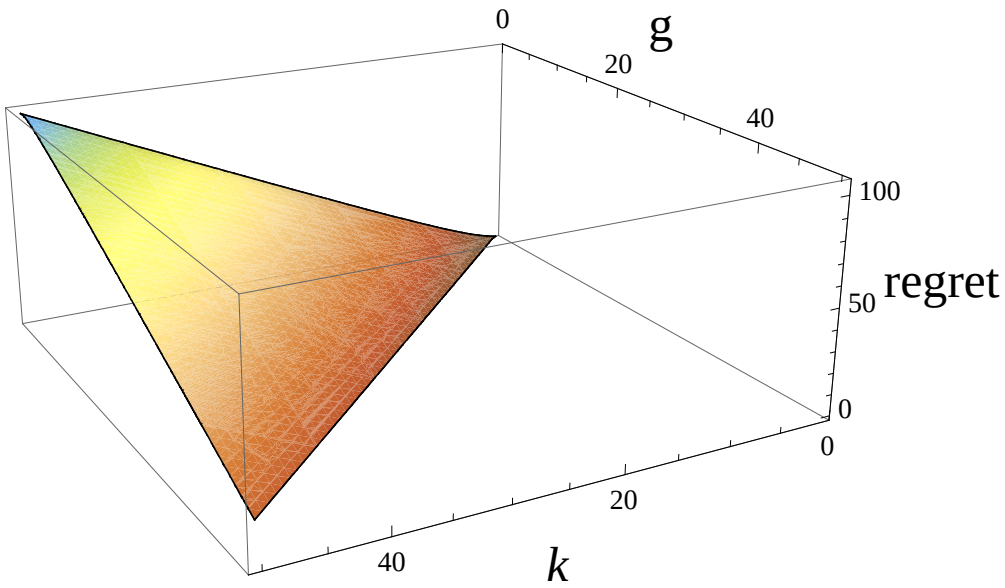
$M_1(g)$ is described by the recursion

$$M_1(g) = \sum_{i=1}^{g-1} M_1(i) M_1(g-i) = C_{g-1}.$$

$$M_1 : 1 \quad 1 \quad 2 \quad 5 \quad 14 \cdots$$

This results in a prior that is larger for models with fewer free parameters. A contribution to the log-regret or model selection cost is written as

$$r_{1,\mathcal{M}}(k,g) = -\log_2 \frac{M_1(g)}{T_1(k)}.$$



$$r_{1,\mathcal{M}}(1,1) = 0,$$
$$r_{1,\mathcal{M}}(50,50) = 16.263,$$
$$r_{1,\mathcal{M}}(50,1) = 104.982.$$

$W_1(k)$

Recall brute force

There are $2^{k-1}-1$ models.

There are $\binom{k-1}{g-1}$ models with g groups.

A model with g groups requires $g-1$ multiplications. So we have a total of $(k-1)\left(2^{k-2}-1\right)$ multiplications and $2^{k-1}-2$ additions.

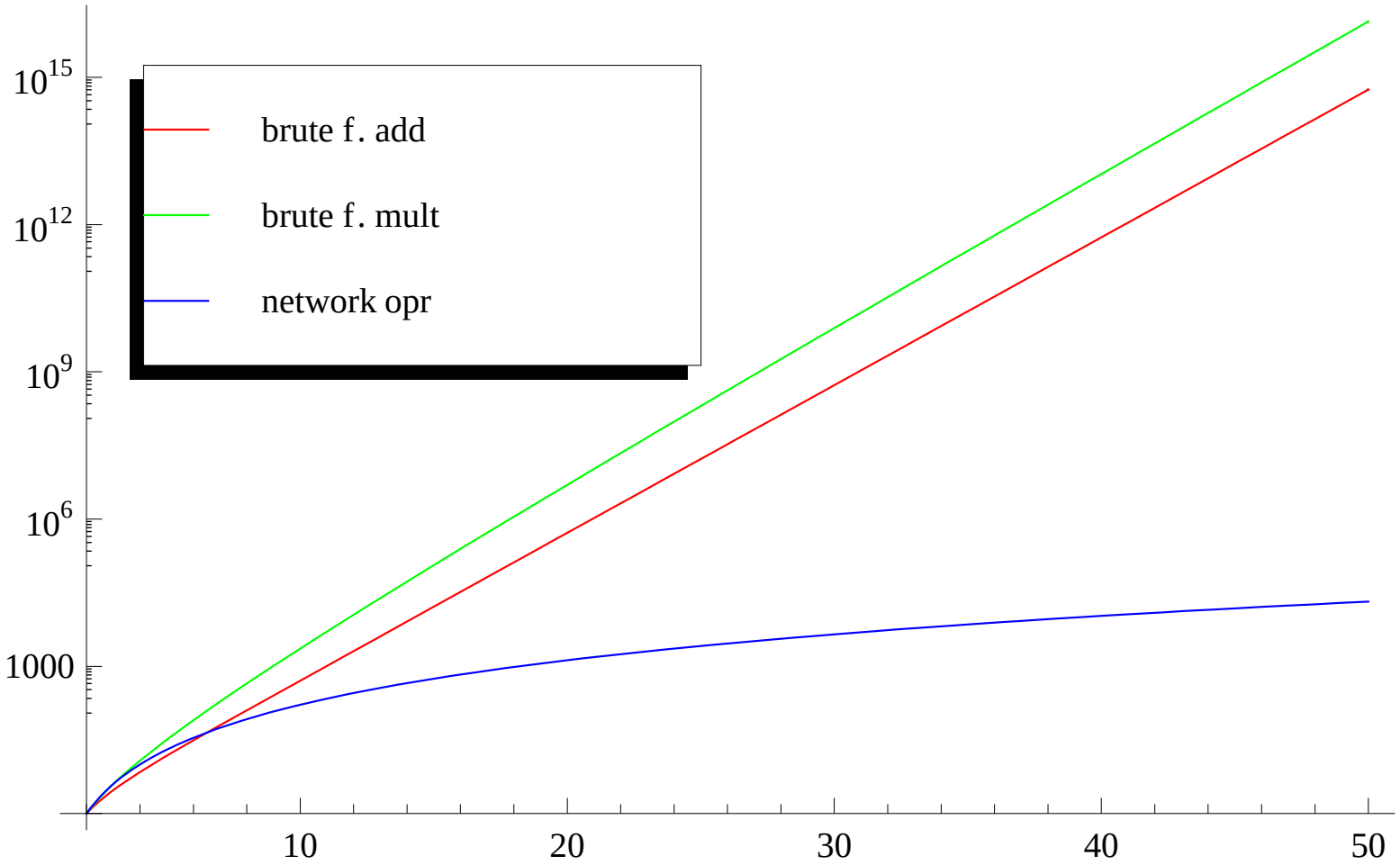
Network model

A node is at level i if it's subset has size i.

