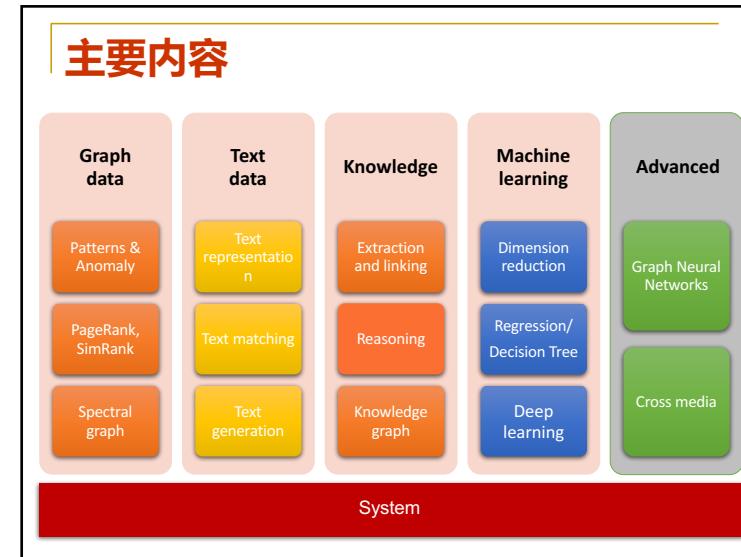


大数据分析

Summary

程学旗
靳小龙
刘盛华



Outline

- 大数据
- 大规模机器学习
- 大图挖掘，谱图方法与理论
- 文本数据分析
- 知识工程与知识图谱
- 知识获取与知识计算
- 链接分析
- 大规模数据计算系统



大数据的认知误区

大数据 ≠ 数据中心

数据中心 (IDC) 是对互联网业务资源进行集中式处理和分发的物理环境。在大数据产业的传输层，是大数据应用的网络基础设施

大数据 ≠ 云计算

云计算是互联网业务的系统平台，实现海量数据的高效存储和利用。在大数据产业的物理层，是大数据应用的系统基础设施

大数据 ≠ 数字化信息

数字化信息是大数据的组成部分，但不是所有的数字化信息都能产生大数据。大数据是数字化信息被生产、消费的过程的记录

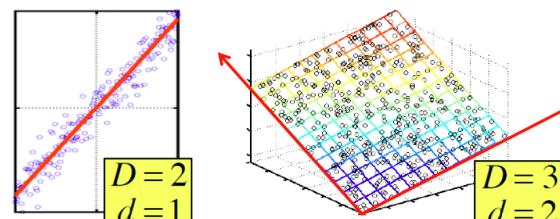
大数据 ≠ 海量数据

海量是大数据的特征之一，但如前所述，大数据并不简单地指海量的数据，而是具有nV特性的海量数据

Outline

- 大数据
- 大规模机器学习
- 大图挖掘，谱图方法与理论
- 文本数据分析
- 知识工程与知识图谱
- 知识获取与知识计算
- 链接分析
- 大规模数据计算系统

Dimensionality Reduction

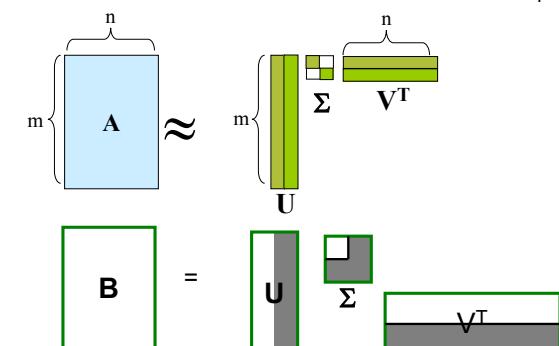


- Assumption: Data lies on or near a low d -dimensional subspace
- Axes of this subspace are effective representation of the data

7

SVD

$$\mathbf{A} \approx \mathbf{U}\Sigma\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$



