第七章 网络安全态势评估

网络安全态势

- **网络态势感知概念**
- **态势**

- Tim Bass在1999年首次提出了**网络态势感知概念**，它是利用态势感知技术全局的分析网络环境、快速判断当前形式、对未来进行预测、做出及时响应等一系列的过程。
- **态势**是指围绕终端设备、通信环境、用户行为等众多因素所构成的整个网络的状态及其变化趋势，具有全局性、多变性、复杂性、不确定性、扩散性等特点。

I. 来源
   - 网络中的网络管理设备
   - 网络安全设备
   - 网络监管设备

II. 任务
   - 数据融合和再加工
   - 通过如趋势图、饼状图、柱状图、表格等多种表达方式展现
   - 对恶意的网络行为进行识别
   - 对网络威胁进行判断和预警
   - 执行相应的防御策略

研究现状

- **DARPA**
- **CERT**
- **Lincoln 实验室等**

- 美国国防部高级计划署 (DARPA)及其资助的兰德公司、卡内基梅隆大学的 CERT(Computer Emergency Response Team)、普渡大学、麻省理工学院的 Lincoln 实验室等都在具体应用方面开展了研究。通过建立和部署网络安全态势系统，能够实时动态的掌握网络的全局运行状况，对已经或即将出现的安全问题进行及时响应和预测。
- 在本章介绍几种将*支持向量机方法*、*贝叶斯网络方法*、*隐马尔科夫方法*等引入到网络安全态势评估中的过程。这些对知识表示和进行概率推理的优秀算法和框架应用于网络安全态势感知这一领域，具有广阔的发展前景。

目录

支持向量机方法

- **支持向量机**（Support Vector Machine，SVM）是一种监督式学习的方法，由Vapnik和Cortes在1995年提出。
- SVM通过*升维*和*线性化*的方式，非常巧妙的把原样本空间中非线性问题转化为高维空间的线性可分问题，可以很好的进行模式识别、分类、回归分析等。

支持向量机方法

1. **支持向量机原理**
2. **评价指标体系的建立及实现**

- **支持向量机**（Support Vector Machine，SVM）是一种监督式学习的方法，由Vapnik和Cortes在1995年提出。
- SVM通过*升维*和*线性化*的方式，非常巧妙的把原样本空间中非线性问题转化为高维空间的线性可分问题，可以很好的进行模式识别、分类、回归分析等。

支持向量机原理

- **间隔最大化**
- **核方法-非线性**
- **凸优化**

- SVM是最优秀的数据挖掘算法之一。**优点**
  - 可依靠小样本学习
  - 泛化能力强
  - 易训练
  - 局部最优即全局最优（凸优化）
- SVM依靠多种**核函数**避免显示非线性映射，将低维空间中的输入向量映射到高维空间，并在高维空间建立一个超平面。

SVM分类

- 问题描述
- 公式建模
- 求解目标

- 给定训练集$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$，其中$x_i \in X=k^n$是输入的指标向量，其分量称为指标；$y_n \in y=\{-1,+1\}$是输出，i=1,2,...,n。我们将这 个样本所形成的集合称作训练集。对任意给定的一个新的模式x，能够最终推断它所对应的输出结果y为1或-1。这个问题转化为寻找一个把R^n点分成两部分的规则。

SVM分类

- 问题描述
- 公式建模

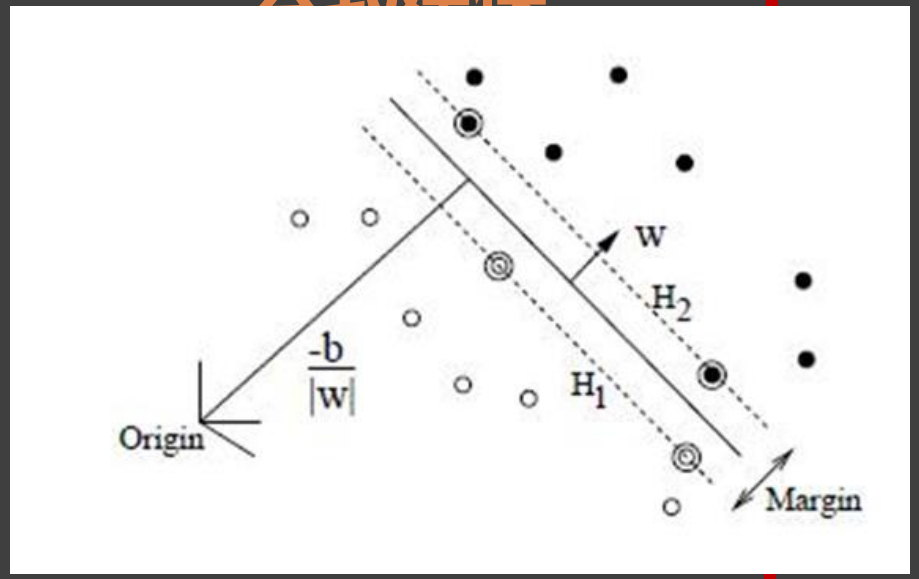


图7.1 支持向量机

如图7.1所示，对一堆训练数据的正负样本$\{x_i, y_i\}, i = 1, \cdots, l,$，其中$y_i \in \{-1,1\}, x_i \in R^d$。假设超平面$H : w \cdot x + b = 0$能将样本准确地分开，且存在两个平行于*H*的超平面：$H_1 : w \cdot x + b = 1$和$H_2 : w \cdot x + b = -1$，使得距离超平面*H*最临近的样本在H_1和H_2上。这些样本即为支持向量。而其它所有的训练样本全位H_1和H_2之外，即

$$w \cdot x_i + b \geq 1 \quad for \quad y_i = 1$$

$$w \cdot x_i + b \leq -1 \quad for \quad y_i = -1$$

合并二式，

$$y_i(w \cdot x_i + b) - 1 \geq 0$$

SVM分类

- 问题描述
- 公式建模
- 求解目标

支持向量机的目标就是要找到能够把样本准确无误地分割成两部分的超平面H，且使得H_1和H_2的**距离最大化**。寻找H的关键是寻找支持向量（落在H_1和H_2上的样本点）。

若点坐标(x_0, y_0, z_0),平面为$Ax_0 + By_0 + Cz_0 + D = 0$，则点到平面的距离为

$$d = \left| \frac{Ax_0 + By_0 + Cz_0 + D}{\sqrt{A^2 + B^2 + C^2}} \right|$$

所以由超平面H_1和H_2的距离为：

$$Margin = \frac{2}{\|w\|}$$

因此，只要最小化$\|w\|^2$，就能找到分类超平面。于是构建目标函数

$$\begin{cases} \min \dfrac{\|w\|^2}{2} \\ s.t. \quad y_i(w \cdot x_i + b) - 1 \geq 0 \end{cases}$$

拉格朗日乘子

- 为去除约束方程，引入拉格朗日乘子。原目标函数变为

$$L(w,b,a_i)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{l}a_i(y_i(w\cdot x_i+b)-1)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{l}a_i y_i(w\cdot x_i+b)+\sum_{i=1}^{l}a_i$$

- 其中，$a_i\geq 0$目标函数式典型的二次规划问题，$\min_{w,b}\max_{a_i\geq 0}L(w,b,a_i)$该问题叫做原问题，通过对偶变换，

$$\min_{w,b}\max_{a_i\geq 0}L(w,b,a_i)=\max_{a_i\geq 0}\min_{w,b}L(w,b,a_i)$$

得到对偶问题 $\max_{a_i\geq 0}\min_{w,b}L(w,b,a_i)$。

其意义是：原凸规划问题转换成对*w*和*b*求偏导，令其等于0消掉*w*和*b* ，再对*a*求 *maxL*。

对*w*和*b*求偏导数

对*w*和*b*求偏导，有

$$\frac{\partial L(w,b,a_i)}{\partial w} = w - \sum_{i=1}^{l} a_i y_i x_i$$
$$\frac{\partial L(w,b,a_i)}{\partial b} = -\sum_{i=1}^{l} a_i y_i$$