So there are $k-(i-1)$ nodes at level i.

A node at level i performs $(i-1)$ additions and $(i-1)$ multiplications.

The total number of multiplications (and additions) equals $1/6(k-1)k(k+1)$.

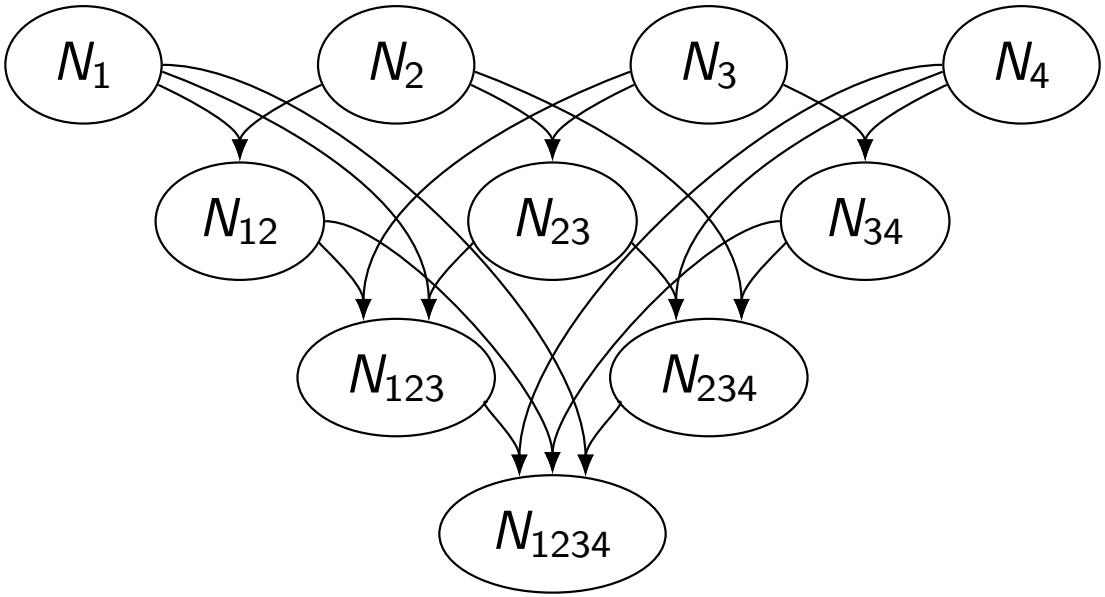

brute f. add
brute f. mult
network opr
10^{15}
10^{12}
10^{9}
10^{6}
1000
10
20
30
40
50

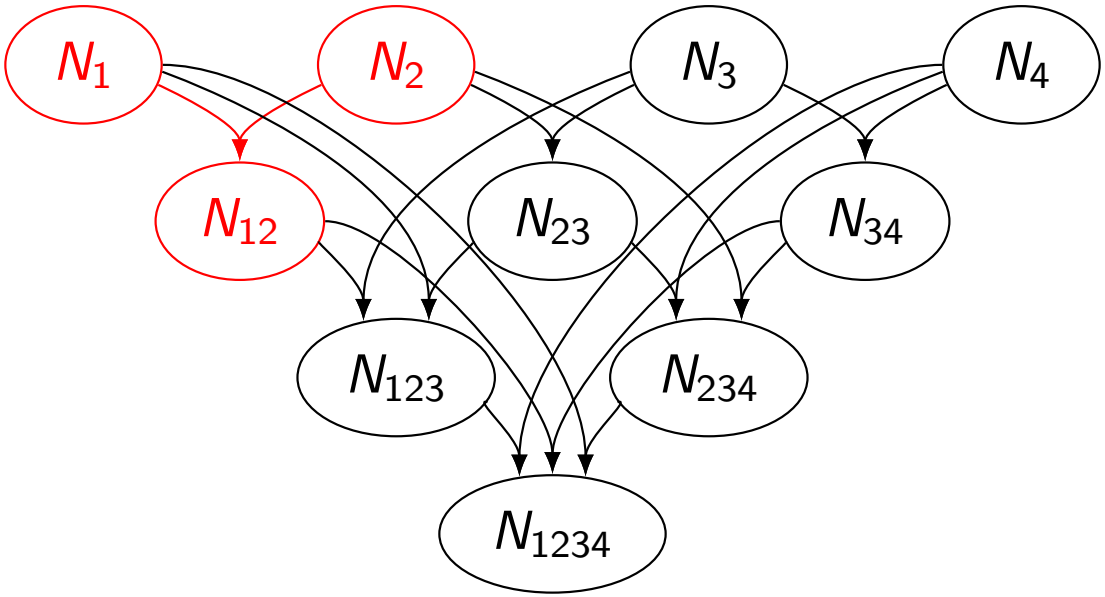
Class II: ordered features and sub-partitioning

It seems reasonable to request a model class that allows some features not to be used at all. I might have added all features I could think of, not knowing how relevant they are.
So I wish to compute