8

Least squares regression

Sketching :

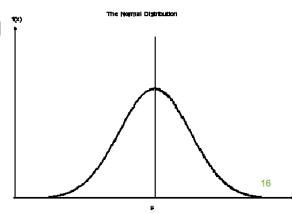
- How to find an approximate solution x to $\min_x \|Ax-b\|_2$?
- **Goal:** output x' for which $\|Ax'-b\|_2 \leq (1+\epsilon) \min_x \|Ax-b\|_2$ with high probability
- Draw S from a $k \times n$ random family of matrices, for a value $k \ll n$
- Compute S^*A and S^*b
- Output the solution x' to $\min_x \|(SA)x-(Sb)\|_2$
 - $x' = (SA)^{-1}Sb$

Sketching matrix S ?

- Recall: output the solution x' to $\min_x \|(SA)x-(Sb)\|_2$
- Lots of matrices work
- S is $d/\epsilon^2 \times n$ matrix of i.i.d. Normal random variables

S is a subspace embedding

For all x , $\|SAx\|_2 = (1 \pm \epsilon) \|Ax\|_2$



ref: David P. Woodruff, Sketching as a Tool for Numerical Linear Algebra, Foundations and Trends in Theoretical Computer Science, vol 10, issue 1-2, pp. 1-157 (ref to 10-40)

Faster Subspace Embeddings S

- CountSketch matrix
- Define $k \times n$ matrix S , for $k = O(d^2/\epsilon^2)$
- S is really sparse: single randomly chosen non-zero entry per column

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

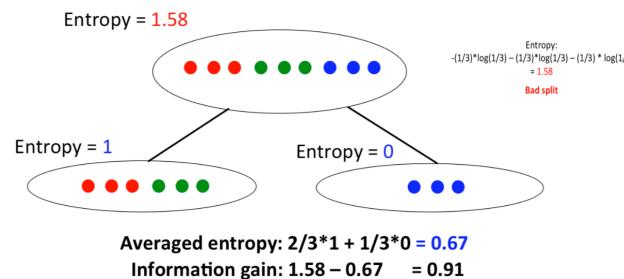
$\text{nnz}(A)$ is number of non-zero entries of A

Can compute
 $S \cdot A$ in $\text{nnz}(A) \ll nd < nd^2$
time!

Decision Tree: Splitting the node

- Classification tree: Split the node to maximize entropy
 - Let S be set of data points in a node, $c = 1, \dots, C$ are labels:
- $$\text{Entropy : } H(S) = - \sum_{c=1}^C p(c) \log p(c),$$
- where $p(c)$ is the proportion of the data belong to class c .
 - Entropy=0 if all samples are in the same class
 - Entropy is large if $p(1) = \dots = p(C)$

Information Gain



Random Forest

- **Random Forest (Bootstrap ensemble for decision trees):**
 - Create T trees
 - Learn each tree using a subsampled dataset S_i and subsampled feature set D_i
 - Prediction: Average the results from all the T trees
- **Benefit:**
 - Avoid over-fitting
 - Improve stability and accuracy
- **Good software available:**
 - R: "randomForest" package Python: sklearn

Gradient Boosted Decision Tree (GBDT)

- Minimize loss $\ell(y, F(x))$ with $F(\cdot)$ being ensemble trees

$$F^* = \underset{F}{\operatorname{argmin}} \sum_{i=1}^n \ell(y_i, F(x_i)) \quad \text{with} \quad F(x) = \sum_{m=1}^T f_m(x)$$

(each f_m is a decision tree)