将偏导数置0，有

$$w = \sum_{i=1}^{l} a_i y_i x_i$$
$$\sum_{i=1}^{l} a_i y_i = 0$$

将结果代回原目标方程

$$\begin{aligned}
\min_{w,b} L(w,b,a_i) &= \frac{1}{2}\|w\|^2 - w\sum_{i=1}^{l} a_i y_i x_i - b\sum_{i=1}^{l} a_i y_i + \sum_{i=1}^{l} a_i \\
&= \frac{1}{2}\|w\|^2 - w\cdot w - b\cdot 0 + \sum_{i=1}^{l} a_i \\
&= -\frac{1}{2}\|w\|^2 + \sum_{i=1}^{l} a_i \\
&= \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j (x_i \cdot x_j)
\end{aligned}$$

对偶问题

- 将求解结果代回对偶问题，得到新的对偶问题表达形式

$$\begin{cases} \max\limits_{a_i}\left\{\sum\limits_{i=1}^{l} a_i - \frac{1}{2}\sum\limits_{i=1}^{l}\sum\limits_{j=1}^{l} a_i a_j y_i y_j (x_i \cdot x_j)\right\} \\ s.t. \quad \sum\limits_{i=1}^{l} a_i y_i = 0 \\ \quad\quad a_i \geq 0 \end{cases}$$

- 这样x_i , y_i均为训练样本给定的已知量，因此，最终寻找间隔最大化问题被转变为求解a_i（拉格朗日乘子）使得目标函数最大化的过程。
- *二次凸优化的性质保证这样的最优解是存在的。*

0约束

- 依据KKT（Karush-Kuhn-Kucter）条件， 只有当

$$a_i(y_i(w\cdot x_i+b)-1)=0$$

时，取得最优解。对于非支持向量而言，一定可以确定

$$y_i(w\cdot x_i+b)-1$$

不为0。因此，对于非支持向量样本，其对应的拉格朗日乘子必为零，才能满足等式约束。

总而言之，除支持向量对应的拉格朗日乘子不为0外，其他均为0。

求得最优*w*和*b*

- 通过求解拉格朗日乘子a_i的最优解，代入

$$w=\sum_{i=1}^{l} a_i y_i x_i$$

和

$$a_i(y_i(w\cdot x_i+b)-1)=0$$

即可求得最优*w*和*b*。

从而，找到了最优超平面，建立了SVM分类模型（参数为*w*和*b* ）。

评价指标体系的建立及实现

- **分类判别函数**
- **SVM多分类问题**
- **基于SVM的网络态势分类模型**

- 样本类别预测
  - 对于新来的样本（x,y），通过判别函数去判定新样本类别

$$f(x) = sign(\sum_i a_i y_i < x_i, x > +b)$$

  -
  - $< x_i, x >$ 表示新样本与支持向量做内积运算，通过判别函数的符号进行分类，$f<0$， 样本被划分为+1类；否则被分为-1类。

SVM多分类问题

- SVM通过寻找最优超平面处理二分类问题。
- 面对多分类问题就需要构造多SVM分类器。即在类别之间建立两两二分类器。
- 在M个类中，建立起任意两个类之间的分类器，共需要 $\frac{M(M-1)}{2}$ 个。

基于SVM的网络态势分类模型

- **网络安全评估指标**
- **警示等级**

- **建立12维输入向量**
  - CPU占用率、内存占用率、端口流量、丢包率、网络可用带宽、平均往返时延、传输率、吞吐率、服务请求率、服务响应率、出错率以及响应时间
- **建立五种输出类别**
  - 将网络的安全态势划分成五个警示等级，依次由高到底的标示出整个网络安全状态。
- **建立输入向量到输出类别的多SVM分类器**
  - 以完成网络台式等级预测

基于SVM的网络态势分类模型

- **网络安全评估指标**
- **警示等级**

- **构造5个二分类器**SVM1、SVM2、SVM3、SVM4、SVM5，决策函数为

$$f_m(x) = sign(\sum_i a_i^* y_i K(x_i, x) + b^*) \qquad m = 1,2,3,4,5$$

- 对于待测样本x，将其依次输入SVM1、SVM2、SVM3，SVM4、SVM5，依次得到f_i（其中i=1,2,…5），max(f_i）为最终的评价结果。

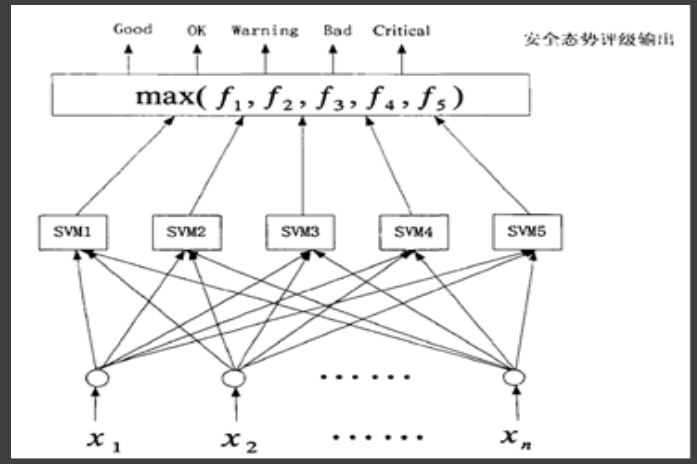


图7.2 基于SVM的态势分类模型

贝叶斯网络概述

1. **贝叶斯网络基础知识**
2. **表示与构成**
3. **特点**
4. **贝叶斯网络构建**
5. **基于贝叶斯网络的网络态势评估模型**

- **贝叶斯网络**（Bayesian network）是一种概率图型模型。
- 作为一种强有力的不确定性推理方法，贝叶斯网络巧妙的利用了*先验信息*和*样本数据*，能够避免对数据的过拟合，最终将每个变量相互之间的因果关系用简明的*图模型*清晰的表达出来，并结合专家知识可以进行定性分析和定量分析，使得推理出来的的结果更具有可信性，也容易理解和接受。

贝叶斯网络基础知识

- **概率论**
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

条件概率公式

设基本事件 A, B ，有 $P(A) > 0$ ，则

$$P(B|A) = \frac{P(AB)}{P(A)}$$

为事件A发生的条件下事件B发生的条件概率。

贝叶斯网络基础知识

- **概率论**
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

先验概率

设基本事件 $B_1, B_2, \cdots B_n$ 为样本空间S中的事件，

$P(B_i)$ 为根据先验知识估计的概率，我们称 $P(B_i)$ 为先验概率。

贝叶斯理论重视对先验知识的收集和加工，形成先验分布，能够提高统计推断的准确性。

贝叶斯网络基础知识

- 概率论
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

先验概率

当历史资料不全或没有的时候，只能凭借主观经验获得先验概率。这一类叫做主观先验概率。
但需要注意的是，这些主观判断并不是随意的，而是需要对所观察的事件有较为透彻的了解和丰富经验的，或者是这一行的专家做出的判断，该主观判断要能够符合实际。

贝叶斯网络基础知识

- **概率论**
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

设基本事件 $B_1, B_2, \cdots B_n$ 为样本空间*S*中的事件，则事件A发生的情况下，B_i 发生的概率 $P(B_i|A)$ 称为后验概率。

