复杂网络上的学习

张长水 王世军
清华大学　 自动化系
zcs@mail.tsinghua.edu.cn
2006.11.5

社会学发现

任务：送信
目标：住在Boston地区的人A
规则：
如果：认识A，就寄给他。
否则：每个人寄给自己的朋友

六度分割

- 从寄信人开始平均只需要6次传递
- 社会学中的6度分割
- six degrees of separation
- 小世界现象。

小世界网络

- 称短路径和高聚集的网络称为小世界网络
- 许多真实网络符合这一特性。

研究目标

- 直觉：复杂系统一定存在一些在某种层次上，与其结构相联系的组织规律。
- 开发出能够定量地度量这种深层组织规律的工具。
- 发现机理

网络的度量指标

- 聚集系数C
- 路径长度L
- 度数分布P(k)

聚集系数C

- 度量了网络中连接的平均局部密度。
- 在朋友网络中，C大意味着某人的多个朋友之间往往也相互是朋友。

$$C_i = \frac{E_i}{k_i(k_i - 1)/2}$$

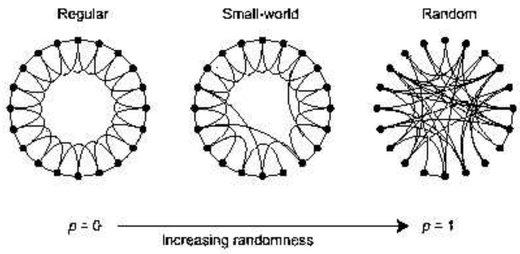
平均路径长度L

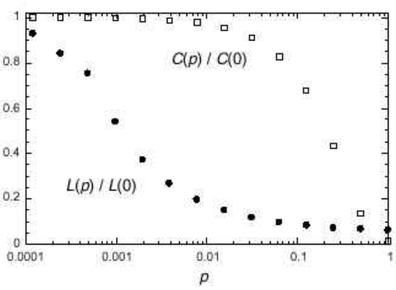
- 网络中所有连通节点之间路径长度的平均值。
- 路径长度：从某点要最少经过几个连接跳转到另一点。

度数分布

- 一个节点的度数指与之相连的边数。
- 度数分布就是具有度数k的节点的个数（或比例）随k的分布。

WS（Watts-Strogatz ）模型

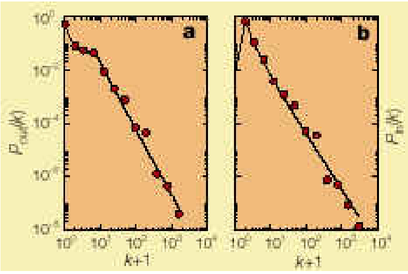




三种结构的比较

	规则图	随机图	小世界
路径长度	大	小	小
聚集性	高	低	较高
度数分布	常数	Possion分布	不涉及/其他分布

Scale-free分布 $P(k) \sim k^{-\gamma}$



WWW页面/超级链接

The Topology of Real Networks:Empirical Result

普遍存在大C、小L和类scale-free分布

生物学：

- 细胞、组织等的新陈代谢及相关化学反映；
- 蛋白质分子网络
- 神经网络结构（C线形虫）
- 生态系统、食物链

The Topology of Real Networks:Empirical Result

社会学

- 六度分割朋友网络
- 演员合作网络
- 科学家合作网络
- 引文网络网络
- 性接触网络
- 语言学（单词在句子中的共现）
- 电话呼叫网

The Topology of Real Networks:Empirical Result

- 人造网络——电网、机场/航线
- 物理学——力偶共振网络、集成电路
- 计算机科学——地图着色问题、囚徒困境问题、元胞自动机
- 传染病的研究——模型及传播的动态性能；谣言、信息传播等
- WWW及Internet的研究

SF模型

增长(growth)和优先粘贴(preferential attachment)是导致SF分布的两要素。

- 增长：从少量节点（m_0）开始，每个时间步增加一个新节点，并从该节点向系统中的其他m（$m \le m_0$）个节点引边。
- 优先粘贴：引边时，假设节点吸引边的概率与其度数成正比

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$

分类器网络

- 复杂网络上的每一个节点是一个分类器
- 怎样使得分类器性能更好？

分类器网络

- 集中式的分类器学习
- 假设：样本独立同分布

Network Boosting 算法

- 输入
  - 样本集 $Z=\langle (x_1,y_1),(x_2,y_2),\ldots,(x_l,y_l)\rangle$
  - 训练次数 T
  - 采样比例 ρ
  - 加权因子 β
- 初始化 $w_{k,1}(x_i)=1 \quad i=1,\ldots,l \quad k=1,\ldots,K$
- 训练循环
  - 有放回的加权采样，为每个节点生成训练集 $T_{k,t}$

