Online Algorithms for Sum-Product Networks with Continuous Variables

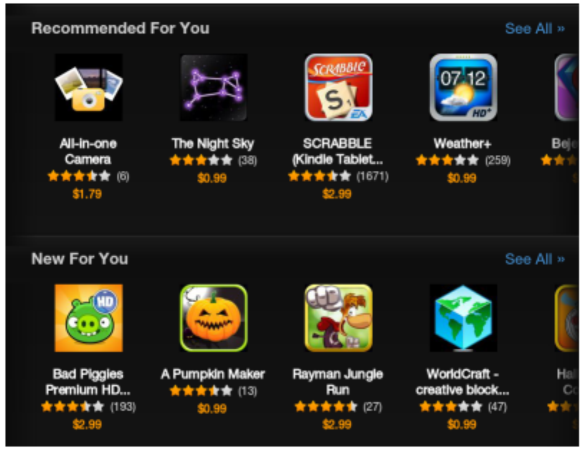
International Conference on Probabilistic Graphical Models, Lugano, Switzerland

Priyank Jaini, Abdullah Rashwan, Han Zhao, Yue Liu, Ershad Banijamali, Zhitang Chen, Pascal Poupart

Streaming Numerical Data

Traffic classification

App recommendation



- **Challenge:** update model after each data vector
- **Solution:** online learning for continuous SPNs

Outline

- Background: Sum product networks
- Gaussian Sum-Product Networks
  - Hierarchical mixture of Gaussians
  - Online algorithm: Bayesian Moment Matching
- Experiments: comparison with
  - Other online algorithms: oEM, SGD
  - Other generative models: stacked RBMs, GenMMNs
- Conclusion and future work

What is a Sum-Product Network?

- Prosposed by Poon and Domingos (UAI 2011) (equivalent to arithmetic circuit, Darwiche 2003)

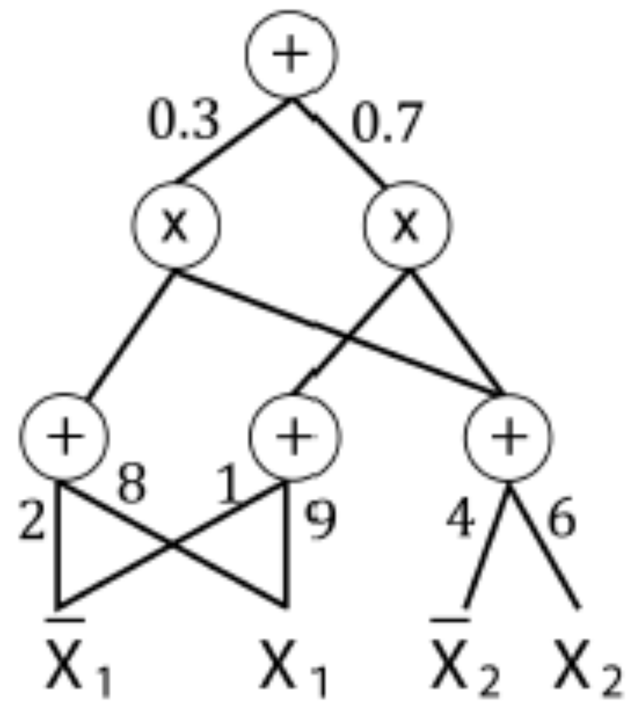
- Two views:

Deep architecture with clear semantics	Tractable probabilistic graphical model

Deep Architecture

- Specific type of deep neural network
  - Sum node: $\log(\sum_{i} w_{i}\, in$
  - Product node: $\exp(\sum_{i} inp$