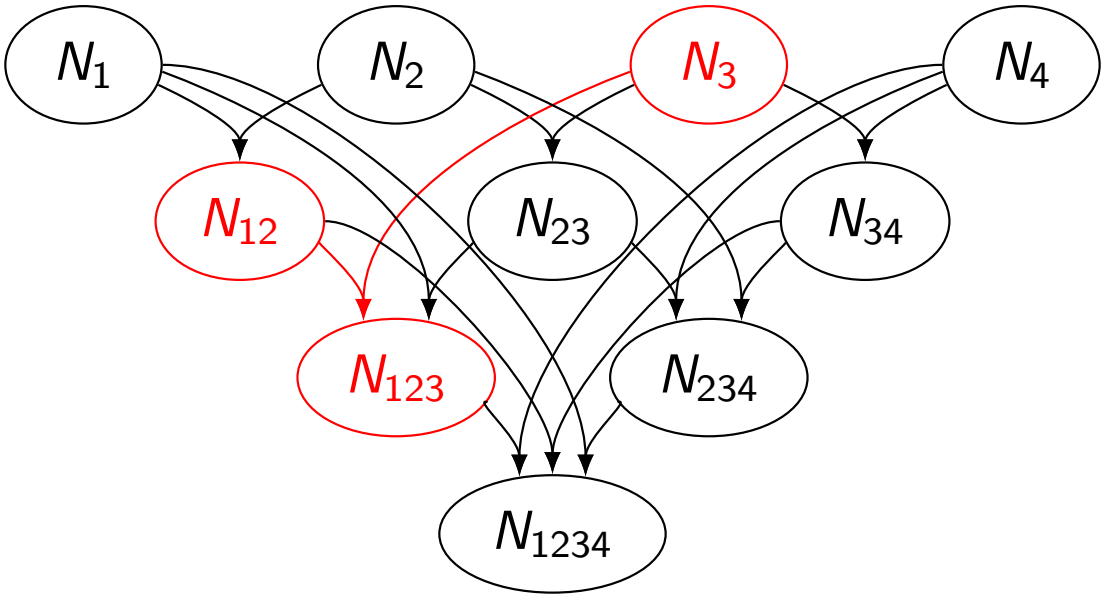
$$P_e(f^k_{(1\ldots n)}|c_{(1\ldots n)}) = 2^{-n\alpha} \sum_{i=1}^{|\mathfrak{M}|} P(\mathcal{M}_i) P_e(f^{\mathfrak{s}_i}_{(1\ldots n)}|c_{(1\ldots n)}, \mathcal{M}_i),$$

where the model subsets contain consecutive indices and form a (partial) partition of $\mathfrak{F}$ and α is the number of unused features.
As we shall see, this can be accommodated in a simple way using the method for Class I.
I use the additional short-hand $Z = 2^{-n}$.


N_1
N_2
N_3
N_4
N_{12}
N_{23}
N_{34}
N_{123}
N_{234}
N_{1234}

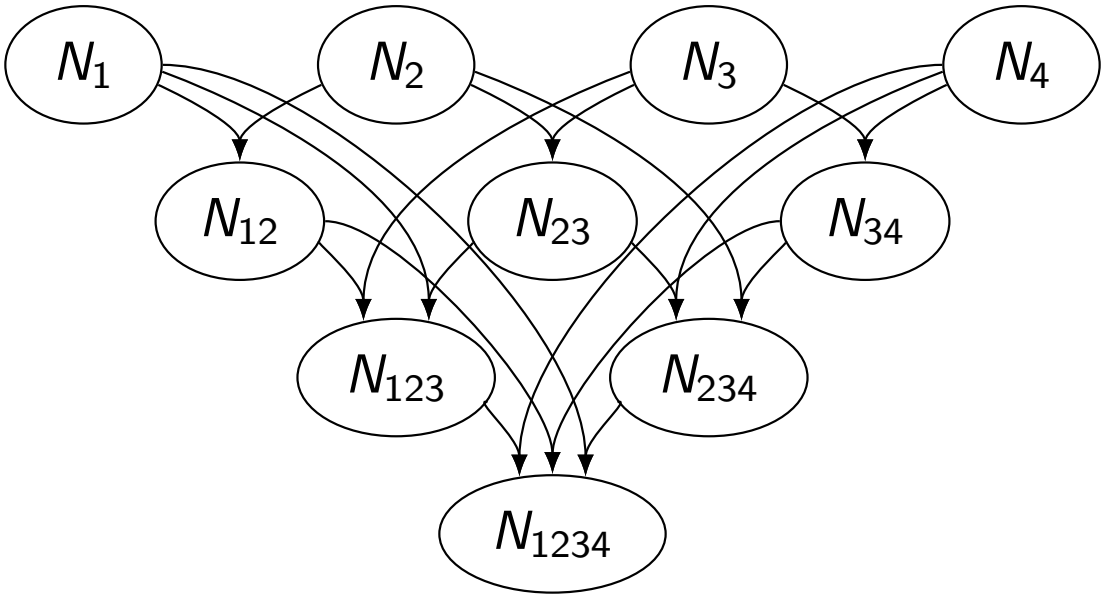


$$N_1 = P_1 + Z; \quad N_2 = P_2 + Z.$$
$$N_{12} = P_{12} + N_1 \cdot N_2$$
$$= P_{12} + P_1 P_2 + P_1 Z + P_2 Z + Z^2.$$