它是在先验概率基础上经过修正后更符合实际的概率，即得到附加信息之后再更新的概率，反应了人们在抽样后对事件认识的调整。

贝叶斯网络基础知识

- **概率论**
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

设A、B为两个事件，且 $P(A)>0$ ，则它们的联合概率为

$$P(AB)=P(B|A)P(A)$$

联合概率是指两个任意事件的乘积的概率，或称之为交事件的概率，也是乘法公式。

贝叶斯网络基础知识

- **概率论**
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

全概率公式是概率论中的重要公式。若影响事件A的所有因素 $B_1, B_2, \cdots B_n$ 满足 $B_i \bullet B_j = \phi(i \neq j)$,并且 $P(B_i) > 0$ ， $i = 1,2,\cdots n$ ，则必有

$$P(A) = \sum_{i=1}^{n} P(B_i) P(A|B_i)$$

贝叶斯网络基础知识

- **概率论**
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

贝叶斯概率

根据先验知识及现有数据，来预测出某未知事件发生的可能性，或对某事件发生可能性的相信程度。

贝叶斯公式或称为后验概率公式。设先验概率为$P(B_i)$，调查所获得的新附加信息为$P(A|B_i)$，其中$i=1,2,\cdots n$则后验概率为

$$P(B_i|A)=\frac{P(A|B_i)P(B_i)}{\sum_{j=1}^{n}P(A|B_j)P(B_j)}$$

贝叶斯网络基础知识

- **概率论**
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

条件独立公式

对概率模式M, A、B和C是U的三个互不相交的变量子集，如果对 $\forall x \in A, \forall y \in B$和$\forall z \in C$，都有 $p(x|y,z) = p(x|z)$ ，其中 $p(y,z) > 0$，称给定C时，A和B条件独立，记为 $I(A,C,B)_M$ 。

条件独立性在某些文献中定义为

$$p(x,y|z) = p(x|z)p(y|z)$$

贝叶斯网络基础知识

- **概率论**
  - 条件概率
  - 先验概率
  - 后验概率
  - 联合概率
  - 全概率
  - 贝叶斯概率
  - 贝叶斯公式
  - 条件独立公式

对于概率模式M,随机变量之间的依赖关系如下

绝对依赖: $I(A,\phi,B)_M$ 不成立，而且对任意的C, $I(A,C,B)_M$ 也不成立；

条件依赖: $I(A,\phi,B)_M$ 成立，但存在C,使 $I(A,C,B)_M$ 不成立；

绝对独立: $I(A,\phi,B)_M$ 成立，而且对任意的C, $I(A,C,B)_M$ 都成立；

条件独立: $I(A,\phi,B)_M$ 不成立，但存在 ，使 $I(A,C,B)_M$ 成立。

贝叶斯网络基础知识

- **图论**
  - 有向图
  - Chains
  - 汇聚节点

有向图G：是由结点集V，边集E表示的二元组 $G = G(V, E)$ ，若 $(x, y) \in E$ 表示从结点x到结点y有一条有向边。称节点x为y的父节点，节点y为x的子节点。通过父节点和子节点的递归定义，可定义祖先和后继。

根节点：无任何父节点的节点。

贝叶斯网络基础知识

- **图论**
  - 有向图
  - Chains
  - 汇聚节点

Chains: 在连接两个结点的路径中，若不考虑路径中边的方向，称这种路径为adjacency、path或chains。这个定义对有向图、无向图和混合图都是适用的。

汇聚节点：对于邻接路径中的任何一个结点v，如果有

$$(x,v)\in E\text{并且}(y,v)\in E$$

则称v为汇聚节点或碰撞节点(collider)。

表示与构成

- **贝叶斯网络**
  - DAG
  - 局部概率分布
  - 链式规则

贝叶斯网络描述了随机变量 $X=\{X_1,...,X_n\}$ 所遵从的联合概率分布，表达了变量之间概率依赖关系，它可以表示为 $B=<G,\Theta>$，由结构图G和局部概率分布 Θ 构成：

1) G：是一个有向无环图DAG，每个节点对应随机变量 $X=X_1,...,X_n$，有向边表示变量间的直接依赖关系，体现了领域知识定性方面的特征。在有向无环图G中，给定 X_i 的父结点，每个 X_i 独立于它的非后继结点。

2) Θ：是与每个变量 X_i 关联的局部概率分布的集合，Θ 中的元素是给定每个变量 X_i 的父节点，该节点取不同值的条件概率表 $P(x_i|Val(Parent(X_i)))$，其中，$Parent(X_i)$ 表示图G中 X_i 的父节点集。体现了领域知识定量方面的特征。

表示与构成

- 贝叶斯网络
  - DAG
  - 局部概率分布
  - 链式规则

由链式规则

$$P(X_1,...,X_n)=\prod_{i=1}^{n}P(X_i \mid X_1,X_2,...,X_{i-1})$$

对于任一X_i,可找到与X_i都不独立的最小子集$Parent(X_i)\subseteq\{X_1,X_2,...,X_{i-1}\}$,有

$$P(X_i \mid X_1,...,X_{i-1})=P(X_i \mid Parent(X_i))$$

因此，当这些变量元组$<X_1,...,X_n>$被赋予具体值后，其联合概率分布可以由下面公式表示

$$P(x_1,...,x_n)=\prod_{i=1}^{n}P(x_i \mid Val(Parent(X_i))=\prod_{i=1}^{n}\theta_{x_i \mid Val(Parent(X_i))}$$

特点

- **贝叶斯网络**

1. 能够方便地可视化节点变量之间的因果关系，也能够更好的进行因果推理。
2. 具有强大的不确定性问题推理能力，可以高效地在不完整的，不确定，不精确的信息条件下进行学习和推理。
3. 能够有效地进行多源信息融合。通过汇聚多节点数据的方式将多种推理信息纳入网络结构中进行信息加工再造，高效地将相关信息按照关联关系进行融合。

贝叶斯网络建造

- 在建造贝叶斯网络模型的步骤中，首先需相关领域的专家的参与构建，模型还需要反复修改、不断完善，在构建好网络中的节点后还需要进行概率估计。贝叶斯网络的结构和参数确定方式为：

I. *变量不多，变量之间关系不复杂*的应用领域情况下，可由该领域的专家根据经验知识确定出贝叶斯网络的结构和参数，早期的贝叶斯网络构造大多采用这种方式；

II. *领域变量之间的依赖关系比较明显，但对领域变量之间依赖关系的依赖程度不是特别清楚*的情况下，可由领域专家根据经验知识确定贝叶斯网络的结构，而网络的参数通过机器学习算法从大量训练数据中学习得到。

III. *领域数据量大，变量复杂，变量之间的依赖关系不明显，领域知识难以完全掌握*的情况下，则结构和参数需要通过机器学习算法从大量训练数据中学习得到，这种方法是由数据驱动的。

基于贝叶斯网络的网络态势评估模型

- **建立模型**
- **风险评估指标提取**
- **网络安全风险评估模型推理**

- 在网络安全态势评估中，对于潜在威胁的严重程度及危害性难以把握，容易导致模糊的语言来进行评价。
- 由于网络安全态势评估过程中的不确定信息难于量化处理，可以引入动态贝叶斯网络算法。
- 主要研究目的是得到软硬件资产、所受到的威胁、系统脆弱性、用户行为等多种风险因素以及它们之间的关系。