- Advantages:
  - Clear semantics
  - Generative models
  - Flexible queries



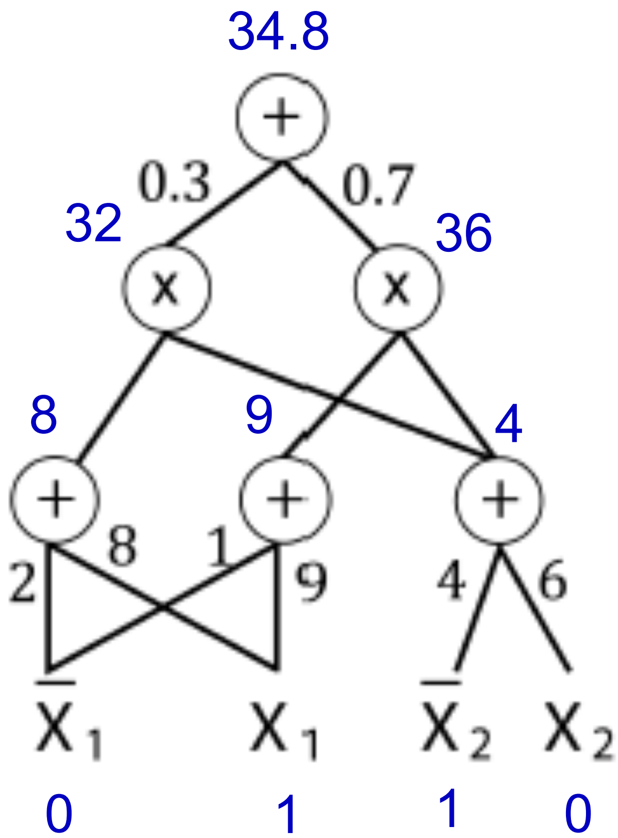
Probabilistic Inference

- SPN represents a joint distribution over a set of random variables

- Example:

$\Pr(X{\downarrow}1 = true, X{\downarrow}2 = fals$

$= 34.8/$

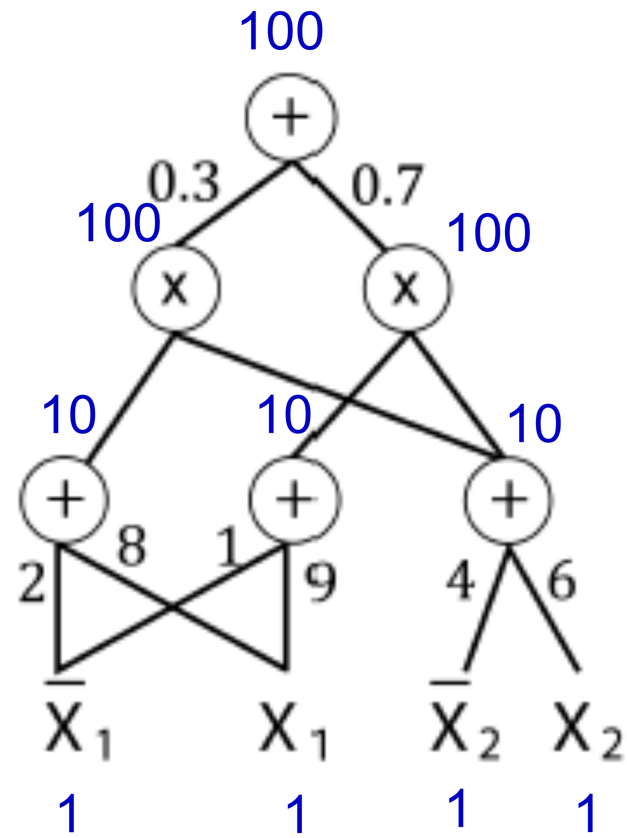


Probabilistic Inference

- SPN represents a joint distribution over a set of random variables

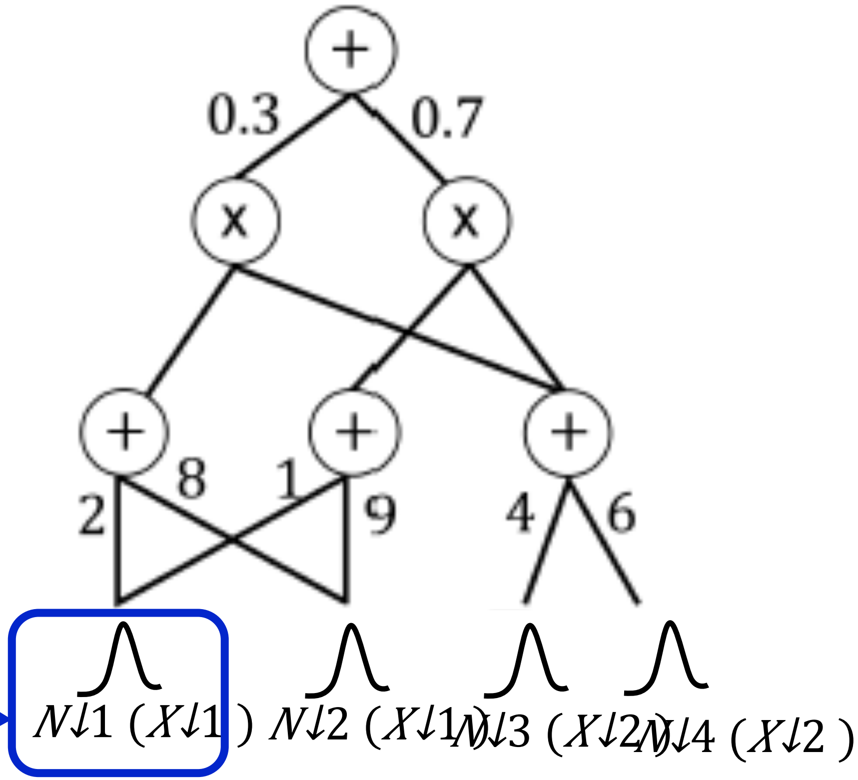
- Example:
$\Pr(X{\downarrow}1 = true, X{\downarrow}2 = fals$
$= 34.8/$
100