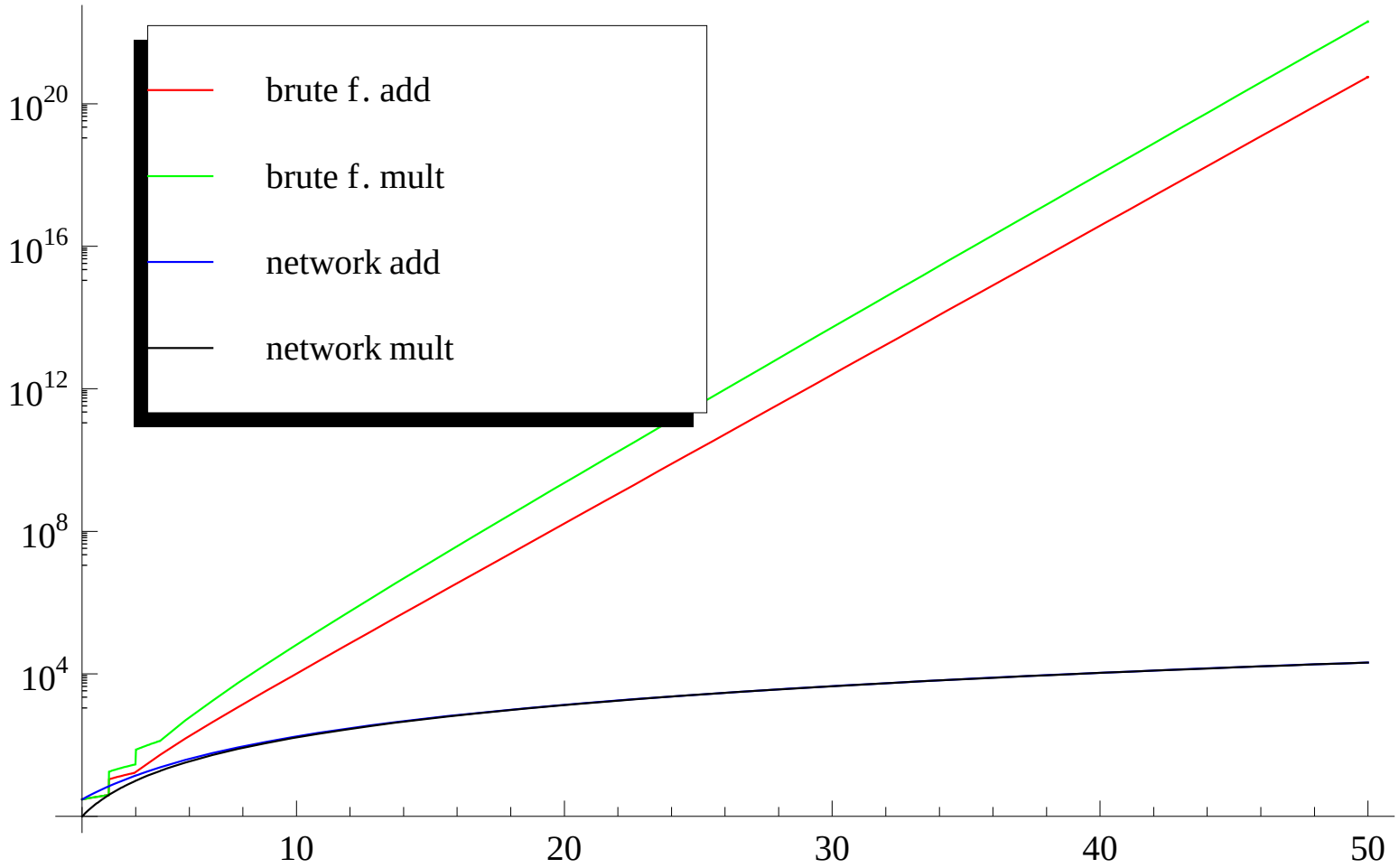


$$N_{12} = P_{12} + P_1P_2 + P_1Z + P_2Z + Z^2; \quad N_3 = P_3 + Z.$$

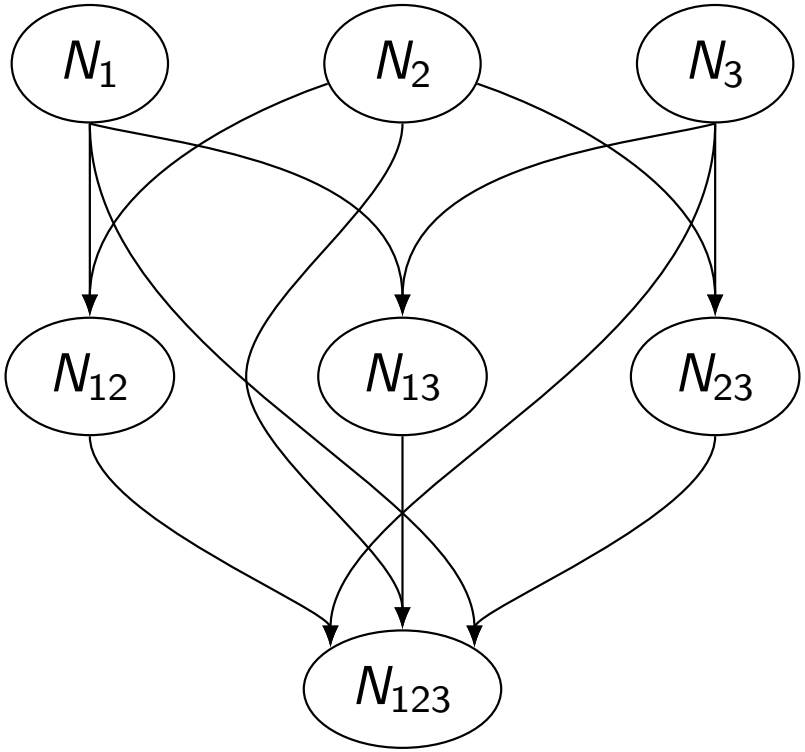
$$\begin{aligned} N_{123} &= P_{123} + N_{12} \cdot N_3 + N_1 \cdot N_{23} \\ &= P_{123} + P_{12}P_3 + P_1P_2P_3 + N_1 \cdot N_{23} + \\ &\quad P_{12}Z + P_1P_2Z + P_1P_3Z + P_2P_3Z + P_1Z^2 + P_2Z^2 + P_3Z^2 + Z^3. \end{aligned}$$



$$N_{123} = P_{123} + P_{12}P_3 + P_1P_{23} + 2P_1P_2P_3 + P_{12}Z + P_{23}Z$$
$$2P_1P_2Z + 2P_1P_3Z + 2P_2P_3Z + 2P_1Z^2 + 2P_2Z^2 + 2P_3Z^2 + 2Z^3.$$
$$N(\mathfrak{F}) = N_{1234} = P_{1234} + P_1P_{234} + P_{12}P_{34} + P_{123}P_4 + 2P_1P_2P_{34} + 2P_1P_{23}P_4$$
$$+ 2P_{12}P_3P_4 + 5P_1P_2P_3P_4 + P_{123}Z + P_{234}Z + 2P_1P_{23}Z + \ldots + 5Z^4.$$

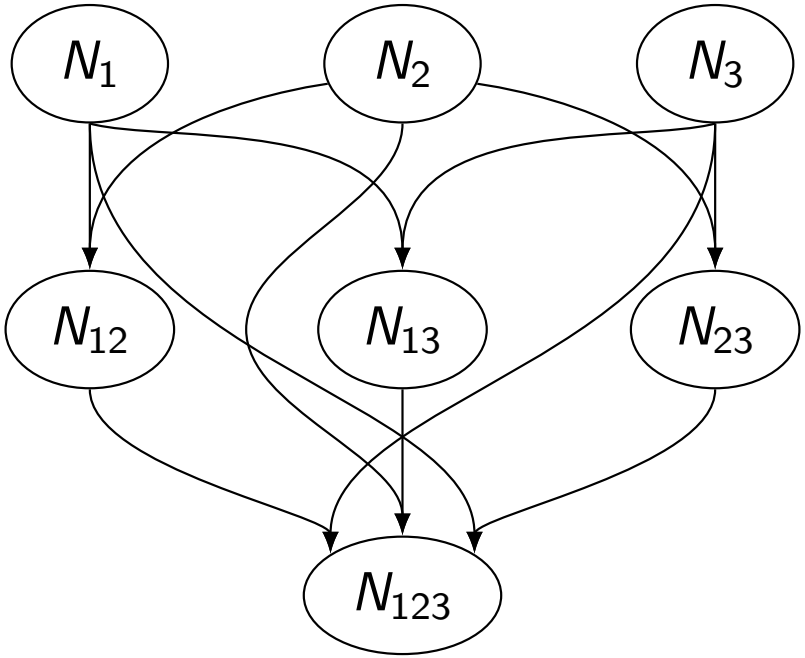

brute f. add
brute f. mult
network add
network mult
10^{20}
10^{16}
10^{12}
10^{8}
10^{4}
10
20
30
40
50