Network Boosting 算法（续1）

- 使用采样生成的训练集 $T_{k,t}$ 训练每个节点的分类算法 ，得到分类器 $h_{k,t}: x\mapsto\{-1,+1\}$
- 更新每个节点的样本权重分布

$$w_{k,t+1}(i)=w_{k,t}(i)\beta^{I(h_{k,t}(x_i)=y_i)+\sum_n I(h_{n,t}(x_i)=y_i)}/Z_{k,t},$$

- 输出

$$F(x)=\frac{1}{KT}\sum_{k=1}^{K}\sum_{t=1}^{T}h_{k,t}(x)$$

$$h_{k,t}: x\mapsto\{-1,+1\} \quad t=1,\ldots,T \quad k=1,\ldots,K$$

算法收敛性

- **定理1**：对于分类器网络上的任意节点 k ，其样本集（环境）在 T 次博弈中的累计损失为：

$$\sum_{t=1}^{T}M(P_{k,t},Q_{k,t})\le c_\beta \ln l+\alpha_\beta\min_P\sum_{t=1}^{T}\{M(P,Q_{k,t})\}$$

$$+\alpha_\beta\left(\min_j\sum_{t=1}^{T}\sum_n M(j,Q_{n,t})-\sum_{t=1}^{T}\min_i\sum_n M(i,Q_{n,t})\right)$$

算法收敛性证明

- 对于 $t=1,\ldots,T$ ，由 $\beta^x\le 1-(1-\beta)x$ 我们有

$$\sum_{i=1}^{l}w_{k,t+1}(i)=\sum_{i=1}^{l}w_{k,t}(i)\beta^{M(i,Q_{k,t})+\sum_n M(i,Q_{n,t})}$$

$$\le\sum_{i=1}^{l}w_{k,t}(i)\left(1-(1-\beta)M(i,Q_{k,t})\right)\beta^{\sum_n M(i,Q_{n,t})}$$

$$\le\sum_{i=1}^{l}w_{k,t}(i)\left(1-(1-\beta)M(i,Q_{k,t})\right)\beta^{\min_i\sum_n M(i,Q_{n,t})}$$

$$=\left(\sum_{i=1}^{l}w_{k,t}(i)\right)\left(1-(1-\beta)M(P_{k,t},Q_{k,t})\right)\beta^{\min_i\sum_n M(i,Q_{n,t})}$$

算法收敛性证明（续1）

- 如果将上式不断沿时间 t 展开，那么我们可以得到：

$$\sum_{i=1}^{l}w_{k,t+1}(i)\le l\prod_{t=1}^{T}\left(1-(1-\beta)M(P_{k,t},Q_{k,t})\right)\prod_{t=1}^{T}\beta^{\min_i\sum_n M(i,Q_{k,t})}$$

$$=l\beta^{\sum_{t=1}^{T}\min_i\sum_n M(i,Q_{k,t})}\prod_{t=1}^{T}\left(1-(1-\beta)M(P_{k,t},Q_{k,t})\right)$$

算法收敛性证明（续2）

- 下面，考虑到对于任何样本j 有

$$\sum_{i=1}^{l} w_{k,T+1}(i) \ge w_{k,T+1}(j) \ge \beta^{\sum_{t=1}^{T} M(j,Q_{k,t})+\sum_{n} M(j,Q_{n,t})}$$

$$(\ln\beta)\sum_{t=1}^{T}\left\{M(j,Q_{k,t})+\sum_{n} M(j,Q_{n,t})\right\}$$

$$\le \ln l+\sum_{t=1}^{T}\ln\left(1-(1-\beta)M(P_{k,t},Q_{k,t})\right)+\ln\beta\sum_{t=1}^{T}\min_{i}\sum_{n} M(i,Q_{n,t})$$

$$\le \ln l-(1-\beta)\sum_{t=1}^{T} M(P_{k,t},Q_{k,t})+\ln\beta\sum_{t=1}^{T}\min_{i}\sum_{n} M(i,Q_{n,t})$$