- **Linear complexity!**



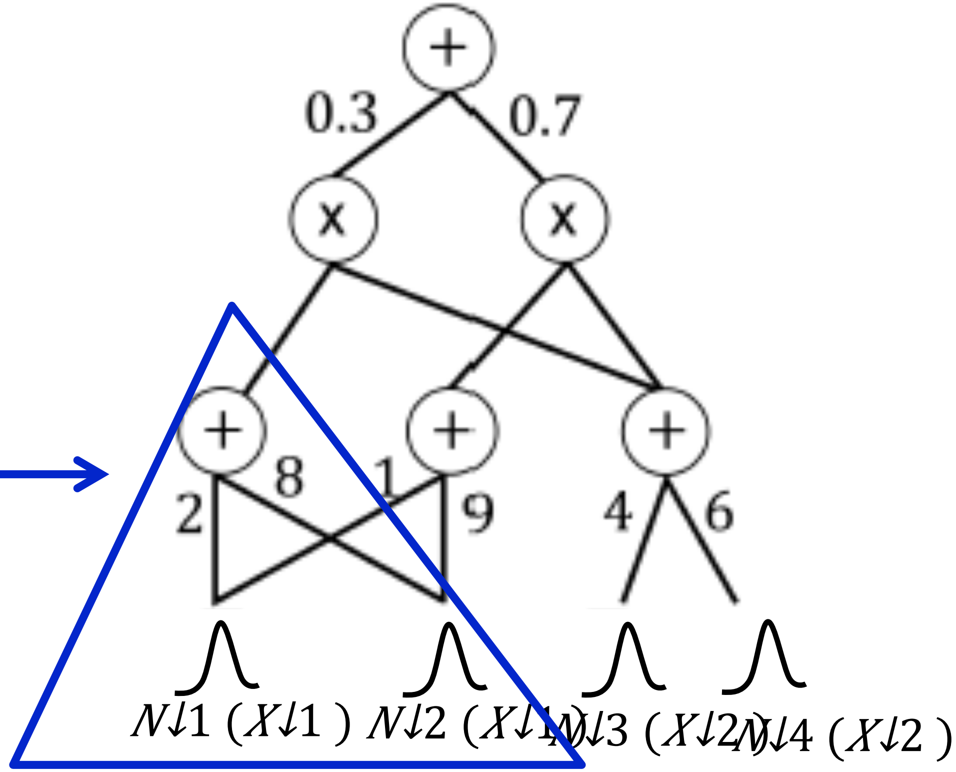
Continuous SPNs

Gaussian



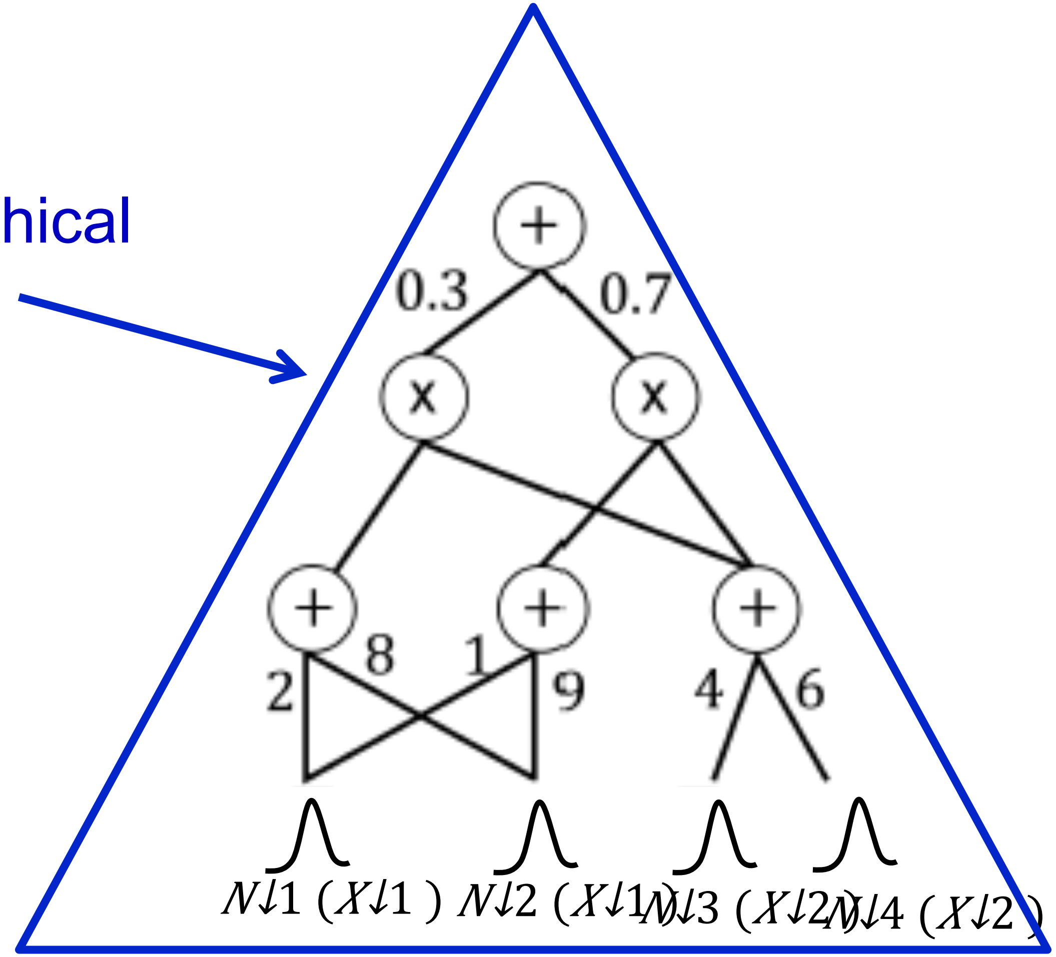
Continuous SPNs

(Unnormalized)
mixture of Gaussians



Continuous SPNs

(Unnormalized) hierarchical mixture of Gaussians



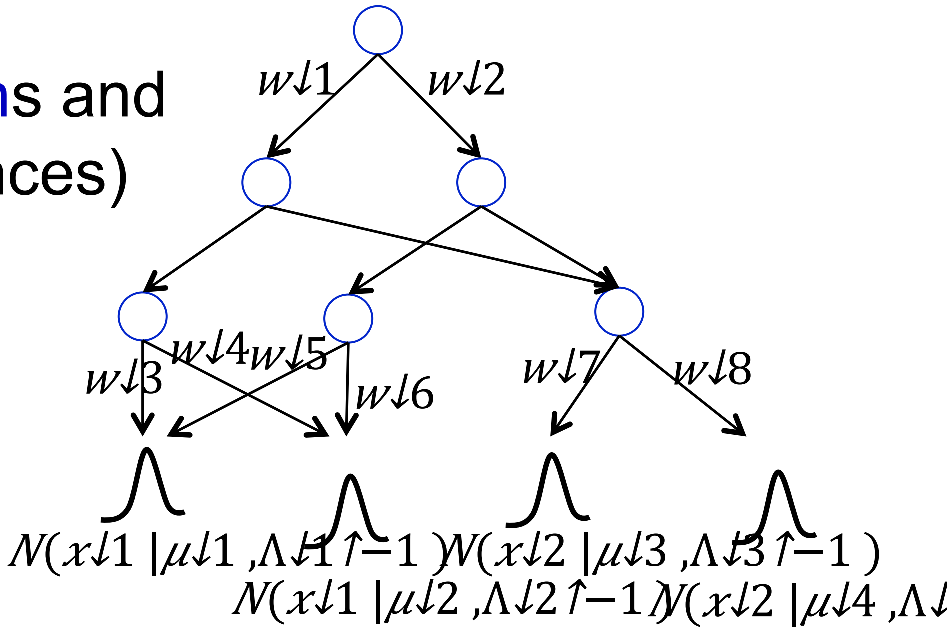
Learning SPNs

- Structure Learning by
  - **Clustering** (Dennis and Ventura, 2012; Gens and Domingos, 2013; Peharz et al., 2013; Rooshenas and Lowd, 2014; Adel et al., 2015; Vergari et al., 2015)
- Parameter Learning by
  - **Maximum likelihood**: stochastic gradient descent (SGD) (Poon & Domingos, 2011), expectation maximization (EM) (Perharz, 2015), signomial programming (Zhao & Poupart, 2016)
  - **Bayesian learning**: Bayesian Moment Matching (BMM) (Rashwan et al., 2015), Collapsed Variational Inference (Zhao et al., 2016)

Streaming Data

- Online parameter learning for discrete SPNs
  - Rashwan, Zhao and Poupart, AISTATS-2016:
    - **Stochastic Gradient Descent:** slow convergence, inaccurate (parameters: step size, mini-batch size, learning rate)
    - **Online Expectation Maximization (oEM):** inaccurate (parameters: mini-batch size, learning rate)
    - **Online Bayesian Moment Matching (oBMM): accurate**
  - Zhao, Adel, Gordon and Amos, ICML-2016:
    - **Collapsed Variational Inference: accurate**
- Online parameter learning for continuous SPNs
  - Contribution: **extend oBMM to Gaussian SPNs**

Bayesian Learning



- **Parameters**: weights, means and precisions (inverse covariances)
- WLOG consider normalized SPNs (normalized weights)
- **Prior**: $P(\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$
  product of Dirichlets and Normal-Wisharts
- **Likelihood:** $P \boldsymbol{x} \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Lambda} = SPN(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$