建立模型

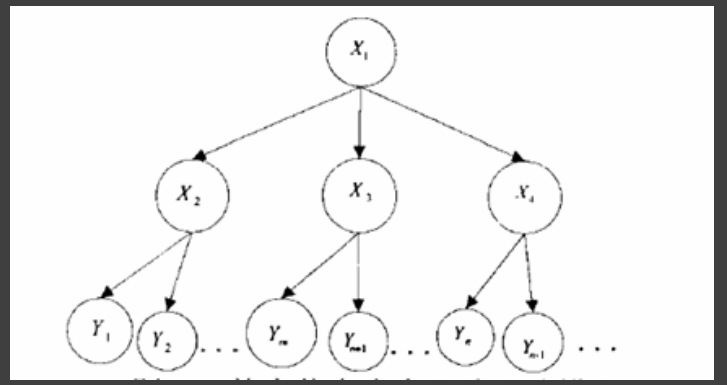
- **静态评估模型**
- **转移模型**
- **动态评估模型**

- 静态网络安全风险评估模型 B_0 由三元组 $<V,E,\theta>$ 构成，属于有向无环图，其中
- (1) $V=(X_i,Y_j)$ 为有限非空集合，V称为结点集(Nodal Set)。X_i 代表隐含结点， Y_i 代表观测结点。
- (2) θ是条件概率表，反映了各节点之间因果关系的强弱。
- (3) E是有限集合，称为边集(Frontier Set)。E中的每个元素都有V中的结点与之对应，称之为边(Edge)。
- (4) 有向边、端点：若图中的边e所对应的结点偶对是有序的，记为 $<a,b>$，则称e 是有向边，a、b分别称为有向边的始点和终点。称e是关联于结点a和结点b的，称结点a和结点b是邻接的。

建立模型

- 静态评估模型
- 转移模型
- 动态评估模型

(5)有向图 $D=(V,E)$，其中V中的元素称为顶点或结点，静态网络安全风险评估模型如下图。



静态网络安全风险评估模型

建立模型

- 静态评估模型
- 转移模型
- 动态评估模型

- 一个转移模型 $B_{->}$ 是一个静态网络安全风险评估模型的片段定义为 $B_{->}=(V,E,\beta)$ ， V中的结点包括 $X \cup X'$ ，X 表示当前时刻的状态，X' 代表下一时刻的状态，β代表根节点 X_1 相邻时刻的转移概率，X 中的结点没有父结点，X' 中的结点具有条件概率分布 $P(X' \mid parent(X'))$ ：