算法收敛性证明（续3）

- 因为 $\ln(1-x)\le -x$ 对于 $x<1$ 成立，重新整理上式，并且注意到该表达式对于任何样本均成立，那么

$$\sum_{t=1}^{T} M(P_{k,t},Q_{k,t})$$

$$\le \frac{\ln l}{1-\beta}+\frac{\ln(1/\beta)}{1-\beta}\min_{j}\sum_{t=1}^{T}\left\{M(j,Q_{k,t})+\sum_{n} M(j,Q_{n,t})\right\}+\frac{\ln\beta}{1-\beta}\sum_{t=1}^{T}\min_{i}\sum_{n} M(i,Q_{n,t})$$

$$=\frac{\ln l}{1-\beta}+\frac{\ln(1/\beta)}{1-\beta}\min_{P}\sum_{t=1}^{T}\left\{M(P,Q_{k,t})\right\}$$

$$+\frac{\ln(1/\beta)}{1-\beta}\left(\min_{j}\sum_{t=1}^{T}\left\{\sum_{n} M(j,Q_{n,t})\right\}-\sum_{t=1}^{T}\min_{i}\sum_{n} M(i,Q_{n,t})\right)$$

算法收敛性

- **推论1**：在定理1的条件下，设β 为$\frac{1}{1+\sqrt{\frac{2\ln l}{T}}}$

  那么当 T 充分大时，节点 k 处的样本集（环境）所遭受的每轮平均损失为

$$\frac{1}{T}\sum_{t=1}^{T} M(P_{k,t},Q_{k,t})\le \min_{P}\frac{1}{T}\sum_{t=1}^{T}\left\{M(P,Q_{k,t})\right\}+\Delta_T$$

$$\Delta_T=\sqrt{\frac{2\ln l}{T}}+\frac{\ln l}{T}$$

算法收敛性

- **推论2**：在推论1的条件下，所有 K 个节点，在 T 轮反复博弈过程中，当 T 充分大时，所招致的平均期望损失为：

$$\frac{1}{KT}\sum_{k=1}^{K}\sum_{t=1}^{T} M(P_{k,t},Q_{k,t})\le v+\Delta_T$$

  其中 v 为博弈值（Game Value）

UCI数据集上的对比实验结果

Name	C4.5	Bagging	AdaBoost	BB	NB			
	$err \pm std$	$err \pm std$	$err \pm std$	$err \pm std$	S1	S2	S3	$err \pm std$
audiology	.2818 ± .0463	.2332 ± .0489	.2126 ± .0420	.2671 ± .0500		-	+	.2264 ± .0461
breast-w	.0610 ± .0125	.0412 ± .0101	.0319 ± .0074	.0339 ± .0101	+	-		.0356 ± .0082
colic	.1792 ± .0263	.1559 ± .0244	.1955 ± .0348	.1561 ± .0229		+		.1598 ± .0242
credit-a	.1695 ± .0176	.1400 ± .0169	.1391 ± .0158	.1337 ± .0173				.1368 ± .0166
diabetes	.2753 ± .0242	.2427 ± .0186	.2647 ± .0201	.2374 ± .0188		+	-	.2439 ± .0221
glass	.3353 ± .0459	.2852 ± .0443	.2572 ± .0493	.3003 ± .0444	+		+	.2662 ± .0410
heart-c	.2413 ± .0355	.2089 ± .0357	.1984 ± .0285	.1780 ± .0281	+	+		.1793 ± .0309
heart-h	.2155 ± .0338	.2008 ± .0282	.2028 ± .0307	.1906 ± .0315	+	+		.1897 ± .0303
hepatitis	.2200 ± .0479	.1829 ± .0387	.1697 ± .0402	.1700 ± .0383	+	+	+	.1563 ± .0359
iris	.0618 ± .0305	.0582 ± .0271	.0647 ± .0257	.0602 ± .0313				.0583 ± .0229
labor	.2139 ± .0899	.1687 ± .0790	.1430 ± .0733	.1652 ± .0865	+	+	+	.1096 ± .0548
lymph	.2500 ± .0484	.2150 ± .0459	.1790 ± .0372	.1840 ± .0485	+	+	+	.1618 ± .0460
soybean	.1259 ± .0223	.0936 ± .0205	.0843 ± .0171	.0927 ± .0200	+	+	+	.0742 ± .0144
vehicle	.2968 ± .0231	.2622 ± .0205	.2322 ± .0203	.2553 ± .0226	+		+	.2373 ± .0195
vote	.0518 ± .0164	.0403 ± .0136	.0541 ± .0158	.0415 ± .0129		+		.0435 ± .0132
waveform	.3142 ± .0395	.2108 ± .0417	.1758 ± .0337	.1727 ± .0303	+	+	+	.1621 ± .0286
average	.2058	.1712	.1628	.1649				.1526