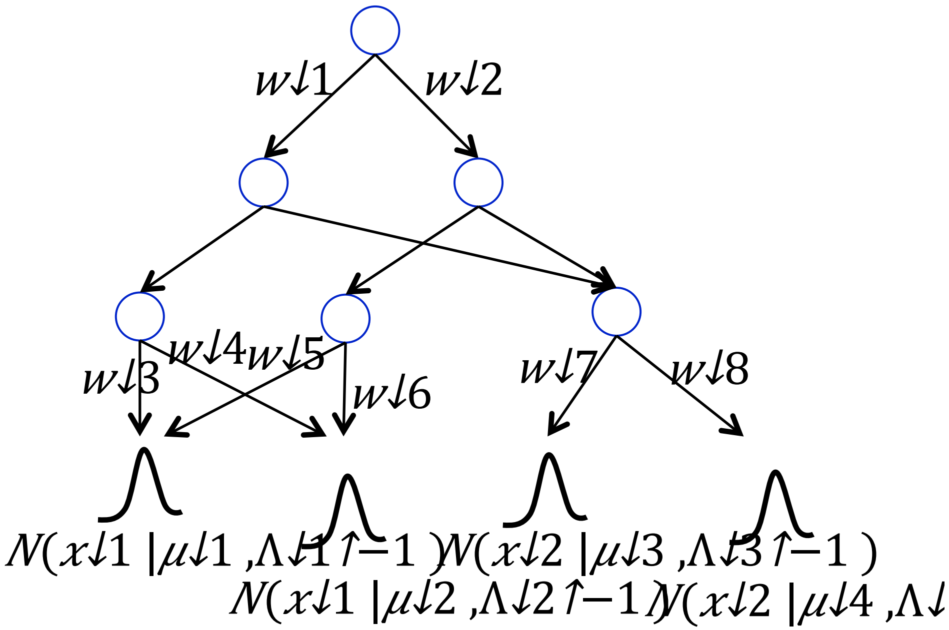
Bayesian Learning

- **Posterior**: $P\,\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{\Lambda}\,data$

$\propto P(\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{\Lambda})\; SPN\,\boldsymbol{x}^{(1)}\;\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{\Lambda}\; SPN\,\boldsymbol{x}^{(2)}\;\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{\Lambda}$
$\ldots SPN\,\boldsymbol{x}^{(n)}\;\boldsymbol{w},\boldsymbol{\mu},\boldsymbol{\Lambda}$