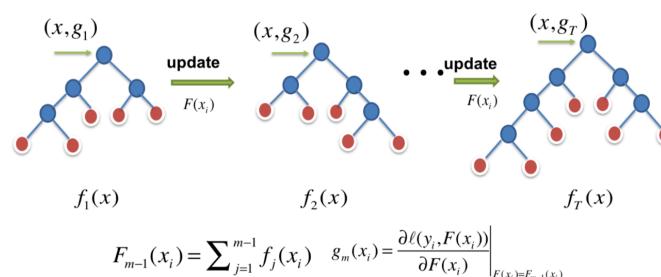
- Finding $f_m(x, \theta_m)$ by minimizing the loss function:

$$\underset{f_m}{\operatorname{argmin}} \sum_{i=1}^N [f_m(x_i, \theta) - g_i/h_i]^2 + R(f_m)$$

Gradient Boosted Decision Tree (GBDT)

- **Key idea:**

- Each base learner is a decision tree
- Each regression tree approximates the functional gradient $\frac{\partial \ell}{\partial F}$



Deep Learning

Multilayer Neural Net

- Consider a network with L hidden layers.

- layer pre-activation for $k > 0$

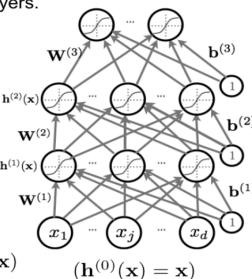
$$\mathbf{a}^{(k)}(\mathbf{x}) = \mathbf{b}^{(k)} + \mathbf{W}^{(k)}\mathbf{h}^{(k-1)}(\mathbf{x})$$

- hidden layer activation from 1 to L:

$$\mathbf{h}^{(k)}(\mathbf{x}) = g(\mathbf{a}^{(k)}(\mathbf{x}))$$

- output layer activation ($k=L+1$):

$$\mathbf{h}^{(L+1)}(\mathbf{x}) = o(\mathbf{a}^{(L+1)}(\mathbf{x})) = f(\mathbf{x})$$



Outline

- 大数据
- 大规模机器学习
- 大图挖掘，谱图方法与理论
- 文本数据分析
- 知识工程与知识图谱
- 知识获取与知识计算
- 链接分析
- 大规模数据计算系统

Training a deep neural networks

Model selection

- training, validation sets

Early stopping

防止过拟合



Mini-batch, Momentum

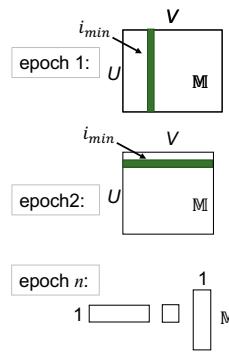
考虑历史梯度

Patterns

- S.1 degree distributions
- S.2: Eigenvalue Power Law
- S.3: Triangle ‘Law’
- S.4: Conn. components
- T.1 Small and Shrinking Diameter
- T.2 Densification Power Law

A near-linear heuristic algorithm to detect dense block

- Given: adjacency matrix M
- $X \leftarrow \{U, V\}$
- While X is not empty
 - $i_{min} \leftarrow \arg \min_{i \in X} \deg(i, X)$
 - $X \leftarrow X \setminus \{i_{min}\}$
 - Keep X_{best} that has the best arithmetic avg degree $g(X_{best})$
- Return X_{best}
- Theoretical boundary: $g(X_{best}) > \frac{1}{2}g(X_{opt})$
- Time complexity with Priority Tree: $O(|E| \log(|V| + |U|))$
- Optimal algorithm needs $O(|V|^2 \log^2 |V|)$



[A.V. Goldberg, Technical report, 1984]
[Asahiro et al, SWAT'96] [M Charikar, 2000] [B Hooi et al, KDD'16]

21

Theoretical Bounds

- The result will have a lower bound:

THEOREM 2. Let \mathcal{A}, \mathcal{B} be the set of users and objects returned by FRAUDAR. Then:

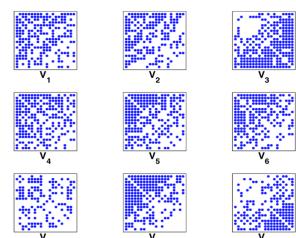
$$g(\mathcal{A} \cup \mathcal{B}) \geq \frac{1}{2}g_{opt}$$

where g_{opt} is the maximum value of g , i.e.

