Algorithm SMMB

Convention: Sets of variables (*i.e.* ensembles) will be denoted in bold font.

Algorithm 1 Stochastic Multiple Markov blanket
SMMB($\mathbf{r}, \mathbf{t}, \mathbf{X}, \mathbf{T}, \mathbf{K}, \mathbf{k}, \mathbf{m}, \boldsymbol{\alpha}$): **consensus_MB**

INPUT:

$\mathbf{r}$, maximal number of Markov blankets ($|\mathbf{MBs}|$) output by the algorithm
$\mathbf{t}$, maximal number of iterations at top level,
$\mathbf{X}$, observed data for genotypes, data matrix of dimension $n * p$, with:
  n, the number of individuals, and
  p, the number of variables
$\mathbf{T}$, observed data for phenotype (vector of dimension n)
$\mathbf{K}$, total number of variables sampled from $\mathbf{X}$ to learn a Markov blanket (top level)
$\mathbf{k}$, number of variables sampled from $\mathbf{K}$ variables (inner level), $\mathbf{k} < \mathbf{K}$
$\mathbf{m}$, maximal number of resamplings (of $\mathbf{k}$ variables) as long as the Markov blanket remains empty
$\boldsymbol{\alpha}$, type I error threshold

OUTPUT:

consensus_MB, a consensus from all learned Markov blankets $\mathbf{MBs}$, $|\mathbf{MBs}| \leq \mathbf{r}$

$\mathbf{MBs} \leftarrow \emptyset$
$i \leftarrow 0$
while ($|\mathbf{MBs}| \leq r$ and i $\leq t$)
  $\mathbf{X}^* \leftarrow$ *sampling_without_replacement*$(K, \mathbf{X})$
  $\mathbf{MB}^* \leftarrow$ *learnMB*$(\mathbf{X}^*, T, k, m, \alpha)$
  if not *empty*$(\mathbf{MB}^*)$ **then** *add*$(\mathbf{MBs}, \mathbf{MB}^*)$ **end if**
  incr(i)
end while
consensus_MB $\leftarrow$ *buildConsensus*$(\mathbf{MBs}, \alpha)$
return consensus_MB

Algorithm 2 learnMB($\mathbf{X}^*, \mathbf{T}, \mathbf{k}, \mathbf{m}, \boldsymbol{\alpha}$): **MB**

OUTPUT:

MB, a Markov blanket, possibly empty

$\mathbf{MB} \leftarrow \emptyset$ /*initialization of candidate Markov blanket*/
$i \leftarrow 0$
repeat
  $\mathbf{S} \leftarrow$ *sampling_without_replacement*$(\mathbf{k}, \mathbf{X}^*)$

  /*Forward step*/
  $\mathbf{s} \leftarrow argmax_{\mathbf{s'} \subseteq \mathbf{S}}\{assoc\_score(\mathbf{s'}, T, \mathbf{MB})\}$
  if not $significant\_indep_{|MB}(\mathbf{s}, T, \mathbf{MB}, \alpha)$ **then**
    $\mathbf{MB} \leftarrow \mathbf{MB} \cup \mathbf{s}$
    /*Backward step*/
    for each $X \in \mathbf{MB}$
      for each $\mathbf{S} \subseteq \mathbf{MB} \setminus \{X\}, \mathbf{S} \neq \emptyset$
        if$(significant\_independence(X, T, \mathbf{S}, \alpha)$ **then** $\mathbf{MB} \leftarrow \mathbf{MB} \setminus \{X\}$; **break end if**
      end for
    end for
  end if
  incr(i)
until ((**not** *empty*(**MB**)) **and** (**MB** *does not change*)) **or** (*empty*(**MB**) **and** $i = m$)
return **MB**

Algorithm 3 buildConsensus($\mathbf{MBs}, \boldsymbol{\alpha}$): **consensus_MB**

INPUT:

$\mathbf{MBs}$, a set of Markov blankets
$\boldsymbol{\alpha}$, type I error threshold