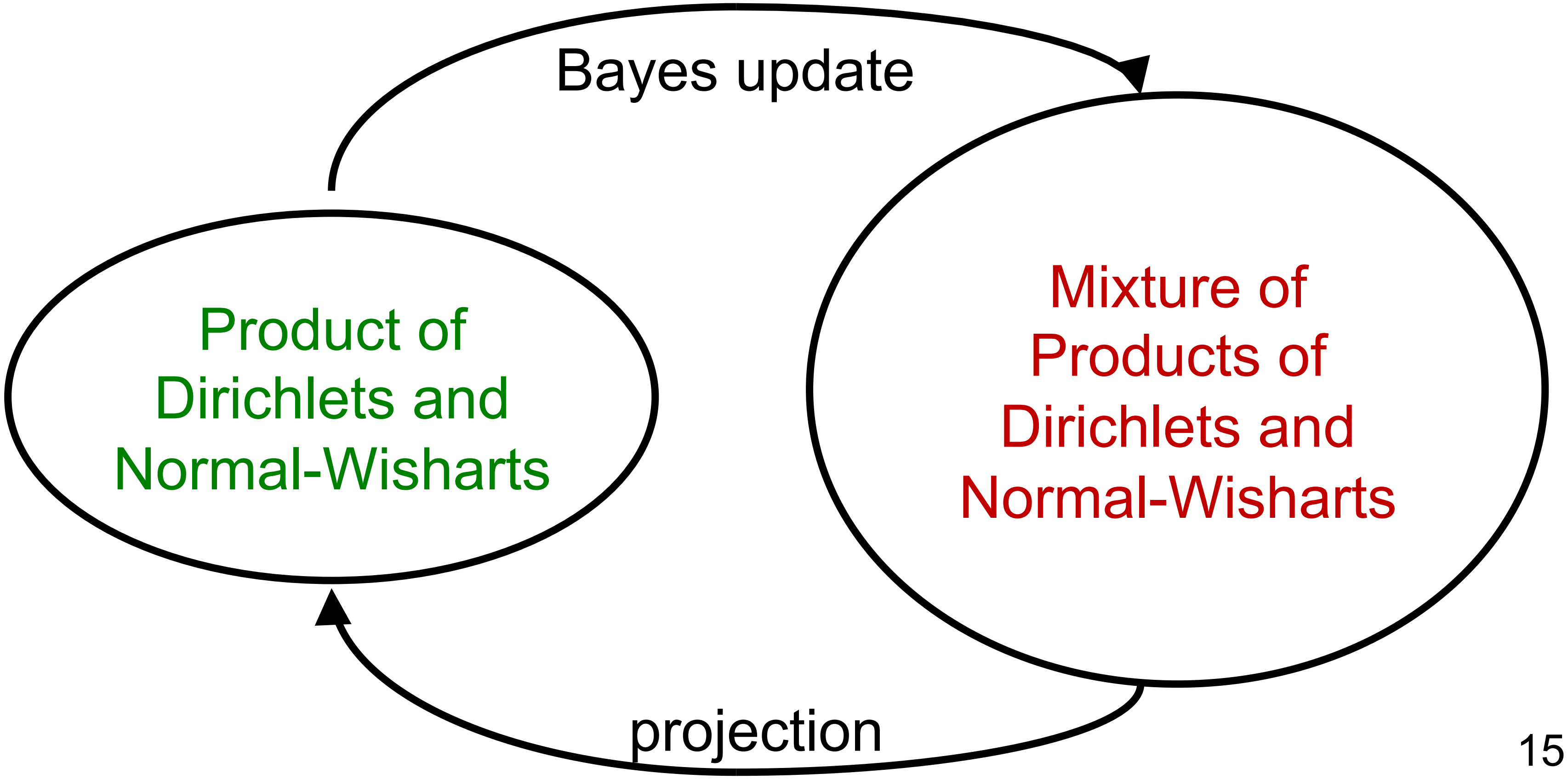
- **Naturally online learning!**

- **Problem: intractable** exponentially large mixture of products of Dirichlets and Normal-Wisharts



Moment Matching

- Solution: project posterior in a tractable family (assumed density filtering)



Sufficient Moments

- Dirichlet: $Dir(w_1,\dots,w_M;\alpha_1,\dots,\alpha_M)$

  $E[w_i]=\alpha_i/\sum_j \alpha_j \qquad E[w_i^2]=\alpha_i(\alpha_i+1)/(\sum_j \alpha_j)(1+\sum_j \alpha_j)$

- Normal-Wishart: $NW(\boldsymbol{\mu},\boldsymbol{\Lambda};\boldsymbol{\mu}_0,\kappa,\boldsymbol{W},\nu)$

  $E[\boldsymbol{\mu}]=\boldsymbol{\mu}_0$

  $E[(\boldsymbol{\mu}-\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)^T]=\kappa+1/\kappa(\nu-d-1)\,\boldsymbol{W}^{-1}$

  $E[\boldsymbol{\Lambda}]=\nu\boldsymbol{W}$

  $Var[\Lambda_{ij}]=\nu(W_{ij}^2+W_{ii}W_{jj})$

Posterior Moments

- Moment of some parameter s is

$$E[s]=\int_s s P(s|x)\, ds$$

- At leaf node i, for each $s \in \{\boldsymbol{\mu}_i, \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T, \boldsymbol{\Lambda}_i, \Lambda_{ijk}^2\}$:

$$E[s]=\int_s s\, NW(\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i | \boldsymbol{\alpha}_i, \kappa_i, \boldsymbol{W}_i, \nu_i)\left(c_i^0 + c_i^1 N(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i^{-1})\right)ds$$

