Bayesian networks
- Time-series models
- Apache Spark & Scala

Dr John Sandiford, CTO Bayes Server
Contents

• Introduction
• Bayesian networks
• Latent variables
• Anomaly detection
• Bayesian networks & time series
• Distributed Bayesian networks
• Apache Spark & Scala
Introduction
Profile
linkedin.com/in/johnsandiford

• PhD Imperial College – Bayesian networks
• Machine learning – 15 years
  • Implementation
  • Application
  • Numerous techniques
• Algorithm programming even longer
  • Scala, C#, Java, C++
• Graduate scheme – mathematician (BAE Systems)
• Artificial Intelligence / ML research program 8 years (GE/USAF)
• BP trading & risk analytics – big data + machine learning
• Also: NYSE stock exchange, hedge fund, actuarial consultancy, national newspaper
Bayesian networks
Insight, prediction & diagnostics

**Insight**

- Automated
- Large patterns
- Anomalous patterns
- Multivariate

**Prediction / inference**

- Supervised or unsupervised
- Anomaly detection
- Time series

**Diagnostics / reasoning**

- Troubleshooting
- Value of information
- Decision support
Bayesian networks

• Efficient application of probability
• DAG
• Subset of wider probabilistic graphical models
• Not a black box
• Handle missing data
• Probabilistic predictions
• Both supervised & unsupervised techniques

• Superset of many well known models
  • Mixture model (cluster model)
  • Naïve Bayes
  • AR
  • Vector AR
  • Hidden Markov model
  • Kalman filter
  • Markov chains
  • Sequence clustering
Example – Asia network

\[ P(X, e) = \sum_{U \backslash X} P(U, e) = \sum_{U \backslash X} \prod_{i} P(U_i | pa(U_i))e \]

- \( U \) = universe of variables
- \( X \) = variables being predicted
- \( e \) = evidence on any variables
Example – Waste network
Example – the bat (40,000 links)
Example – static & temporal
Prediction & uncertainty

• Inputs to a prediction can be missing (null)
• Discrete predictions have an associated probability, e.g.
  • {0.2, 0.8}
• Continuous predictions have both a mean and variance, e.g.
  • mean =0.2, variance = 2.3
• We can calculate joint probabilities over discrete, continuous or hybrid
• We can calculate the likelihood / log-likelihood
Prediction (inference)

• Basically just probability, but with complex algorithms to perform the calculations efficiently
  • Marginalization
    • Sum (discrete), integrate (continuous)
    • Summing in margins
  • Multiplication
• Bayes Theorem
• Exact inference
  • Exact subject to numerical rounding
  • Usually explicitly or Implicitly operating on trees
• Approximate
  • Deterministic
  • Non-deterministic
Latent variables
Latent variables

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>7.9</td>
</tr>
<tr>
<td>6.9</td>
<td>1.98</td>
</tr>
<tr>
<td>0.1</td>
<td>2.1</td>
</tr>
<tr>
<td>1.1</td>
<td>?</td>
</tr>
<tr>
<td>9.1</td>
<td>7.2</td>
</tr>
<tr>
<td>?</td>
<td>9.2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Latent variables
Parameter learning
EM algorithm & extensions for missing data

D3 animated visualization available on our website
Latent variables

- This is exactly the same as a mixture model (cluster model)
- This model only has X & Y, but most models have much higher dimensionality
- We can extend other models in the same way, e.g.
  - Mixture of Naïve Bayes (no longer Naïve)
  - Mixture of time series models
  - A structured approach to ensemble methods?
Latent variables

• Algorithmically capture underlying mechanisms that haven’t or can’t be observed
• Latent variables can be both discrete & continuous
• Can be hierarchical (similar to Deep Belief networks)
Anomaly detection
Univariate Gaussian pdf
Anomaly detection – log-likelihood

• This can also be calculated for
  • Discrete, continuous & hybrid networks
  • Networks with latent variables
  • Time series networks

• Allows us to perform anomaly detection

• Under the hood, great care has to be taken to avoid underflow
  • Especially with temporal networks
Anomaly detection
Time series anomaly detection

D3 animated visualization available on our website
Bayesian networks
Time series (DBN)
Sample time series data

- Multiple time series instances
- Multivariate (X1, X2)
- Different lengths
Time series, unrolled

Distributions are shared
Time series, unrolled – lag 1
Time series, unrolled – lag 4
Dynamic Bayesian network (DBN)
Equivalent model

- Structural learning algorithms can often automatically determine the links
- Cross, auto & partial correlations
Time series

• We can mix static & temporal variables in the same Bayesian network
• We can include discrete and/or continuous temporal variables
Time series & latent variables

• We can include static or temporal latent variables
  • Discrete or continuous

• In the same way that we used 3 multivariate Gaussians earlier, we can model mixtures of multivariate time series
  • i.e. model different multivariate time series behaviour
  • E.g. 2 time series may be correlated in a certain range, and anti-correlated in another
Types of time series prediction (t=time)