$$\beta = P(X' \mid X) = \prod_{i=1}^{n} P(X_i' \mid parent(X_i'))$$

-

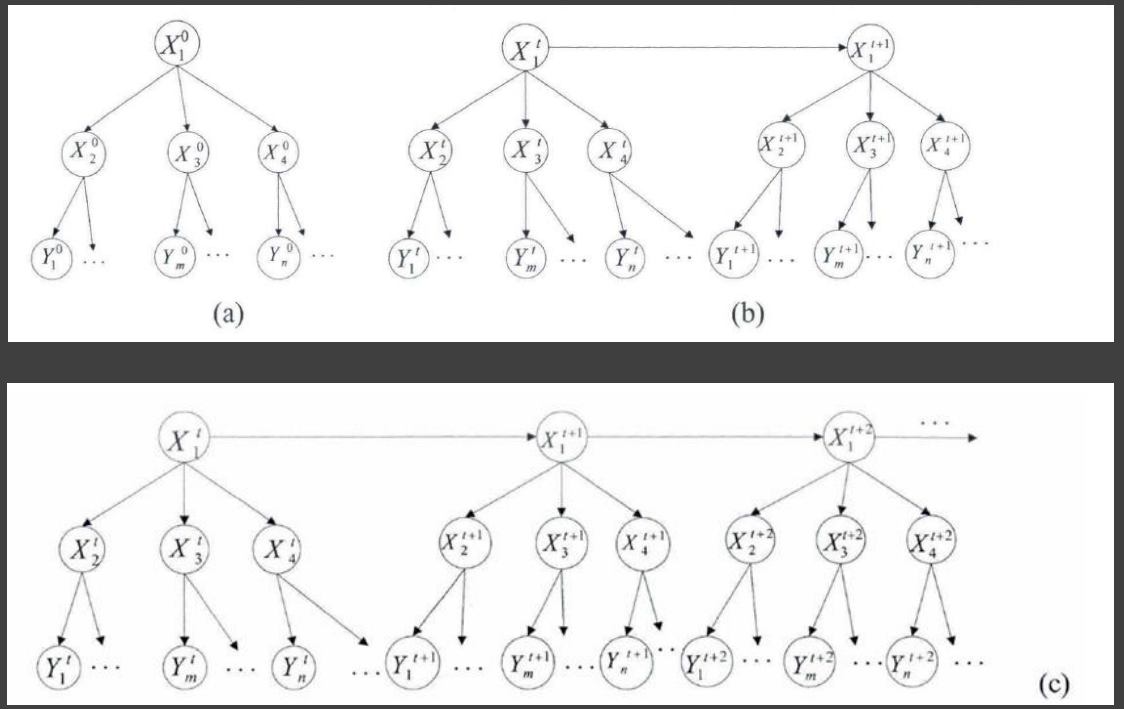
建立模型

- 静态评估模型
- 转移模型
- 动态评估模型

- 风险评估系统可以表示为一个三元组 $(B_0, B_{->}, T))$，B_0 为系统初始模型，$B_{->}$ 为系统的转移模型，T 为时间片长度。

建立模型

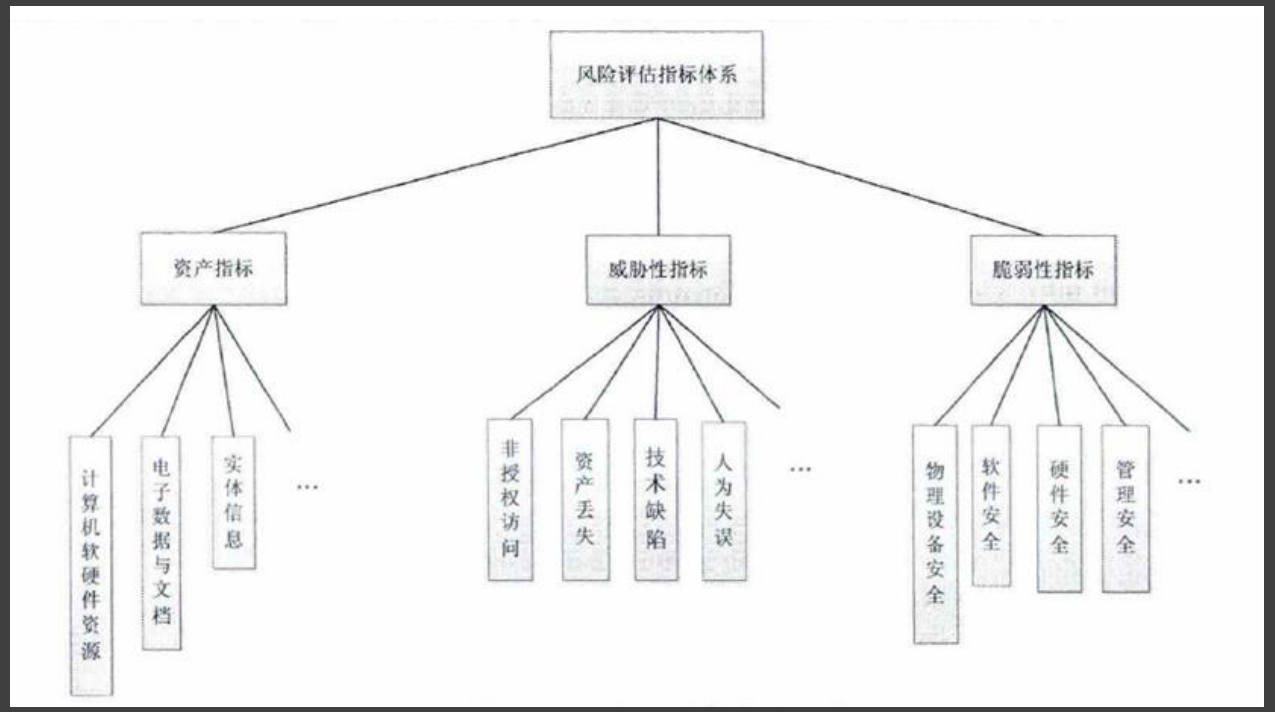
- 静态评估模型
- 转移模型
- 动态评估模型



动态网络安全风险评估模型

风险评估指标提取

- 网络态势评估指标由资产指标、威胁性指标、脆弱性指标的安全性指标等三项大指标，还可细分为各子类安全指标。下图构建了风险评估指标识别图。



网络安全风险评估模型推理

- **静态风险评估**
- **动态风险评估**

- 网络安全风险评估模型包含*观察节点*(其节点状态能够直接得到)和*隐含节点*(其状态必须经过推理得到)两种节点。
- *结构图*G表达了各个评估指标之间的因果关系；*局部概率*分布 Θ 表达了各个评估指标之间联系的强弱，用条件概率表示。已知 Y_t 可以得到 X_t ，推理的目标是通过观察量 $y_{1:t}$ 得到所需要的相应概率。在网络安全风险评估模型中，感兴趣的结点是 X_1, X_2, X_3, X_4 ，目标是通过*观测变量* $Y_1,...,Y_m,Y_n...$ 的状态，准确推理出*隐含结点*的概率。

网络安全风险评估模型推理

- **静态风险评估**
- **动态风险评估**

静态网络安全风险评估模型中具有4个隐藏结点（hidden），并假设含有r个观测结点（observed），推理的本质是计算

$$P(x_1,x_2,x_3,x_4 \mid y_1,y_2,...,y_r)=\frac{\prod_i p(x_i \mid parent(x_i))\prod_j p(y_j \mid parent(y_i))}{\sum_{x_1,x_2,x_3,x_4}\prod_i p(x_i \mid parent(x_i))\prod_j p(y_j \mid parent(y_i))}$$

上式可以得到风险概率公式

$$P(x_1)=\frac{P(x_1,x_2,x_3,x_4 \mid y_1,y_2,y_3,y_4)}{\prod_m p(x_m \mid parent(x_m))} \quad (7\text{-}21)$$

网络安全风险评估模型推理

- **静态风险评估**
- **动态风险评估**

- 用 X^t, Y^t 分别表示1：t的隐含序列和观测序列，用 x^t 表示t时刻隐含变量的值，y^t 表示t时刻观测变量的值，x_i^t 表示t时刻第i个隐含变量的状态值，y_j^t 表示t时刻第j个观测变量的状态值。
- 离散静态评估模型随着时间推移就得到*T个时间片组成的动态评估模型*，每一个时间片中都有4个隐藏结点和*r*个观测结点。

网络安全风险评估模型推理

- 静态风险评估
- 动态风险评估

对这个动态的风险评估模型进行推理，就是计算在所有的观测变量处于某一个观测状态的条件下，隐含结点的联合分布，计算如下

$$P(x_1^0, x_2^0, \ldots x_4^0, \ldots x_1^T, x_2^T, \ldots x_4^T \mid y_1^0, y_2^0, \ldots y_r^0, \ldots y_1^T, y_2^T, \ldots y_r^T)$$
$$= \frac{p(x_1^0, x_2^0, \ldots, x_4^0, \ldots x_1^T, x_2^T, \ldots x_4^T, y_1^0, y_2^0, \ldots, y_r^0, \ldots, y_1^T, y_2^T, \ldots y_r^T)}{\sum_{x_1^0, x_2^0, \ldots x_4^0, \ldots x_1^T, x_2^T, \ldots x_4^T} p(x_1^0, x_2^0, \ldots x_4^0, \ldots y_1^0, y_2^0, \ldots y_r^0, \ldots y_1^T, y_2^T, \ldots y_r^T)} \quad (7\text{-}22)$$

由于网络安全风险评估模型本身符合条件独立性假设，因此有

$$P(x_1^0, x_2^0, \ldots x_4^0, \ldots x_1^T, x_2^T, \ldots x_4^T, y_1^0, y_2^0, \ldots y_r^0, \ldots y_1^T, y_2^T, \ldots y_r^T)$$
$$= \prod_{i,j} p(y_j^i \mid parent(y_j^i)) \prod_{i,k} p(x_k^i \mid parent(x_k^i)) \quad (7\text{-}23)$$

网络安全风险评估模型推理

- 静态风险评估
- 动态风险评估

将(7-22)式代入(7-21)，得到公式(7-23)如下

$$P(x_1^0,x_2^0,...x_4^0,...x_1^T,x_2^T,...x_4^T \mid y_1^0,y_2^0,...y_r^0,...y_1^T,y_2^T,...y_r^T) = \frac{\prod_{i,j} p(y_j^i \mid parent(y_j^i)) \prod_{i,k} p(x_k^i \mid parent(x_k^i))}{\sum_{x_1^0,x_2^0,...x_4^0,...x_1^T,x_2^T,...x_4^T} \prod_{i,j} p(y_j^i \mid parent(y_j^i)) \prod_{i,k} p(x_k^i \mid parent(x_k^i))} \quad (7\text{-}24)$$

由(7-23)式可导出T个时间片内风险的联合概率分布为

$$P(x_1^0,x_1^1,...x_1^T \mid y_1^0,y_2^0,...y_r^0,...y_1^T,y_2^T,...y_r^T) = \frac{P(x_1^0,x_2^0,...x_4^0,...x_1^T,x_2^T,...x_4^T \mid y_1^0,y_2^0,...y_r^0,...y_1^T,y_2^T,...y_r^T)}{\prod_{i,m} p(x_m^i \mid parent(x_m^i))}$$

以初始时刻为例，计算初始时刻风险的概率如下式所示

$$P(x_1^0) = \frac{P(x_1^0,x_1^1,...x_1^T \mid y_1^0,y_2^0,...y_r^0,...y_1^T,y_2^T,...y_r^T)}{\prod_{m} p(x_1^n \mid parent(x_1^n))}$$

其中，$i \in [0,T], j \in [1,r], k \in [1,4], n \in [1,T], m \in [2,4]$

隐马尔可夫方法

1. HMM模型概述
2. 隐马尔可夫模型概念
3. HMM的基本算法
4. 建立网络态势评估模型

- **马尔科夫过程**(Markov process)是一种随机过程。1907年俄国数学家A.A.马尔可夫于提出了马尔科夫链概念，在此基础上发展成为了马尔科夫过程。
  - 马尔可夫过程的最大特点是：*在已知目前状态的条件下它未来的变化状态不依赖于它的过去状态*。
  - 举例
    - 如液体中微粒所作的布朗运动
    - 传染病受感染的人数
    - 车站的候车人数等

隐马尔可夫方法

1. **HMM模型概述**
2. **隐马尔可夫模型概念**
3. **HMM的基本算法**
4. **建立网络态势评估模型**

- **隐马尔科夫过程**(Hidden Markov Model，HMM) 最初由 L. E. Baum 等人用来描述*含有隐含未知参数*的马尔可夫过程，从可观察的参数中确定与之关联的隐含参数（hidden parameter）。
  - 应用领域
    - 自然语言处理
    - 计算机视觉
    - 故障分析
    - 生物信息处理
    - 语音识别

HMM模型概述

- 设系统有S_1， S_2 ，…，S_n 状态，系统可以从某一状态转换为另一状态。设q_t为系统在t时刻的状态，t时刻处于状态S_t的概率跟系统在时刻1,2,…,t-1的状态有关，概率为