Class III: unordered features and full partitioning



$$N(\mathfrak{F}) = N_{123} = P_{123} + P_1 P_{23} + P_2 P_{13} + P_3 P_{12} + 2 P_1 P2 P_3.$$

Class IV: unordered features and sub-partitioning



$$N_1 = P_1 + Z; \quad N_2 = P_2 + Z; \quad \cdots$$

$$\begin{aligned} N(\mathfrak{F}) = N_{123} &= P_{123} + P_1P_{23} + P_2P_{13} + P_3P_{12} + 3P_1P2P_3 \\ &+ P_{12}Z + P_{13}Z + P_{23}Z + 3P_1P_2Z + 3P_1P_3Z + 3P_2P_3Z \\ &+ 3P_1Z^2 + 3P_2Z^2 + 3P_3Z^2 + 3Z^3. \end{aligned}$$

The ML model

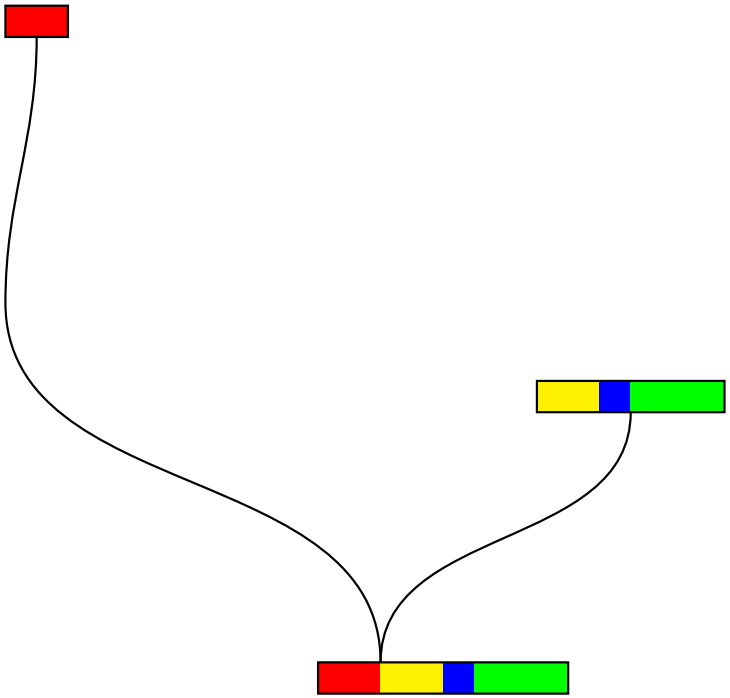
Consider the Class I model class.
We wish to find

$$\mathcal{M}_* = \arg \max_{\mathcal{M} \in \mathfrak{M}} P_e(f^k_{(1\ldots n)} | c_{(1\ldots n)}, \mathcal{M}).$$

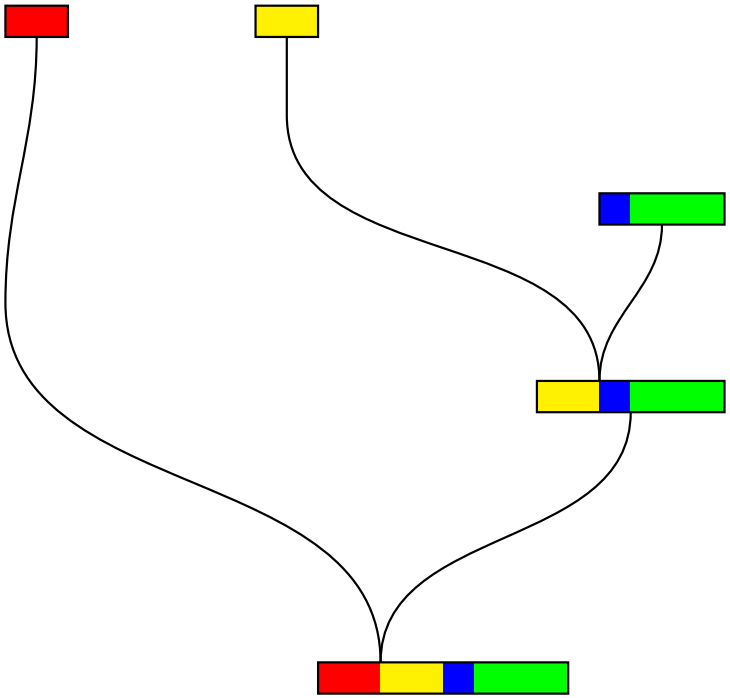
Can this be done in a similar efficient fashion?