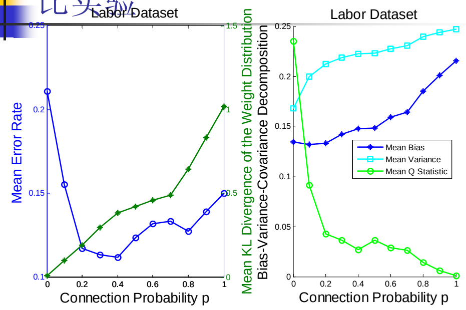
连接度数变化对于样本权重分布相关性的影响

- **定理2**：定义节点k在训练轮次t和t+1时的样本权重分布分别为$P_{k,t}$ 和 $P_{k,t+1}$ 。那么两分布之间的KL距离的上界为：

$$KL\left(P_{k,t}\,\|\,P_{k,t+1}\right)$$

$$\le\left\{\min_{i}\left[I\left(h_{k,t}(x_i)=y_i\right)+\sum_{n} I\left(h_{n,t}(x_i)=y_i\right)\right]-\max_{i}\left[I\left(h_{k,t}(x_i)=y_i\right)+\sum_{n} I\left(h_{n,t}(x_i)=y_i\right)\right]\right\}\log(\beta)$$

不同连接概率下的随机图上的对比实验



不同连接概率下，基于随机图结构的Network Boosting在labor数据集上的性能曲线。横轴为随机图的连接概率p。

加权竞争尺度无关网络上的对比实验

表 3.6 Network Boosting算法在加权竞争尺度无关网络上的对比实验

数据集	平均错误率		平均偏差		平均方差		平均Q统计量		平均KL距离	
	非加权	加权	非加权	加权	非加权	加权	非加权	加权	非加权	加权
breast-w	0.0315	0.0391	0.0294	0.0320	0.0608	0.0625	0.7526	0.7650	0.5486	0.8246
colic	0.1653	0.1517	0.1250	0.1195	0.1384	0.1414	0.5304	0.5210	0.6027	0.6564
credit-a	0.1384	0.1493	0.0987	0.1034	0.1235	0.1234	0.5817	0.5880	0.5926	0.7154
diabetes	0.2423	0.2293	0.1612	0.1557	0.1837	0.1833	0.3646	0.3600	0.3943	0.5847
glass	0.3094	0.2965	1.3354	1.3487	1.6886	1.7672	0.3561	0.3403	0.3467	0.5711
heart-c	0.1843	0.1694	0.1377	0.1283	0.1653	0.1712	0.4310	0.3740	0.3971	0.6402
heart-h	0.2060	0.1872	0.1380	0.1332	0.1465	0.1436	0.5085	0.5197	0.5239	0.6782
hepatitis	0.1984	0.1839	0.1251	0.1150	0.1393	0.1440	0.5241	0.4826	0.4833	0.6758
iris	0.0600	0.0417	0.0455	0.0352	0.0836	0.0977	0.6144	0.4744	0.4936	0.8229
labor	0.1773	0.2091	0.1289	0.1429	0.1919	0.1906	0.1320	0.1580	0.3120	0.6092
lymph	0.2051	0.1831	0.1925	0.1945	0.2279	0.2367	0.3583	0.3336	0.4081	0.7051

分布式环境中的分类器网络

- 假设：数据独立同分布
- 如何尽可能少的通讯

分布式Network Boosting算法

- 输入
  - 分类器网络N，节点数目K
  - 训练集 S_k $k=1,2,...,K$
  - 权重更新参数 β
  - 训练轮次T
- 初始化样本权重 $D_{k,1}(x_i)=1/l_k$ $k=1,...,K$
- 训练循环
  - 有放回的加权采样，为每个节点生成训练集 $T_{k,t}$

分布式Network Boosting 算法（续1）

- 使用采样生成的训练集$T_{k,t}$ 训练每个节点的分类算法， 得到分类器$h_{k,t}: x \mapsto \{-1,+1\}$
- 更新每个节点的样本权重分布

$$D_{k,t+1}(i)=D_{k,t}(i)\beta^{\left[-\alpha_{k,t}y_i h_{k,t}(x)-\sum_n \alpha_{n,t}y_i h_{n,t}(x)\right]}/Z_{k,t}$$

$$\alpha_{k,t}=0.5*\log\left((1-\varepsilon_k)/\varepsilon_k\right), \qquad k=1,...,K$$

- 输出

$$H_{k,T}(x)=\sum_{t=1}^{T}\left[\alpha_{k,t}h_{k,t}(x)+\sum_n \alpha_{n,t}h_{n,t}(x)\right] \qquad k=1,...,K$$

假设空间中的协作梯度下降

- 对于任意节点k，我们定义在训练轮次t该节点关于样本x的集群输出为：

$$H_{k,t}(x)=\sum_{s=1}^{t}\alpha_{k,s}h_{k,s}(x)+\sum_n\sum_{s=1}^{t}\alpha_{n,s}h_{n,s}(x)$$