• $P(X_1@t=4)$
  • Returns probabilities for discrete, mean & variance for continuous

• $P(X_1@t=4, X_2@t=4)$
  • Joint time series prediction (funnel)

• $P(X_1@t=2, X_1@t=3)$
  • Across different times

• $P(A, X_1@t=2)$
  • Mixed static & temporal

• Log-likelihood of a multivariate time series
  • Anomaly detection
Distributed Bayesian networks
Different types of scalability

- **Data size**: Big data?
- **Connectivity**: (discrete -> exponential)
- **Network size**: Rephil > 1M nodes
- **Inference**: (distributed)
Data

• Algorithm is agnostic to the distributed platform
• We will look at how it can be used with Apache Spark
• .NET, Java and therefore derivatives such as Scala
Apache Spark

- Spark SQL
- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)

RDD Objects

```
rdd1.join(rdd2)
  .groupBy(...)
  .filter(...)
```
Apache Spark

• RDD (Resilient distributed dataset)
• In memory
• DAG execution engine
• Serialization of variables
Apache Spark

• Cache & iterate
  • Great for machine learning algorithms, including Bayesian networks
• Scala, Java, Python
Bayes Server Distributed architecture

• On each thread on each worker node, Bayes Server simply calculates the sufficient statistics
  • This often requires an inference algorithm per thread/partition
  • This plays nicely with Bayes Server streaming, without any hacking
• Could be on Hadoop + Spark + YARN, Cassandra, a desktop, or next gen platforms
Spark integration

• Moving from Hadoop mapReduce to Spark
  • Proof of concept took a single afternoon
  • Due to agnostic approach & streaming
  • RDD.mapPartitions

• Spark serialization
  • Use of companion object methods (standard approach)
Example – distributed learning

```scala
/**
 * Some test data. Normally you would load the data from the cluster.
 * We have hard coded it here to keep the example self contained.
 * @return An RDD
 */

def createRDD = {
    sc.parallelize(Seq(
        Seq((1.0, 2.3), (2.3, 4.5), (6.2, 7.2), (4.2, 6.6)),
        Seq((3.3, -1.2), (3.2, 4.4), (-3.3, -2.3), (4.15, 1.2), (8.8, 2.2), (4.1, 9.9)),
        Seq((1.0, 2.0), (3.3, 4.1)),
        Seq((5.0, 21.3), (4.3, 6.6), (-2.1, 4.5)),
        Seq((4.35, -3.25), (13.44, 12.4), (-1.3, 3.33), (4.2, 2.15), (12.8, 4.25)),
        Seq((1.46, 2.22), (1.37, 3.15), (2.2, 2.25))
    ))
}
```
Example – distributed learning

```scala
val sc = new SparkContext(conf)

// hard code some test data. Normally you would read data from your cluster.
val data = CreateRDD.cache()

// A network could be loaded from a file or stream
// we create it manually here to keep the example self contained
val network = createNetwork

val parameterLearningOptions = new ParameterLearningOptions

// Bayes Server supports multi-threaded learning
// which we want to turn off as Spark takes care of this
parameterLearningOptions.setMaximumConcurrency(1)

// parameterLearningOptions.setMaximumIterations(...) // this can be useful to limit the number of iterations

val config = new MemoryNameValues // we could also use broadcast variables

val output = ParameterLearning.learnDistributed(network, parameterLearningOptions,
      new BayesSparkDistributor(Seq[((Double, Double)])
          (data,
           config,
           (ctx, iterator) => new TimeSeriesEvidenceReader(ctx.getNetwork, iterator)
      ))
```
Distributed time series prediction

```scala
// make some time series predictions into the future

val predictions = Prediction.predict[TimeSeries](
    network,
    testData,
    Seq(
        PredictVariable("X1", Some(PredictTime(5, Absolute))),
        PredictVariance("X1", Some(PredictTime(5, Absolute))),
        PredictVariable("X2", Some(PredictTime(5, Absolute))),
        PredictVariance("X2", Some(PredictTime(5, Absolute))),
        PredictVariable("X1", Some(PredictTime(6, Absolute))),
        PredictVariance("X1", Some(PredictTime(6, Absolute))),
        PredictVariable("X2", Some(PredictTime(6, Absolute))),
        PredictVariance("X2", Some(PredictTime(6, Absolute))),
        PredictLogLikelihood() // this value can be used for Time Series anomaly detection
    ),
    (network, iterator) => new TimeSeriesReader(network, iterator)
)
predictions.foreach(println)
```
Scala

• JVM
• Functional & OO
• Statically typed
• Apache Spark is written in Scala
Spark streaming

- Great for real time anomaly detection
GraphX

- Machine learning on table data, queried from Graph
Thank you

• www.bayesserver.com - download, documentation
• www.bayesserver.com/Visualization.aspx
• www.bayesserver.com/bayesspark.aspx
  • Apache Spark source code & examples

• Professional services
  • Training
  • Consultancy
  • Proof of concepts