

Contents

1 SetValueOfAssignment.m

```
% SetValueOfAssignment Sets the value of a variable assignment in a factor.
%
% F = SetValueOfAssignment(F, A, v) sets the value of a variable assignment,
% A, in factor F to v. The order of the variables in A are assumed to be the
% same as the order in F.var.
%
% F = SetValueOfAssignment(F, A, v, VO) sets the value of a variable
% assignment, A, in factor F to v. The order of the variables in A are given
% by the vector VO.
%
% Note that SetValueOfAssignment *does not modify* the factor F that is
% passed into the function, but instead returns a modified factor with the
% new value(s) for the specified assignment(s). This is why we have to
% reassign F to the result of SetValueOfAssignment in the code snippets
% shown above.
%
% See also GetValueOfAssignment.m and SampleFactors.m
%
% Copyright (C) Daphne Koller, Stanford University, 2012

function F = SetValueOfAssignment(F, A, v, VO)

if (nargin == 3),
    indx = AssignmentToIndex(A, F.card);
else
    map = zeros(length(F.var), 1);
    for i = 1:length(F.var),
        map(i) = find(VO == F.var(i));
    end;
    indx = AssignmentToIndex(A(map), F.card);
end;

F.val(indx) = v;

end
```

2 NormalizeCPDFactors.m

```
% Copyright (C) Daphne Koller, Stanford University, 2012

function [F] = NormalizeCPDFactors(F)

NumFactors = length(F);
for i=1:NumFactors

    f = F(i);
    dummy.var = f.var(2:end);
    dummy.card = f.card(2:end);
    dummy.val = zeros(1,prod(dummy.card));

    % Now for each joint assignment to parents, renormalize the
    % values for that joint assignment to sum to 1.

    for a=1:length(dummy.val)
```

```

A = IndexToAssignment(a, dummy.card);
Indices = [];
for d=1:f.card(1)
    Indices = [Indices AssignmentToIndex([d A], f.card);];
end
if sum(f.val(Indices)) == 0
    % Set f.val(Indices) to 0
    f.val(Indices) = 0;
else
    f.val(Indices) = f.val(Indices) / sum(f.val(Indices));
end
end

f.val(find(isnan(f.val))) = 0;

F(i) = f;

end

```