- 基于样本间隔的代价函数为

$$C(H_{k,t}):=\frac{1}{m}\sum_{i=1}^{m}C\left(y_i H_{k,t}(x_i)\right)$$
$$=\frac{1}{m}\sum_{i=1}^{m}\exp\left\{-y_i\sum_{s=1}^{t}\alpha_{k,s}h_{k,s}(x_i)-y_i\sum_n\sum_{s=1}^{t}\alpha_{n,s}h_{n,s}(x_i)\right\}$$

假设空间中的协作梯度下降（续1）

- 在t+1轮，我们希望节点k处所学习到的假设$h_{k,t+1}$ 和代价函数C在 $H_{k,t}$ 处的负梯度的内积最大化

$$-\left\langle \nabla C\left(H_{k,t}\right), h_{k,t+1}\right\rangle$$
$$=-\frac{1}{m^2}\sum_{i}^{m} y_i h_{k,t+1}\left(x_i\right) C'\left(y_i H_{k,t}\left(x_i\right)\right)$$
$$=\frac{1}{m^2}\sum_{i}^{m} y_i h_{k,t+1}\left(x_i\right)\exp\left\{-y_i\sum_{s=1}^{t}\alpha_{k,s}h_{k,s}\left(x\right)-y_i\sum_{n}\sum_{s=1}^{t}\alpha_{n,s}h_{n,s}\left(x\right)\right\}$$

假设空间中的协作梯度下降（续2）

- 经简单变换后我们有

$$-\left\langle \nabla C\left(H_{k,t}\right), h_{k,t+1}\right\rangle=\frac{1}{m^2}\sum_{i:y_i=h_{k,t+1}(x_i)}^{m}\exp\left\{-y_i\sum_{s=1}^{t}\alpha_{k,s}h_{k,s}\left(x\right)-y_i\sum_{n}\sum_{s=1}^{t}\alpha_{n,s}h_{n,s}\left(x\right)\right\}$$
$$-\frac{1}{m^2}\sum_{i:y_i\neq h_{k,t+1}(x_i)}^{m}\exp\left\{-y_i\sum_{s=1}^{t}\alpha_{k,s}h_{k,s}\left(x\right)-y_i\sum_{n}\sum_{s=1}^{t}\alpha_{n,s}h_{n,s}\left(x\right)\right\}$$

- 如果我们定义节点k处样本x_i 在t+1轮的权重为

$$D_{k,t}\left(i\right)=\exp\left\{-y_i\sum_{s=1}^{t}\alpha_{k,s}h_{k,s}\left(x_i\right)-y_i\sum_{n}\sum_{s=1}^{t}\alpha_{n,s}h_{n,s}\left(x_i\right)\right\}$$

分类性能分析—训练错误

- **定理3**：在分布式应用环境中，假设训练轮次t时，节点k处样本的权重更新采用如下方式：

$$D_{k,t+1}\left(i\right)=D_{k,t}\left(i\right)\exp\left\{-y_i\alpha_{k,t}h_{k,t}\left(x_i\right)-y_i\sum_{n}\alpha_{n,s}h_{n,t}\left(x_i\right)\right\}/Z_{k,t}$$

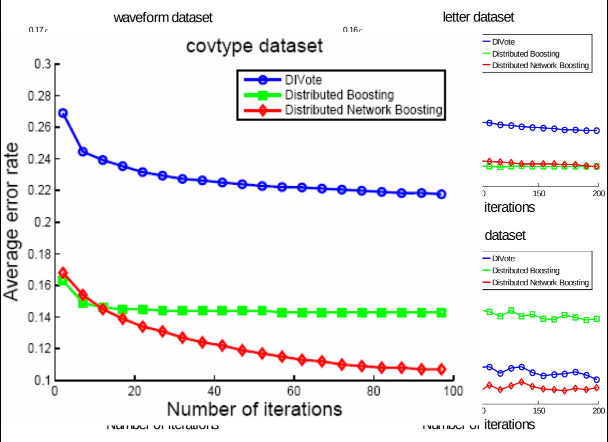
经过T轮训练后，训练错误的上界为

$$\prod_{t=1}^{T}Z_{k,t}=\prod_{t=1}^{T}\sum_{i=1}^{l_k}D_{k,t}(i)\exp\left(-\alpha_{k,t}y_ih_{k,t}(x_i)-\sum_{n}\alpha_{n,t}y_ih_{n,t}(x_i)\right)$$