Network ML construction

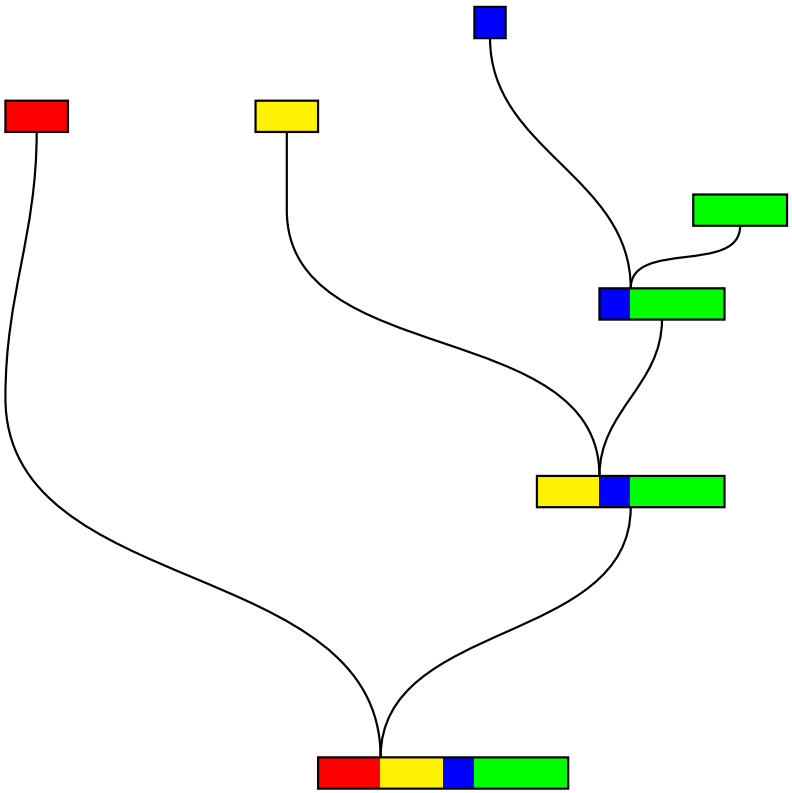
Network ML construction



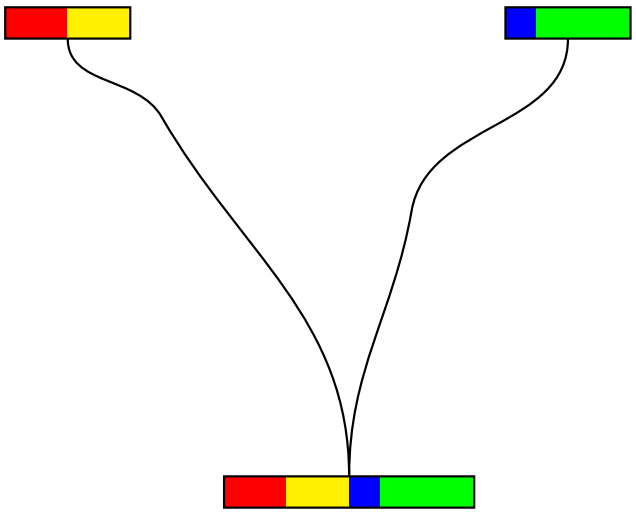
Network ML construction



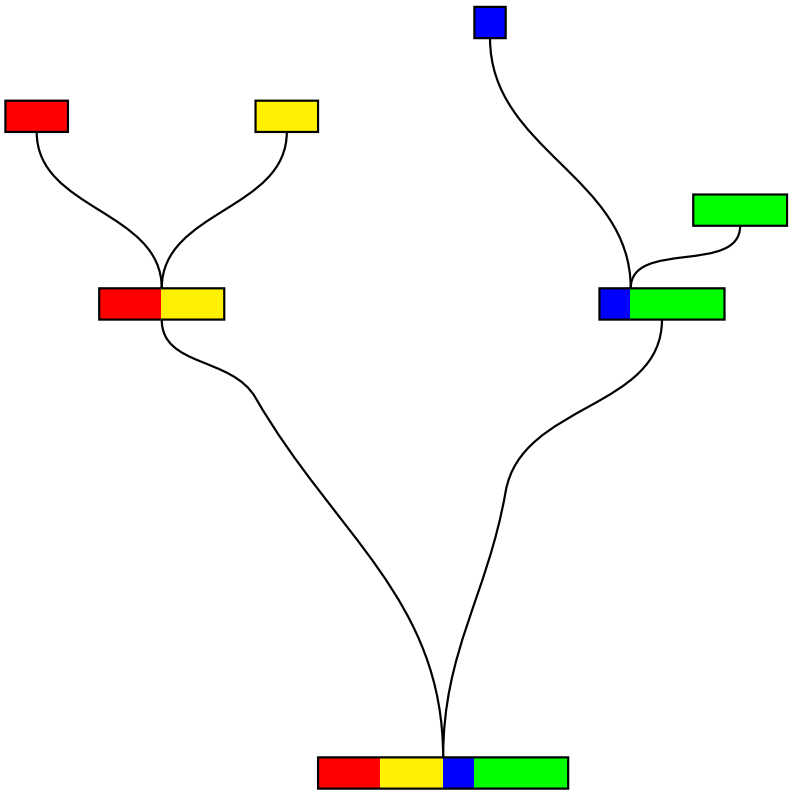
Network ML construction



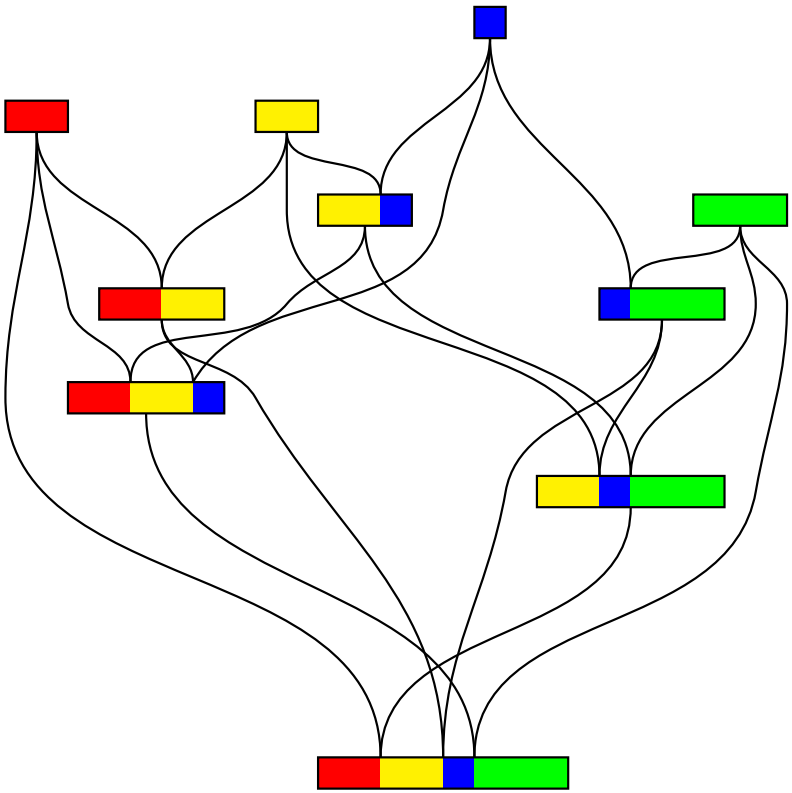
Network ML construction



Network ML construction



Network ML construction



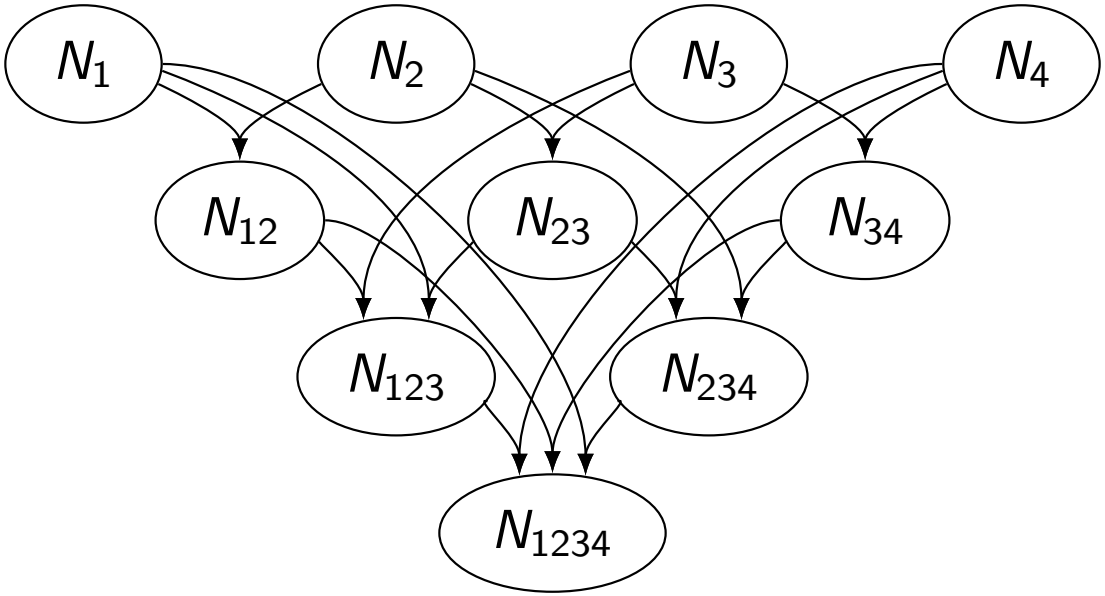
Network node operations

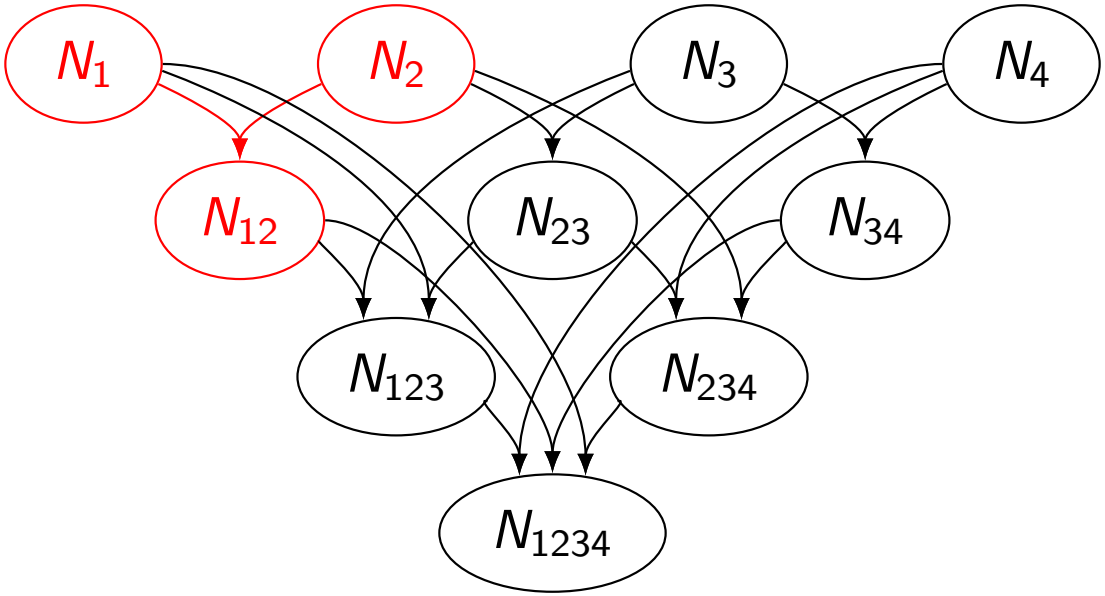