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, ...)$$

如果系统在t时刻的状态只与t-1时间的状态相关，则为离散的马尔可夫过程

$$P(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, ...) = P(q_t = S_j \mid q_{t-1} = S_i)$$

若只考虑独立于时间t的随机过程

$$P(q_t = S_j \mid q_{t-1} = S_i) = a_{i,j}, 1 \le i, j \le N$$

其中$a_{i,j}$ >=0，$\sum_{j=1}^{N} a_{i,j} = 1$,则为马尔可夫模型。

HMM模型概念

- 概念
- 假设
- 过程

- 在HMM中观察到的事件是状态的随机函数，其中状态转移过程是隐蔽的马尔可夫链，而可观察的事件的随机过程是隐蔽的状态转换过程的随机函数。
- 一个HMM可表达为 $\lambda=(N,M,\pi,A,B)$ ，其参数含义如下

1) N: 模型中状态的数目。状态的集合 $S=\{S_1,S_2,\cdots,S_N\}$ 。记N个状态为 $\theta_1,\cdots,\theta_N$ ，记t时刻马尔可夫链所处状态为 q_t ，显然 $q_t\in(\theta_1,\cdots,\theta_N)$ 。

HMM模型概念

- 概念
- 假设
- 过程

2) M: 每个状态对应的可能的观察值数目。观测符号集合$V=\{v_1,v_2,\cdots,v_M\}$。

3) T:观测序列的长度值，有观测序列$O=\{O_1,O_2,\cdots,O_T\}$,

令t时刻观察到的观察值为o_t，其中$o_t\in(V_1,\cdots,V_M)$。

4) π: 初始状态概率$\pi=(\pi_1,\cdots,\pi_N)$，有

$$\pi_i=P(q_1=\theta_i),\ 1\le i\le N$$

5) A:是与时间无关的状态转移概率矩阵$(a_{ij})_{N\times N}$

$$a_{ij}=P(q_{t+1}=\theta_j|q_t=\theta_i)\ \ 1\le i,j\le N$$

6) B: 为观察值概率矩阵$(b_{jk})_{N\times N}$

$$b_{jk}=P(o_t=V_k|q_t=\theta_j)1\le j\le N,1\le k\le M$$

HMM模型概念

- 概念
- 假设
- 过程

- HMM模型的假设
  - 假设1：有限历史假设

$$P(q_i|q_{i-1}\cdots q_1)=P(q_i|q_{i-1})$$

  - 假设2：齐次性假设(状态与具体时间无关)

$P(q_{i+1}|q_i)=P(q_{j+1}|q_j)$ 对于任意 i,j成立

  - 假设3：输出独立性假设(输出仅与当前状态有关)

$$P(o_1,\cdots,o_T|q_1,\cdots,q_T)=\prod P(o_t|q_t)$$

- 假设一个HMM模型从n=1时刻开始运行，在n=1～N 诸时刻所给出的N个随机矢量y_n构成一个广义N维行向量即矩阵$Y=[y_1, y_2,\ldots, y_N]$。在HMM模型中，每次运行过程中所得到的马尔可夫链X对外界而言是看不见的，我们能够观测到值的只是Y；也即使是说，HMM模型的状态必须通过观察序列的随机过程才能表现出来

HMM模型概念

- 概念
- 假设
- 过程

1. 根据初始状态分布概率π ，设定初始状态$n=1$，令 。
2. 根据B，得出$S_i(n=1)$状态下输出的概率分布$b_{m1}(n=1$时) 。
3. 根据A，由n时刻的S_i状态转移到$n=n+1$时为S_j 状态的转移概率分布，来得到下一个状态，并置$n=n+1$ 。
4. 如果$n<N$ ，则回到第2步，否则结束。

HMM的基本算法

- **HMM需解决的三个问题**
  - 解决问题1
    - **前向-后向算法**
    - **后向算法**
  - 解决问题2
    - **Viterbi算法**
  - 解决问题3
    - **Baum-Welch算法**

- HMM 用于识别时，需要解决三个问题

1) 给定观测序列$O=(o_1, o_{2,\ldots,} o_T)$及模型$\lambda=(\pi,A,B)$，计算出$O$ 序列出现的概率$P(O|\lambda)$;

2) 给定观测序$O=(o_1, o_{2,\ldots,} o_T)$及模型$\lambda=(\pi,A,B)$，计算出产生此$O$时最可能经历的状态$S=(o_1, o_{2,\ldots,} o_T)$。这是一个识别问题，对于给定的$O$输出所有可能的路径中概率最大的路径。

3) 根据模型的若干输出O进行反复修正模型的参数，优化模型参数$\lambda=(\pi,A,B)$，使$P(O|\lambda)$最大。

HMM的基本算法

- **解决问题1**
  - **前向-后向算法**
- **解决问题2**
  - **Viterbi算法**
- **解决问题3**
  - **Baum-Welch算法**

- 给定观测序列$O=(o_1, o_{2,\ldots,} o_T)$及模型$\lambda=$（$\pi,A,B$），计算出$O$ 序列出现的概率$P(O|\lambda)$;
- 前向-后向算法的核心思想是对于HMM的参数先进行一个初始估计，通过给定的数据评估这些参数的价值并减少错误来不断修、调整正这些HMM参数。

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

- **前向算法**

对于一个固定状态序列$Q=(q_1, q_{2,\ldots,} q_T)$, 有

$$P(O|Q,\lambda)=\prod_{t=1}^{T}P(o_t|q_t,\lambda)=b_{q_1}(o_1)b_{q_2}(o_2)\cdots b_{q_T}(o_T)$$

其中$b_{q_t}(o_t)$表示q_t状态下观测到o_t的概率。

$$b_{q_t}(o_t)=b_{jk}\Big|_{q_t=\theta_j,o_t=V_k}\quad 1\le t\le T$$

对于给定的λ, 产生Q的概率

$$P(Q|\lambda)=\pi_{q_1}a_{q_1q_2}a_{q_2q_3}\cdots a_{q_{T-1}q_T}$$

因此，所求概率为

$$\begin{aligned}P(O|\lambda)&=\sum_{\forall Q}P(O,Q|\lambda)=\sum_{\forall Q}P(O|Q,\lambda)\cdot P(Q|\lambda)\\&=\sum_{\forall Q}\pi_{q_1}b_{q_1}(o_1)a_{q_1q_2}b_{q_2}(o_2)a_{q_2q_3}\cdots a_{q_{T-1}q_T}b_{q_T}(o_T)\end{aligned}$$

由此可以看见其计算复杂度非常大，为$2TN^T$

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

- **简化复杂度**
  - 定义**前向变量** $\alpha_t(i)$, 表示输出o序列在t时刻处于状态i 的输出概率。

$$\alpha_t(i) = P(o_1, o_2, \cdots, o_t, q_t = i|\lambda)$$

  - **前向变量**的性质
    - 初值 $\alpha_1(i) = P(o_1, q_1 = i) = \pi_i b_i(o_1), \ 1 \le i \le N$
    - 递推 根据 $P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda)$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}) \quad 2 \le t \le T-1, 1 \le j \le N$$

    - 最后有 $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$ $\quad b_j(o_{t+1}) = b_{jk}\big|_{o_{t+1}=V_k}$
    - 通过该方法可以简化计算复杂度，原因在于，每一次 $\alpha_t(i)$，都可以用 $\alpha_{t-1}(i)$ 来计算，不用重复计算。