$$g_{opt} = \max_{\mathcal{A}', \mathcal{B}'} g(\mathcal{A}' \cup \mathcal{B}')$$

EigenSpoke: Spectral-based method

- Find **dense** groups of users by eigenvectors
 - 20 nodes with the highest magnitude projection along the first 9 singular vectors



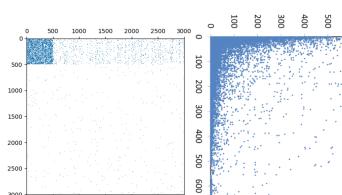
inducing sub-graphs contain **near-cliques**.

Prakash, B.A., Sridharan, A., Seshadri, M., Machiraju, S., Faloutsos, C. In PAKDD. Springer (2010)

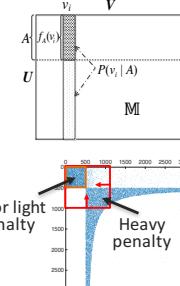
23

HoloScope: contrast suspiciousness

- Avoid hyperbolic community



Our solution:

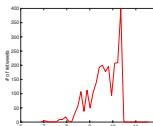
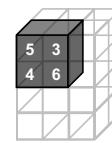


No or light penalty
Heavy penalty

Graph with temporal information

- D-cube: modeling as tensors
- HoloScope: capture the spike and drops in time series
- EigenPulse: sketching matrix A, and incremental SVD

$$l \begin{matrix} m \\ Q \\ \text{sketchin} \\ g \end{matrix} \cdot m \begin{matrix} n \\ A \end{matrix} = l \begin{matrix} n \\ B \end{matrix}$$



Spectral Graph Theory

- Connections between
 - combinatorial properties of graphs
 - the eigenvalues and eigenvectors of their associated matrices

Laplacian Matrix

- $L_G = D - A = \sum_e L_e$
- $x^T L_G x = \sum_{(u,v) \in E} w_{u,v} (x(u) - x(v))^2$
- $\lambda_2 > 0$. if and only if graph is connected

minimum cut

$$x(a) = \begin{cases} 1 & a \text{ in } S \\ 0 & a \text{ not in } S \end{cases} \quad \sum_{(a,b) \in E} (x(a) - x(b))^2$$

通过特征向量求cut

Outline

- 大数据
- 大规模机器学习
- 大图挖掘，谱图方法与理论
- 文本数据分析
- 知识工程与知识图谱
- 知识获取与知识计算
- 谱图方法与理论
- 链接分析
- 大规模数据计算系统

单词的表示方法

- 要将自然语言的问题转化为计算机可以处理的问题，首先要找到可以将文本符号数字化方法。文本表达的结果直接影响整个机器学习系统的性能
- 单词作为语言的基本单元，其表示学习也一直是文本处理领域的核心问题
- 常用的表示方法可以分为局域性表示和分布式表示两种

单词的表示方法

- 局域性表示
 - 独热表示
- 分布式表示
 - 横向组合关系
 - 隐性语义索引(Latent Semantic Indexing , LSI)
 - 概率隐性语义索引(Probabilistic Latent Semantic Indexing , PLSI)
 - 隐性狄利克雷分析(Latent Dirichlet Allocation , LDA)
 - 纵向聚合关系
 - 神经网络概率语言模型(Neural Prob. Language Model, NPLM)
 - 排序学习模型(C&W)
 - 上下文预测模型(Word2Vec)
 - 全局上下文模型(GloVe)

29

单词的分布式表示

- 横向组合关系指两个单词同时出现在一段文本区域中，强调它们可以进行组合，在句子中往往起到不同的语法作用，如下图中“国科大”和“大学”即存在横向组合关系。对横向组合关系建模的模型通常使用文档作为上下文
- 纵向聚合关系指的是纵向的可替换的关系，如图中的“国科大”和“清华”。纵向聚合关系通常使用当前单词周边的单词作为其上下文