Thus we can achieve the goal of finding the ML model when we replace the node operation

$$N_{\mathfrak{s}} = P_{\mathfrak{s}} + \sum_{\substack{\text{incoming} \\ \text{arrow pairs}}} N_{\mathfrak{s}(\text{left})} N_{\mathfrak{s}(\text{right})}$$

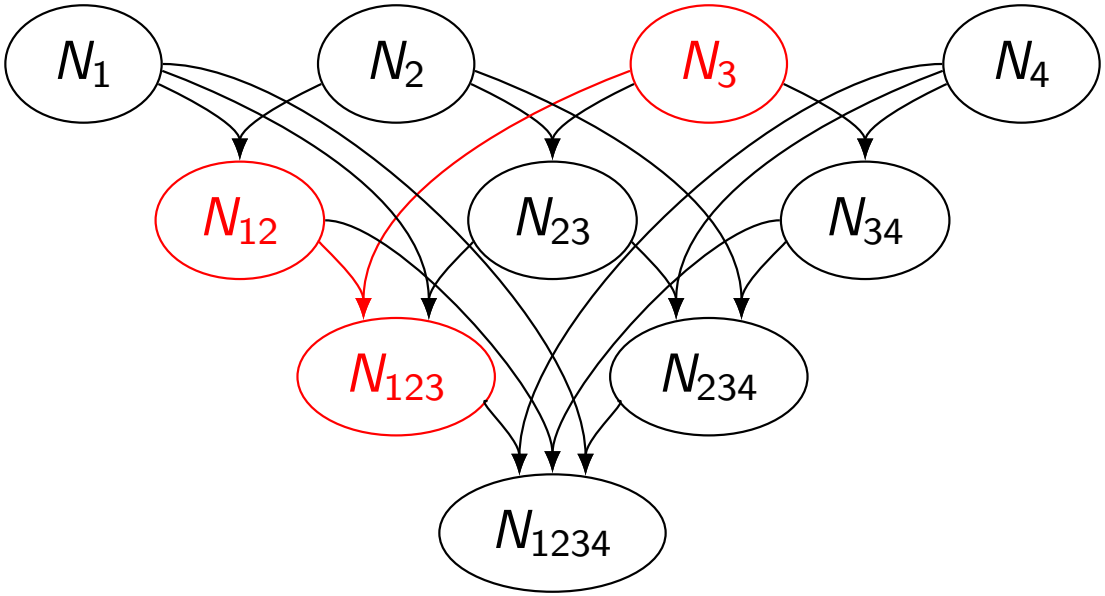
by

$$N_{\mathfrak{s}} = \max \left\{ P_{\mathfrak{s}}, \max_{\substack{\text{incoming} \\ \text{arrow pairs}}} N_{\mathfrak{s}(\text{left})} N_{\mathfrak{s}(\text{right})} \right\}.$$