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

前向算法是按输出观察值序列的时间，从前往后顺序的递推计算输出概率，其中

$O = o_1, o_2, \cdots, o_T$ 为输出的观察序列

$P(O|\lambda)$为给定模型λ时，输出序列O的概率

a_{ij}为从状态S_i开始直到状态S_j的转移概率

$b_{ij}(o_t)$为从状态S_i开始直到状态S_j发生转移时输出o_t的概率

$\alpha_t(j)$为前向概率。即输出序列$o_1, o_2, \cdots, o_t$并且到达状态S_j的概率

由上面符号的定义，则 可有下面的递推公式计算得到：

（1）初始化 $\alpha_0(1) = 1,\ \alpha_0(j) = 0 \quad (j \neq 1)$ ，

（2）递推公式 $\alpha_t(j) = \sum_i \alpha_{t-1}(i) a_{ij} b_{ij}(o_t) \ (t = 1,2,\cdots,T \ \ i, j = 1,2,\cdots,N)$

（3）最后结果 $P(O|\lambda) = \alpha_T(N)$

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

- 利用前向递推算法计算HMM模型在$\lambda=(\pi,A,B)$条件下，输出观察符号序列$O=(o_1, o_2, \ldots, o_T)$的概率$P(O|\lambda)$，其计算过程为
  A. 对每个状态赋予数组变量$\alpha_t(j)$，初始化状态S_1的数组变量$\alpha_0(1)$为1，其它状态数组变量$\alpha_0(j)$为0；
  B. 计算t时刻输出的观察值o_t的概率为$\alpha_t(j)$

$$\alpha_t(j)=\sum_i \alpha_{t-1}(i)a_{ij}b_{ij}(o_t)=\alpha_{t-1}(1)a_{1j}b_{1j}(o_t)+\alpha_{t-1}(2)a_{2j}b_{2j}(o_t)+\cdots$$
$$+\alpha_{t-1}(N)a_{Nj}b_{Nj}(o_t) \quad (j=1,2,\cdots,N)$$

当状态S_i到状态S_j没有转移时，$a_{ij}=0$

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

C. 当t ≠T时转到B，否则继续下一步

D. 获得$\alpha_T(N)$的值，有$P(O|\lambda)= \alpha_T(N)$

$$\alpha_{t+1}(j)=\sum_{i=1}\alpha_t(i)a_{ij}b_{ij}(o_{t+1})=\alpha_t(1)a_{1j}b_{1j}(o_{t+1})+\alpha_t(2)a_{2j}b_{2j}(o_{t+1})+\cdots$$
$$+\alpha_t(N)a_{Nj}b_{Nj}(o_{t+1}) \quad (j=1,2,\cdots,N,1\le t\le T-1)$$

$\alpha_t(i)a_{ij}$ 表示t时刻的观测符号序列{O_1, $O_{2,\ldots,}$ O_t}，并由t时刻S_i转移到t+1时刻的状态S_j发生的概率。

$\sum_{i=1}\alpha_t(i)a_{ij}$ 表示观测到的符号序列{O_1, $O_{2,\ldots,}$ O_t}在t+1时刻处于状态S_j发生的概率；

$\alpha_{t+1}(j)=\sum_{i=1}\alpha_t(i)a_{ij}b_{ij}(o_{t+1})$ 表示给定模型下，产生t+1以前的部分观测符号序列(包括t+1在内) {O_1, $O_{2,\ldots,}$ O_t , O_{t+1}},且时刻又处于状态S_j的概率。

$P(O/\lambda)=\sum_{i=1}^{N}a_T(i)$ 将所有的$a_T(i)$ 对i求和

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

- **后向算法**

类似地，定义后向变量

$$\beta_t(i)=P(o_{t+1},o_{t+2},\cdots,o_T,q_t=i|\lambda)\quad 1\le t\le T-1$$

- 初始化 $\beta_T(i)=1\quad 1\le t\le T$
- 递归 $\beta_t(i)=\sum_{i=1}^{N}a_{ij}b_j(o_{t+1})\beta_{t+1}(j)\quad t=T-1,T-2,\cdots,1,1\le i\le N$
- 终结 $P(O|\lambda)=\sum_{i=1}^{N}\beta_1(i)$

- *后向算法与前向算法很相似，后向算法是按输出序列的时间从后往前递推出概率值*

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

其中，

$O=(o_1, o_{2,\ldots}, o_T)$为输出的观察符号序列；

$P(O|M)$为给定模型M时，输出符号序列O的概率；

a_{ij}为从状态S_i开始直到状态S_j的转移概率；

$b_{ij}(o_t)$为从状态S_i开始直到状态S_j发生转移时输出的o_t概率；

$\beta_t(i)$ 为后向概率。从状S_i开始直到状态S_N结束输出o_{t+1}, $o_{t+2,\ldots}, o_T$ 序列的概率。

$\beta_t(i)$ 可由下面的递推公式计算得到

（1）初始化 $\beta_T(N)=1, \quad \beta_T(j)=0 \quad (j \neq N)$

（2）递推公式

$$\beta_t(i)=\sum_j \beta_{t+1}(j)a_{ij}b_{ij}(o_{t+1})\beta_t(i)=\sum_j \beta_{t+1}(j)a_{ij}b_{ij}(o_{t+1}) \quad t=T,T+1,\cdots,1 \qquad i,j=1,2,\cdots,N$$

（3）最后结果 $P(O/M)=\sum_{i=1}^{N}\beta_1(i)\pi_i=\beta_0(1)$

后向算法的计算量在N^2T数量级。根据前向和后向概率的定义，有

$$P(O/\lambda)=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_{ij}(o_{t+1})\beta_{t+1}(j) \quad 1\le t\le T-1$$

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

- 给定观察序$O=(o_1, o_{2,\ldots,} o_T)$以及一个模型$\lambda=(\pi,A,B)$时，怎样寻找满足这种观察序列意义最优的隐含状态序列Q。定义3个符号如下：
  - $\delta t(j)$，表示t时刻处于状态j下，沿路径$q_1\ q_1\ldots\ q_t$输出$o_1, o_{2,\ldots,} o_T$最大概率；于是有

$$\delta_t(j)=\max_{1\le i\le N}\left[\delta_{t-1}(i)a_{ij}\right]b_j(o_t)$$

  - $\phi_t(j)$，表示的是一个状态值，该状态值产生了上面的输出

$$\phi_t(j)=\arg\max_{1\le i\le N}\left[\delta_{t-1}(i)a_{ij}\right]$$

  - q_t^*，表示在观察时刻t中所有的状态中，最大的那个状态，$q_t^*=\phi_{t+1}(q_{t+1}^*)$。
- 采用Viterbi算法在当已知观察序列下，求解最优状态序列时与前面讲求最大观察值概率的算法相似。区别是，在求概率时不再是将其来源相加，而是取其中最大的那个。

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

- 对于给定的可观察状态序列 $O = o_1, o_2, \ldots, o_T$而言，很难直接得到最优的HMM 参数$\lambda$使得$P(O|\lambda)$，它是泛函极值问题。 Baum-Welch算法利用递归思想，刚开始时候对参数λ进行初始估计，再通过对于给定的数据评估参数λ的有效性，减少其所引起的错误，使得$P(O|\lambda)$局部最大，通过不断更新参数λ使得和给定的训练数据的误差变小，最终获得$\lambda=(\pi, A, B)$。

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

定义$\varepsilon_t(i,j)$为为给定训练观察序列O和参数模型λ时，时刻t马尔可夫链处于状态i，而时刻$t+1$处于状态j的概率：

$$\varepsilon_t(i,j)=P(S_t=\theta_i,S_{t+1}=\theta_j|O,\lambda)$$

根据前向和后向变量的定义

$$\alpha_t(i)=P(o_1,o_2,\cdots,o_t,s_t=\theta_i|\lambda)$$

$$\beta_t(i)=P(o_{t+1},o_{t+2},\cdots,o_T|s_t=\theta_i,\lambda)$$

$$\varepsilon_t(i,j)=\frac{P(S_t=\theta_i,S_{t+1}=\theta_j,O|\lambda)}{P(O|\lambda)}=\frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}=\frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}$$