30

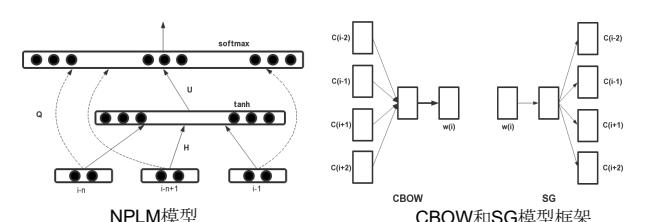
横向组合关系

- 常用的横向组合关系有：
 - 隐性语义索引(Latent Semantic Indexing , LSI)
 - 概率隐性语义索引(Probabilistic Latent Semantic Indexing , PLSI)
 - 隐性狄利克雷分析(Latent Dirichlet Allocation , LDA)
 - 均属于矩阵分解模型

31

纵向聚合关系

- 常用的纵向聚合算法有
 - 神经网络概率语言模型(Neural Probabilistic Language Model, NPLM)
 - 排序学习模型(C&W)
 - 上下文预测模型(Word2Vec)
 - 全局上下文模型(GloVe)等



32

句子的表示方法

■ 传统方法

- 词集模型
- 词袋模型
- TF-IDF表示 **既有局部又有全局**

■ 分布式表示方法

- 主题模型
- 基于单词分布式表示组合的表示方法
- 由原始语料直接学习的表示方法

33

词集模型

■ 词集模型(**Set of Words**)是一个由单词构成的集合，忽略文本的词序与语法，只记录单词是否出现的情况。按照集合的定义，集合中的每个元素只有1个，因此词集中的每个单词都只有一个

■ 词集模型是最简单的**句子表示方法**，因为其数值非0即1，可以很好地支持位运算，在检索应用场景中能够执行快速的查询处理

■ 示例：

- 句子1：“我 来自 中国 科学院 大学”
- 句子2：“他 在 中国 科学院 计算所 学习”
- 单词表vocab={(我:0, 来自:1, 中国:2, 科学院:3, 大学:4, 他:5, 在:6, 计算所:7, 学习:8)}
- 句子1的词集模型向量表示：(1, 1, 1, 1, 0, 0, 0, 0)
- 句子2的词集模型向量表示：(0, 0, 1, 1, 0, 1, 1, 1)

34

词袋模型

■ 词袋模型(**Bag of Words**)是在词集模型的基础上，考虑了**单词出现的次数**，因此，在词袋模型中，句子向量中每个单词对应的位置上记录的是该单词出现的次数，这也体现了各个单词在该句子中的重要程度

■ 示例：

- 句子：“我 来自 中国 科学院 大学，他 在 中国 科学院 计算所 学习”
- 单词表vocab={(我:0, 来自:1, 中国:2, 科学院:3, 大学:4, 他:5, 在:6, 计算所:7, 学习:8)}
- 词袋模型向量表示：(1, 1, 2, 2, 1, 1, 1, 1)

35

基于单词分布式表示组合的表示方法

■ 句子的分布式表示建立在单词的分布式表示的基础之上。其主要思想是：针对具体任务，对单词的分布式表示进行组合或选择等，最终得到一个向量作为句子的分布式表示，这是一个特征组合、提取的过程。