- At sum node i:

$$E[w_{ij}^k]=\int_{\boldsymbol{w}_{i\cdot}} w_{ij}^k\, Dir(\boldsymbol{w}_{i\cdot} | \boldsymbol{\alpha}_{i\cdot})\left(c_i^0 + c_i^1 \sum_{j'} w_{ij'} V_{j'}(\boldsymbol{x})\right) d\boldsymbol{w}_{i\cdot}$$

Overall Algorithm

- Recursive computation of all c_i^0 and c_i^1
  - Two passes (details in paper)
  - Linear complexity in size of SPN
- Moment matching
  - System of linear equations
  - Linear complexity in size of SPN
- Streaming data
  - Posterior update: constant time w.r.t. amount of data

Empirical Results

Log likelihood and standard error based on 10-fold cross validation

Dataset	Flow Size	Quake	Banknote	Abalone	Kinematics	CA	Sensorless Drive
# of vars	3	4	4	8	8	22	48
oBMM (random)	-	-	-	**-1.82** $\pm$ 0.19	**-11.19** $\pm$ 0.03	**-2.47** $\pm$ 0.56	**1.58** $\pm$ 1.28
oEM (random)	-	-	-	-11.36 $\pm$ 0.19	-11.35 $\pm$ 0.03	-31.34 $\pm$ 1.07	-3.40 $\pm$ 6.06
oBMM (GMM)	**4.80** $\pm$ 0.67	**-3.84** $\pm$ 0.16	**-4.81** $\pm$ 0.13	**-1.21** $\pm$ 0.36	**-11.24** $\pm$ 0.04	**-1.78** $\pm$ 0.59	-
oEM (GMM)	-0.49 $\pm$ 3.29	-5.50 $\pm$ 0.41	-4.81 $\pm$ 0.13	-3.53 $\pm$ 1.68	-11.35 $\pm$ 0.03	-21.39 $\pm$ 1.58	-

oBMM more accurate than oEM

Empirical Results

Log likelihood and standard error based on 10-fold cross validation

Dataset	Flow Size	Quake	Banknote	Abalone	Kinematics	CA	Sensorless Drive
# of vars	3	4	4	8	8	22	48
oBMM (random)	-	-	-	-1.82 ± 0.19	-11.19 ± 0.03	-2.47 ± 0.56	**1.58** ± 1.28
oEM (random)	-	-	-	-11.36 ± 0.19	-11.35 ± 0.03	-31.34 ± 1.07	-3.40 ± 6.06
oBMM (GMM)	**4.80** ± 0.67	-3.84 ± 0.16	-4.81 ± 0.13	**-1.21** ± 0.36	-11.24 ± 0.04	**-1.78** ± 0.59	-
oEM (GMM)	-0.49 ± 3.29	-5.50 ± 0.41	-4.81 ± 0.13	-3.53 ± 1.68	-11.35 ± 0.03	-21.39 ± 1.58	-
SBRM	-0.79 ± 0.01	**-2.38** ± 0.01	-2.76 ± 0.01	-2.28 ± 0.01	**-5.55** ± 0.02	-4.95 ± 0.01	-26.91 ± 0.03
GenMMN	0.40 ± 0.01	-3.83 ±	**-1.70** ± 0.03	-3.29 ± 0.10	-11.36 ± 0.02	-5.41 ± 0.14	-29.41 ±1.19

oBMM competitive with SBRM and GenMMN

Conclusion

- Contributions
  - Continuous SPNs with Gaussian leaves
  - Online Bayesian Moment Matching for streaming data

- Future work
  - More extensive experiments with larger problems
  - Online structure learning
  - Generalize leaf distributions to exponential family