分类性能分析—泛化错误

- **定理4**：对于分类器网络上的任一节点k，假定基假设空间的VC维为d。那么在训练轮次T对于任意 δ，$\delta\in(0,1)$，节点k及其近邻节点所学习到的集合假设以概率对于任意$1-\delta$ 有：$\theta>0$

$$P_D\left[yH_{k,T}(x)\le 0\right]\le P_{S_k}\left[yH_{k,T}(x)\le\theta\right]+O\left(\frac{1}{\sqrt{l}}\left(\frac{d\log^2(l/d)}{\theta^2}+\log(\frac{1}{\delta})\right)^{1/2}\right)$$
$$\le\exp\left(\theta\left(\sum_{t=1}^{T}\alpha_{k,t}+\sum_{n}\sum_{t=1}^{T}\alpha_{n,t}\right)\sum_{k=1}^{K}\sum_{t=1}^{T}\alpha_{k,t}\right)\left(\prod_{t=1}^{T}Z_{k,t}\right)+O\left(\frac{1}{\sqrt{l}}\left(\frac{d\log^2(l/d)}{\theta^2}+\log(\frac{1}{\delta})\right)^{1/2}\right)$$



分布式半监督Network Boosting算法

- 初始标签和初始权重
- 采用每个节点及其近邻节点所学到的假设集合的加权，作为未标样本标签近似

$$y_{k,i}=H_{k,t}\left(x_i\right)=\sum_{s=1}^{t}\left[\alpha_{k,s}h_{k,s}\left(x_i\right)+\sum_{n}\alpha_{n,s}h_{n,s}\left(x_i\right)\right]$$

- 更新权重

$$D_{k,t+1}(i)=D_{k,t}(i)\beta^{\left[-\alpha_{k,t}y_ih_{k,t}(x)-\sum_{n}\alpha_{n,t}y_ih_{n,t}(x)\right]}/Z_{k,t}$$

- 采样，生成新的假设。

半监督对比实验结果

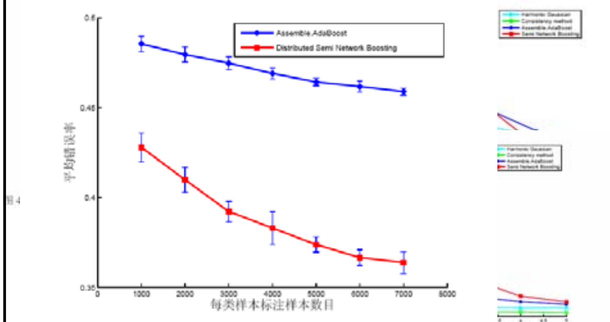


图 4.10 SecStr数据集上不同数目标注样本情况下的对比实验

References

- Shijun Wang, Zhongbao Kou and Changshui Zhang (2006). "Network Boosting on Different Networks." Physica A, 366(1): 561-570
- Shijun Wang and Changshui Zhang (2005). "Microscopic Model of Financial Markets Based on Belief Propagation." *Physica A* **354C**: 496-504
- Shijun Wang and Changshui Zhang (2004). "Weighted Competition Scale-Free Network." *Phys. Rev. E* **70**: 066127
- Zhongbao Kou and Changshui Zhang (2003). "Reply Networks on a Bulletin Board System." *Phys. Rev. E* **67**: 036117
- Shijun Wang and Changshui Zhang. **Network Game and Boosting.** In: João Gama, Rui Camacho, Pavel B. Brazdil, Alípio M. Jorge and Luís Torgo (Eds.), Lecture Notes in Computer Science, Poceedings, Springer-Verlag GmbH, ISBN 3-540-29243-8, vol. 3720 / 2005: pp. 461-472. The 16th European Conference on Machine Learning (ECML 2005), Porto, Portugal, October 3-7, 2005.

Graph model selection using maximum likelihood （ICML 2006）

Ranking Graph Models

- A model P assigns a probability $p(G)$ to every graph G.
- The quantity $P(G)$ is called the likelihood. We score a model by the log-likelihood $-\log(P(G))$
- The Maximum Likelihhood Estimate prefers the model with the largest likelihood.

Preliminaries

- Directed graph $G=(V,E)$
  - V: vertex set
  - E: edge set
- $n=|V|$ is the number of vertices
- $m=|E|$ is the number of edges
- For a node v, $in(V)$ is its indegree; $out(V)$ is its outdegree.

Graph Models

- Erdos-Renyi Random model (ER)
- Power-law Random model (PRG)
- Preferential Attachment model (PA)
- Small-world model (SW)

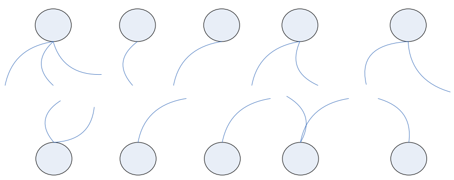
Erdos-Renyi Random model (ER)

- For every pair of vertices u, v and edge from u to v appears independently with probability $p \in [0,1]$

Power-law model (PRG)

- Power-law exponents: β_{in} β_{out} $p(k) = k^{-\beta}$
- Cutoffs: c_{in} c_{out}
- Generate the indegrees and outdegrees of each of the n vertices independently according to the power-law distribution.
- Connect the vertices randomly.