OUTPUT:

consensus_MB, a consensus from all Markov blankets $\mathbf{MBs}$

consensus_MB $\leftarrow \bigcup_{\mathbf{MB} \in \mathbf{MBs}} \mathbf{MB}$ /* initialization of consensus */

/*Backward step*/
for each $X \in$ **consensus_MB**
  for each $\mathbf{S} \subseteq$ **consensus_MB** $\setminus \{X\}, \mathbf{S} \neq \emptyset$
    if ($significant\_independence(X, T, \mathbf{S}, \alpha)$) **then**
      consensus_MB $\leftarrow$ **consensus_MB**$\setminus \{X\}$; **break**
    end if
  end for
end for

return consensus_MB

Algorithm 4 $assoc\_score(\mathbf{S_1}, \mathbf{T}, \mathbf{S_2})$: *maximal score*

INPUT:

$\mathbf{S_1}$, a set of variables
$\mathbf{T}$, a variable, $\mathbf{T} \notin \mathbf{S_1}$
$\mathbf{S_2}$, a set of variables, $\mathbf{T} \notin \mathbf{S_2}$

1: $score_{max} \leftarrow -\infty$
2: **for each** $X \in \mathbf{S_1}$
3:   $stat \leftarrow stat\_independence\_test(X, T, \mathbf{S_2} \cup (\mathbf{S_1} \setminus \{X\}))$
4:   **if** ($stat > score_{max}$) **then** $score_{max} \leftarrow stat$; $memorize$(p-value) **end if**
5: **end for**
6: **return** $score_{max}$

Comments

Algorithms 2 (learnMB) and 3 (buildConsensus) use function $significant\_independence(X, T, \mathbf{S}, \alpha)$, with $X \neq T$ *and* $X \notin \mathbf{S}$. Function $significant\_independence(X, T, \mathbf{S}, \alpha)$ runs a G-test of independence between variables X and T, conditional on set $\mathbf{S}$. We denote $stat_o$ the observed statistic returned by the test for $Stat$, the random variable with an unknown distribution. For the G-test, the distribution P_{H_0} of $Stat$ under the hypothesis of independence H_0 is known, it is the $Chi - Squared$ law. Thus, the function $significant\_independence$ returns true if and only if $P_{H_0}(Stat \geq stat_o) \geq \alpha$.

Algorithm 4 ($assoc\_score(\mathbf{S_1}, T, \mathbf{S_2}, \alpha)$) iteratively runs function $stat\_independence\_test(X, T, \mathbf{S_2} \cup (\mathbf{S_1} \setminus \{X\}))$, $\forall\ X \in \mathbf{S_1}$. The test used by $stat\_independence\_test$ is the G-test, to assess independence between X and T, conditional on $\mathbf{S_2} \cup (\mathbf{S_1} \setminus \{X\})$. The higher the test statistic, the higher the dependence.

In Algorithm 2 (learnMB), function $assoc\_score(\mathbf{s}', T, \mathbf{MB})$ (line 5) is run on all subsets $\mathbf{s}'$ of $\mathbf{S}$, including subset $\mathbf{s}$, the future candidate. In particular, in Algorithm 4 ($assoc\_score$), function $assoc\_score(\mathbf{s}, T, \mathbf{MB})$ will return the maximal G-test statistic computed for some $X^* \in \mathbf{s}$ by $stat\_independence\_test(X^*, T, \mathbf{MB} \cup (\mathbf{s} \setminus \{X^*\}))$. Let us denote test* this test. In Algorithm 2 (learnMB), function $significant\_indep_{|_{MB}}$ $(\mathbf{s}, T, \mathbf{MB}, \alpha)$ (line 6) is run after function $assoc\_score(\mathbf{s}, T, \mathbf{MB})$ (in particular) has been run. The p-value of test* was memorized during the execution of $assoc\_score(\mathbf{s}, T, \mathbf{MB})$ (Algorithm 4, line 4); this p-value is then available. In learnMB, this p-value is directly used by the function $significant\_indep_{|_{MB}}(\mathbf{s}, T, \mathbf{MB}, \alpha)$, to assess independence of $\mathbf{s}$ et T, given the current MB.