Pseudo-code for the Multi-Modal Mean-Fields algorithm

Algorithm summarises the operations to split one mode into two, or, in other words, to obtain the two additional constraints which are used to define the two newly created subsets. Algorithm 2 summarises the operations to obtain the Multi-Modal Mean Field Distribution by constructing the whole Tree.

In Algorithm 2, $ConstraintTree$, is taken to be a Tree in the form of a list of constraints, one for each branching-point, or leaf,—except for the root—, in a breadth first order. The function `pathto`$(nNode)$, returns the set of indices corresponding to the branching points on the path to the branching point, or leaf with index $nNode$, including index $nNode$ itself.

Algorithm 1: Function:`Split`$(ConstraintList)$

Input:

$E(\mathbf{x})$: An Energy function defined by a CRF;

`SolveMF`$(E, ConstraintList)$: A Mean Field solver with cardinality constraint.;

$Temperatures$: A list of temperatures in increasing order;

$\mathcal{H}_{low}, \mathcal{H}_{high}$: Entropy thresholds for the phase transition. 0.3 and 0.6 here.

C: A cardinality threshold

Output:

$LeftConstraints$: A triplet containing a list of variables, clamped to value, -C

$RightConstraints$: A triplet containing a list of variables, clamped to value, C

$Q^{T_0} \leftarrow$ `SolveMF`(E)
for `T in` $Temperatures$ **do**
  $Q^T \leftarrow$ `SolveMF`$(\frac{E}{T}, ConstraintList)$
  $i_{list} \leftarrow [.]$
  $v_{list} \leftarrow [.]$
  for `index in 1...len`(Q^t)`, v in` $labels$ **do**
    if $\mathbb{1}[\mathcal{H}(q^T_{index}) > 0.6]\mathbb{1}[\mathcal{H}(q^{T_0}_{index}) < 0.3]\mathbb{1}[q^{T_0}_{index,v} > 0.5] = 1$ **then**
      i_{list}`.append(index)`,v_{list}`.append(v)`
    end if
  end for
  if `len`$(i_{list}) > 0$ **then**
    `exit for loop`
  end if
end for
$LeftConstraints = i_{list}, v_{list}, -C$
$RightConstraints = i_{list}, v_{list}, C$
return $LeftConstraints$,$RightConstraints$

Algorithm 2: Compute Multi-Modal Mean Field

Input:

$E(\mathbf{x})$: An Energy function defined on a CRF;

`SolveMF`$(E, ConstraintList)$: A Mean Field solver with cardinality constraint;

`Split`$(ConstraintList)$: Alg. . A function that computes the new constraints.

$NModes$: A target for the number of modes in the Multi-Modal Mean Field

Output:

$Qlist$: A list of Mean Field distributions in the form of a table of marginals

$mlist$: A list of probabilities, one for each mode

$ConstraintTree = [.]$
We first build the tree by adding constraints.
while $nNode < NModes$ **do**
  $ConstraintList = [.]$
  for p `in pathto`$(nNode)$ **do**
    $ConstraintList$`.append(ConstraintTree[p])`
  end for
  $LeftConstraints, RightConstraints \leftarrow$ `Split`$(ConstraintList)$
  $ConstraintTree$.append($LeftConstraints$)
  $ConstraintTree$.append($RightConstraints$)
end while
We now turn to the computation of on MF distribution per leaf.
$Qlist = [.], Zlist = [.], mlist = [.]$
for mode in $0 \dots NModes$ **do**
  $ConstraintList = [.]$
  for p `in pathto`$(mode + NModes - 1)$ **do**
    $ConstraintList$`.append(ConstraintTree[p])`
  end for
  Q,Z $\leftarrow$ `SolveMF`(E,ConstraintList)
  $Qlist$.append(Q)
  $Zlist$.append(Z)
end for
Finally, we compute the mode probabilities.
for $mode$ in $0 \dots NModes$ **do**
  $mlist$.append($\frac{Zlist[mode]}{\sum Zlist}$)
end for
return $Qlist, mlist$