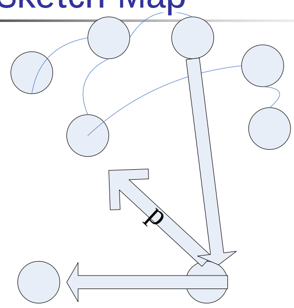
PRG Sketch Map



Preferential Attachment model (PA)

- Start with a single vertex.
- In each iteration i=2,…,n a random vertex “appears”.
- With probability p an edge from the new vertex is created and the other end-point v is chosen with probability proportional to $in(v)+\gamma$
- With probability 1-p-q the process stops and new iteration i+1 begins.

PA Sketch Map



Small-world model (SW)

- The basic idea is to add random links to an underlying well-structured graph, in our case the grid.
- For every pair of vertices u, v, an edge from u to v is added with probability

$$\alpha \times dist(u,v)^{-\beta}$$

ER Algorithm

- n: the number of nodes
- m: the number of links
- n(n-1)-m: the number of non-edges
- The probability that the ER model generates G is

$$\Pr(G) = p^m (1-p)^{n(n-1)-m}$$

$$p = \frac{m}{n(n-1)}$$

PRG Algorithm

- In the PRG model a vertex gets assigned an indegree $d \in \{0,\ldots,c_{in}\}$ with probability

$$d^{-\beta_{in}} / Z_{in} \qquad Z_{in} := \sum_{d=1}^{c_{in}} d^{-\beta_{in}}$$

$$P_{in} := \prod_{v \in V} \frac{(in(v))^{-\beta_{in}}}{Z_{in}}$$

$$\Pr(G) = P_{in} P_{out} \frac{1}{m!} \prod_{v \in V} in(v)!out(v)!$$

PA Algorithm

$$\Pr(G \mid \pi) = \prod_{i=2}^{n} d_\pi(i)!\, r \prod_{\substack{j<i: \\ (\pi_j,\pi_i)\in E}} p \frac{in_i^\pi(\pi_j)+\gamma}{m_i^\pi} \prod_{\substack{j<i: \\ (\pi_j,\pi_i)\in E}} q \frac{out_i^\pi(\pi_j)+\gamma}{m_i^\pi}$$

$$in_i^\pi(v) = \left|\left\{ j : j < i \wedge (\pi_j, v) \in E \right\}\right|$$

$$out_i^\pi(v) = \left|\left\{ j : j < i \wedge (v, \pi_j) \in E \right\}\right|$$

$$m_i^\pi = \sum_{k=1}^{i-1} (in_i^\pi(\pi_k)+\gamma) = \sum_{k=1}^{i-1} (out_i^\pi(\pi_k)+\gamma)$$

$$d_\pi(i) = in_{i+1}^\pi(v) + out_{i+1}^\pi(v) \qquad r = 1 - p - q$$

PA Algorithm

$$\Pr(\pi \mid G) = \frac{\Pr(\pi, G)}{\Pr(G)} = \frac{\Pr(G \mid \pi)\Pr(\pi)}{\Pr(G)}$$

$$\Pr(G) = \frac{\Pr(G \mid \pi)\Pr(\pi)}{\Pr(\pi \mid G)} = \frac{\Pr(G \mid \pi)}{n!\prod_{i=1}^{n} \Pr(\pi_i \mid G, \pi_1, \pi_2, \ldots, \pi_{i-1})}$$

$\mathrm{PA}(G, p, q, \gamma, T)$

- Let σ be the permutation of vertices sorted by the total degree $\mathrm{in}(v) + \mathrm{out}(v)$ in decreasing order.
- return $\dfrac{\Pr(G|\sigma)}{n!\prod_{i=1}^{n} \mathrm{Self-iter}(G, \sigma, i, p, q, \gamma, T)}$.

$\text{Self-iter}(G, \sigma, i, p, q, \gamma, T)$ // *estimates* $\Pr(\sigma_i | G, \sigma_1 \ldots \sigma_{i-1})$.

- $\pi := \sigma$
- est := 0
- for t=1 to T
  - $v :=$ random vertex from $\{\sigma_i, \sigma_{i+1}, \ldots, \sigma_n\}$
  - $a[j] := \Pr(G|\mathrm{Shift}(\pi, v, j))$ for $j \in \{i, i+1, \ldots, n\}$
  - choose j at random from $\{i, \ldots, n\}$ with probability $a[j]/\sum_{k=i}^{n} a[k]$.
  - $\pi := \mathrm{Shift}(\pi, v, j)$ // *random walk step*
  - $b[j] := \Pr(G|\mathrm{Shift}(\pi, \sigma_i, j))$ for $j \in \{i, i+1, \ldots, n\}$
  - est := est + $b[i]/\sum_{k=1}^{n} b[k]$ // *estimation step*
- return $\frac{\mathrm{est}}{T}$.