定义$\varepsilon t(i)$为给定观察序列O和参数模型λ，t时刻处于状态i的概率

$$\varepsilon_t(i)=P(S_t=\theta_i|O,\lambda)=\sum_{j=1}^{N}\varepsilon_t(i,j)$$

HMM的基本算法

- 解决问题1
  - 前向-后向算法
- 解决问题2
  - Viterbi算法
- 解决问题3
  - Baum-Welch算法

其中 $\sum_{t=1}^{T}\varepsilon_t(i)$ 为从状态i开始出发的状态的转移数的期望值；$\sum_{t=1}^{T}\varepsilon_t(i,j)$是从状态$i$转换到$j$状态的转移数的期望值。并且有

$$\overline{\pi}=\varepsilon_1(i)$$

$$\overline{a_{ij}}=\frac{\sum_{i=1}^{T-1}\varepsilon_t(i,j)}{\sum_{i=1}^{T-1}\varepsilon_t(i)}$$

$$\overline{b_j}(k)=\frac{\sum_{t=1,s,t,o_t=v_k}^{T}\varepsilon_t(i,j)}{\sum_{i=1}^{T}\varepsilon_t(i)}$$

HMM参数的计算过程为根据观察序列O和参数模型$\lambda=(\pi,A,B)$，由重估公式得到新的一组参数 $\overline{\pi},\overline{a}_{ij},\overline{b}_j(k)$ ，这些参数属于新的模型 $\overline{\lambda}=(\overline{A},\overline{B},\overline{\pi})$ ，并且这些新得到的$\overline{\lambda}$要比λ更好。

建立网络态势评估模型

- **网络态势评估模型举例**

- 系统状态
  - 安全状态
  - 受攻击状态
- 为网络中的每台主机建立一个用于威胁评估的 HMM模型，具体如下：

(1)状态空间 $\phi = \{0,1\}$，这里用0表示主机处于安全状态，1表示主机处于受攻击状态，状态数 N=2；

(2)观察符号空间V={V_0 ,V_1} ，这里V_0表示没有攻击事件发生， V_1表示有攻击事件发生，每个状态对应的可能的观察值符号数目M=2 ；

(3)初始状态概率分布π ={π_0, π_1}，其中π_i=P{主机初始时处于状态 i}，$i \in \phi$；

(4)状态转移概率分布

$$A = [a_{i,j}]_{N \times N} = \begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix}$$

其中$a_{i,j}$=P{下一步处于状态j|*当前*处于状态 i}}，$i, j \in \phi$ 。

建立网络态势评估模型

- **网络态势评估模型举例**

(5)观察值概率分布

$$B=[b_i(V_k)]_{N\times M}=\begin{bmatrix} b_0(V_0) & b_0(V_1) \\ b_1(V_0) & b_1(V_1) \end{bmatrix}$$

$b_0(V_1)$为IDS的误报率， $b_1(V_0)$为IDS的漏报率。

(6)观察符号序列$O=o_1, o_{2,\ldots,} o_T$，其中 $O_t \in V, t=1,2,\cdots T$ ，为t时刻观测到的观察符号。

于是基于HMM的网络态势评估模型可以用 $\lambda=\{\phi, V, A, B, \pi\}$ 来表示。

- 利用 HMM对模型参数进行估计可减少计算量提高系统的实时性。在实际的网络环境中，对于不同的时间段网络攻击特征有较大的不同。需要算法在对参数的估计过程中有良好的自适应性、动态性才能保证评估结果的准确性。
- 利用前一时间段收集到的观察序列作为训练数据集D_A，训练出模型λ，λ反映了D_A的特性。再将下一时间段所收集到的新的观察序列再进行训练，并将此序列的特性反映在模型λ中，以此不断修正模型参数。

建立网络态势评估模型

- **网络态势评估模型举例**

- 设初始时的训练序列为$O^{(0)}$，将前 n-1 个时间段收集到的观察序列也作为训练序列，得到 n-1个训练序列$O^{(1)}$, $O^{(2)}$, …, $O^{(n-1)}$，， 则在第 n个时间段内的状态转移概率分布为：

$$\hat{a}_{i,j}(n)=\frac{\sum_{l=0}^{n-1} trans-counts(i,j,l)}{\sum_{l=0}^{n-1} state-counts(i,l)}$$

- 其中 $trans-counts(i,j,l)$ 为序列$O^{(1)}$中从状态i转换为状态j的次数期望， $state-counts(i,l)$为处于状态i的次数期望，采用Baum-Welch算法求解。

建立网络态势评估模型

- **网络态势评估模型举例**

在建立了 HMM后，利用 HMM和观察符号序列计算主机处于安全状态和受攻击状态的概率。在t时刻，主机状态概率分布的计算公式可以表述为：

$$
\begin{aligned}
r_t(i) &= P(\text{时刻 } t\text{主机状 态为}i \mid O_1, O_2, \cdots, O_t, \lambda) \\
&= \frac{P(O_1, O_2, \cdots, O_t, \text{时刻}t\text{主 机 状 态为}i \mid \lambda)}{P\{O_1, O_2, ..., O_t \mid \lambda)} \\
&= \frac{P(O_1, O_2, \cdots, O_t, \text{时刻}t\text{主机状态为}i \mid \lambda)}{\sum_{j \in \Phi} P\{O_1, O_2, ..., O_t, \text{时刻}t\text{主机状态为}j \mid \lambda)} \\
&= \frac{a_t(i)}{\sum a_t(i)}, t \geq 1, i \in \Phi
\end{aligned}
\qquad (7\text{-}35)
$$

由前向变量的定义，在初始时刻，即t=1时 $\alpha_1(i) = \pi_j b_j(O_1), j \in \Phi$

当t>1时，

$$\alpha_t(j) = \sum_{i \in \Phi} \alpha_{i-1}(i) a_{i,j} b_j O_t, j \in \Phi \qquad (7\text{-}37)$$

随着t的增加，$\alpha_t(j)$ 的值明显降低。为了防止当t相当大时造成运算下溢，可添加比例因子。

建立网络态势评估模型

- **网络态势评估模型举例**

设 $\overline{\alpha}_{t+1}(j)=[\sum_{j\in\Phi} r_t(i)a_{i,j}]b_j(O_{t+1}),t\geq 1]$ ，由公式(7-35)和(7-37)可得

$$\frac{\overline{\alpha}_{t+1}(i)}{\sum_{i\in\Phi}\overline{\alpha}_{t+1}(i)}=\frac{\overline{\alpha}_{t+1}(i)}{\sum_{i\in\Phi}\overline{\alpha}_{t+1}(i)},t\geq 1$$

通过上述分析，可得t时刻主机处于状态i的概率

$$\gamma(i)=\begin{cases}\dfrac{\alpha_1(i)}{\sum_{i\in\Phi}a_1(i)},t=1,i\in\Phi\\ \dfrac{\overline{\alpha}_t(i)}{\sum_{i\in\Phi}\overline{\alpha}_t(i)},t>1,i\in\Phi\end{cases}$$

小结

- 本章首先阐述了网络安全态势评估相关的概念和重要意义。在此基础上介绍了几种数据挖掘的常用算法应用到网络态势这一领域，包括*SVM方法、贝叶斯网络方法、隐马尔科夫方法*等。这些方法是数据挖掘中非常优秀的算法，在处理不确定信息的智能化系统中已得到了重要的应用，已成功地用于医疗诊断、统计决策、专家系统，学习预测等领域。
- 作为一个新兴的热点，网络安全态势的研究将是一个长期的过程。

Thanks!