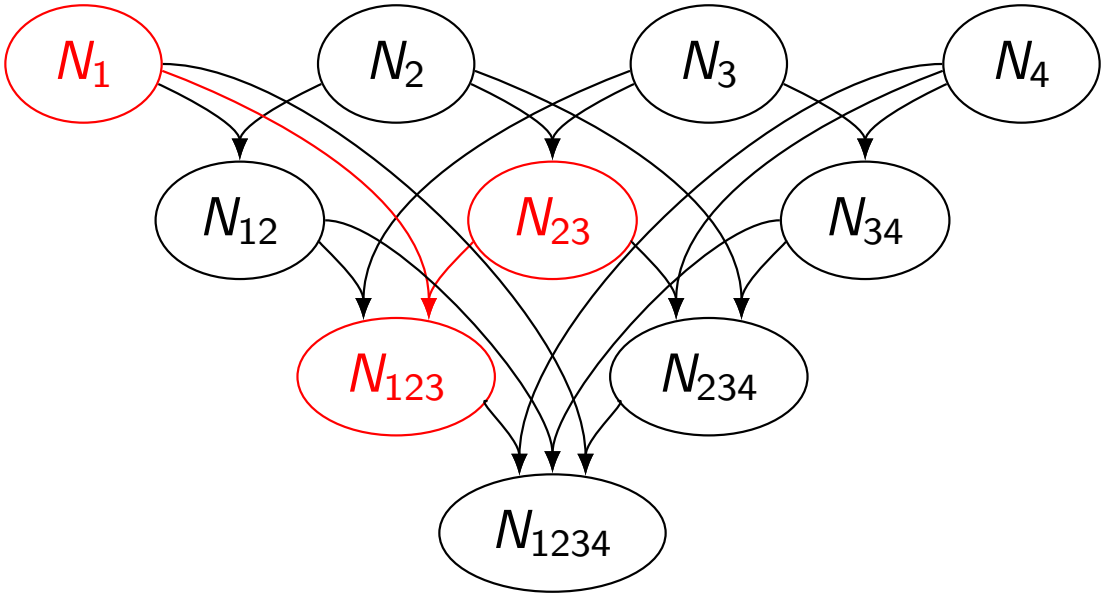

N_1
N_2
N_3
N_4
N_{12}
N_{23}
N_{34}
N_{123}
N_{234}
N_{1234}



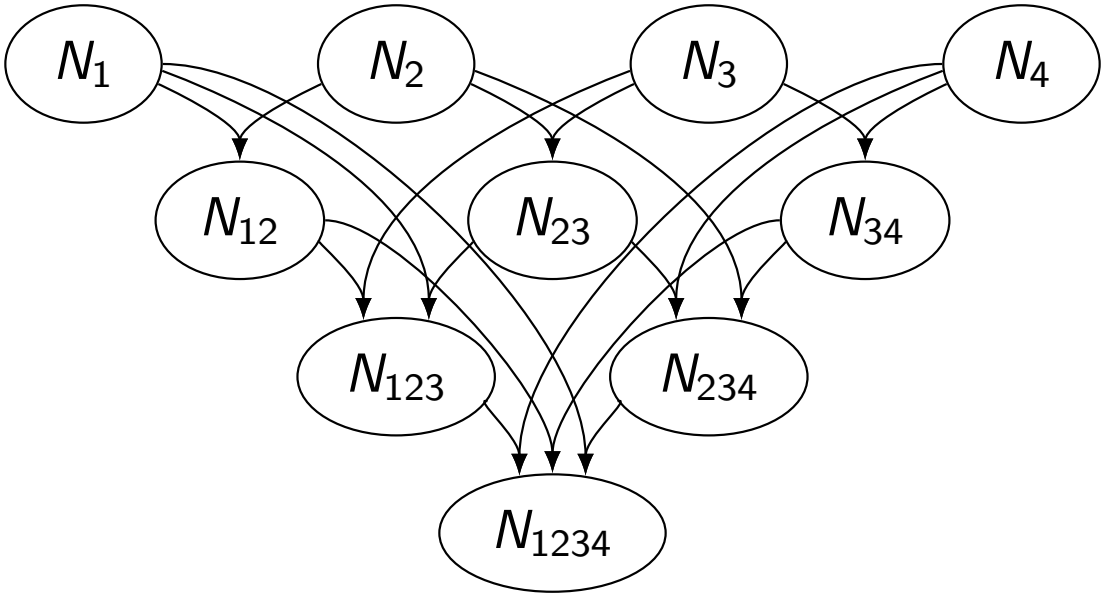
$$N_1 = P_1; \quad N_2 = P_2.$$
$$N_{12} = \max\{P_{12}, N_1 \cdot N_2\}$$
$$= \max\{P_{12}, P_1 P_2\}$$



$$N_{12} = \max\{P_{12}, P_1 P_2\}; \quad N_3 = P_3.$$
$$N_{123} = \max\{P_{123}, N_{12} \cdot N_3, N_1 \cdot N_{23}\}$$
$$= \max\{P_{123}, P_{12} P_3, P_1 P_2 P_3, N_1 \cdot N_{23}\}.$$



$$N_1 = P_1; \quad N_{23} = \max\{P_{23}, P_2 P_3\}.$$
$$\begin{aligned} N_{123} &= \max\{P_{123}, N_{12} \cdot N_3, N_1 \cdot N_{23}\} \\ &= \max\{P_{123}, N_{12} \cdot N_3, P_1 P_{23}, P_1 P_2 P_3\}. \end{aligned}$$



$$N_{123} = \max\left\{P_{123}, P_{12}P_3, P_1P_{23}, P_1P_2P_3\right\}.$$

$$N(\mathfrak{F}) = N_{1234} = \max\left\{P_{1234}, P_1P_{234}, P_{12}P_{34}, P_{123}P_4, P_1P_2P_{34}\right.$$
$$\left.P_1P_{23}P_4, P_{12}P_3P_4, P_1P_2P_3P_4\right\}.$$

Conclusions and remarks

- The network method gives a substantial reduction in number of operations.
- The method is flexible and can be used for several model classes.
- The class and feature variables need not be binary. The influence of the alphabet size is only in the 'basic' probabilities and thus in the complexity of calculating these probabilities and the log-regret.
- The unordered features classes are still very complex.

Also, real valued features and classes are possible as the result only deals with the separation of features into conditionally independent groups.