$\mathrm{Shift}(\pi, v, j)$ is the permutation obtained from π by inserting v at position j.

SW Algorithm

- If we know the initial grid alignment

$$\Pr(G \mid g) = \prod_{(u,v)\in E} \alpha\, dist(u,v)^{-\beta} \prod_{(u,v)\notin E} [1 - \alpha\, dist(u,v)^{-\beta}]$$

$$dist(u,v) = \left|u_x^g - v_x^g\right| + \left|u_y^g - v_y^g\right|$$

$$\Pr(G) = \frac{1}{n!}\sum_g \Pr(G \mid g) \le \max_g \Pr(G \mid g)$$

SW($G, \alpha, \beta, T_0, R, \delta$) // *Finds* $\max_g \Pr(G|g)$.

1) $T := T_0$ // *temperature*
2) $g :=$ random initial assignment to the grid.
3) repeat: // *until convergence of* $\Pr(G|g)$.
   a) r:=0 // *step counter*
   b) Let u and v be two vertices on grid g. Let g' be the grid g with u and v swapped.
   c) with probability $\max\left\{1, \left(\frac{\Pr(G|g')}{\Pr(G|g)}\right)^{1/T}\right\}$, do:
      - g:=g'
      - $r := r + 1$
      - if $r > R$, then $r := 0$; $T := T(1 - \delta)$

Fig. 3. SW model computation

Experiments

- Snapshots of the AS-level Internet topology
  - 1997: 3,117 vertices 6,024 edges
  - 1999: 6,266 vertices 13,681 edges
  - 2001: 11,080 vertices 25,485 edges

Log-likelihoods

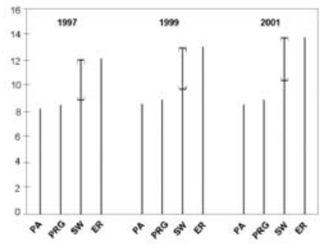


Figure 4. The log-likelihoods (per edge) of the data sets using the four models. The SW model has an uncertainty region, but we can still rank it with respect to all the others.

Parameter

The optimal parameters for the power-law models have not changed significantly over time.

1997:

PA	PRG	SW	ER
8.30	**8.60**	**8.96**	**12.10**
$p = 0.58$	$\beta^{in} = 1.55$	$\alpha = 0.111$	$p = 6.2e\text{-}4$
$q = 0.08$	$\beta^{out} = 2.39$	$\beta = 1.9$	
$\gamma = 0.5$	$c^{in} = 610$		
	$c^{out} = 69$		

1999:

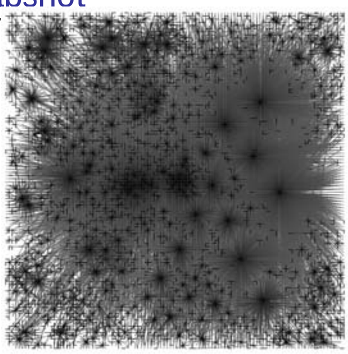
PA	PRG	SW	ER
8.55	**8.83**	**9.76**	**12.93**
$p = 0.61$	$\beta^{in} = 1.57$	$\alpha = 0.092$	$p = 3.5e\text{-}4$
$q = 0.08$	$\beta^{out} = 2.44$	$\beta = 1.8$	
$\gamma = 0.4$	$c^{in} = 1410$		
	$c^{out} = 172$		

2001:

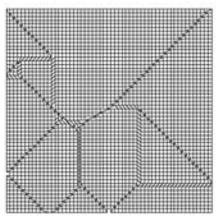
PA	PRG	SW	ER
8.58	**8.85**	**10.42**	**13.68**
$p = 0.63$	$\beta^{in} = 1.57$	$\alpha = 0.088$	$p = 2.1e\text{-}4$
$q = 0.07$	$\beta^{out} = 2.5$	$\beta = 1.8$	
$\gamma = 0.3$	$c^{in} = 2421$		
	$c^{out} = 214$		

Figure 5. Raw data results of the four algorithms on datasets (not including error bars). Bold numbers represent log-likelihood per edge, the rest are the best parameter values. Note: SW only computes a lower bound on the log-likelihood.

The results of the simulated annealing on the 2001 Internet snapshot



The results of simulated annealing on a grid graph



Thank you