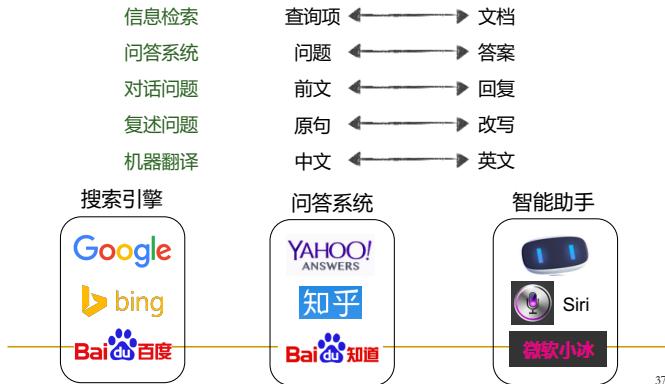
■ 常用的方法

- 基于**卷积神经网络(CNN)**的分布式表示
- 基于**循环神经网络(RNN)**的分布式表示
- 基于**递归神经网络(RecNN)**的分布式表示
- 基于**DAN(Deep Averaging Networks)**的分布式表示

36

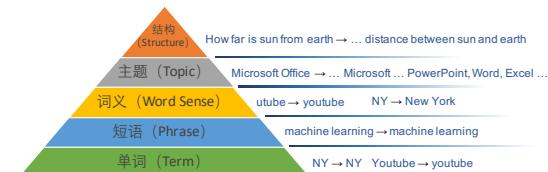
文本中的匹配问题

文本匹配是自然语言理解的一个核心问题，许多文本处理的问题可以抽象成**文本匹配**的问题



37

文本匹配的挑战



挑战：

- 词语匹配的多元性
- 短语匹配的结构性
- 文本匹配的层次性

荷花 = 芙蓉 苹果 = 公司 or 水果

机器学习 --- 学习机器

词 - 短语 - 句子 - 段落 - 篇章

38

文本匹配方法与评价

- 基于规则的文本匹配
 - 启发式规则
 - 隐语义表达
- 基于学习的文本匹配
 - 人工特征融合
 - 表达学习
- 文本匹配的评价方法

39

文本匹配的评价方法

- 分类准确率 (Accuracy)**：用于评价分类任务的指标，对于文本匹配任务，只有两类标签，匹配为1，不匹配为0。因此可以把文本匹配看作是一个二分类问题。使用分类准确率可以方便的评价模型对每一对文本的分类是否正确。分类正确的数量占总测试样本数量的比例就是分类准确率
- P@k (Precision at k)**：表示前k个文档的排序准确率。假定预测结果排序后前k个文档中相关文档的数量为 Y_k ，那么P@k可以定义为：
$$P@k = \frac{Y_k}{k}$$
- R@k (Recall at k)**：表示前k个文档的排序召回率。按照标注的相关度排序后前k个文档中相关文档的数量为 G_k ，那么可定义R@k为：
$$R@k = \frac{G_k}{k}$$

40

文本匹配的评价方法

- MAP (Mean Average Precision) : 该指标综合考虑了所有相关文档的排序状况。将所有相关文档在预测结果排序中的位置定义为 r_1, r_2, \dots, r_G , 则平均精度均值指标可定义为 :

$$\text{MAP} = \frac{\sum_{i=1}^G P@r_i}{G}$$

- MRR (Mean Reciprocal Rank) : 如果只考虑在预测结果排序中第一个出现的相关文档的位置 r_1 , 可以定义MRR指标为 :

$$\text{MRR} = P@r_1 = \frac{1}{r_1}$$

41

文本匹配的评价方法

- nDCG (normalized Discounted Cumulative Gain) 归一化折扣累计收益
 - 有些任务当中标注的相关度本身就有大小之分而不是单纯的匹配和不匹配两个级别, 这个时候nDCG这个指标就会更加有效。nDCG让相关度越高的排在越前面
 - 给定按照标注的文档相关度排序后的文档相关度值分别为 $\hat{rel}_1, \hat{rel}_2, \dots, \hat{rel}_N$, 若按照预测结果排序后的文档相关度的值分别为 $rel_1, rel_2, \dots, rel_N$ 。所以nDCG指标的定义如下 :

$$IDCG = \hat{rel}_1 + \sum_{i=2}^n \frac{\hat{rel}_i}{\log_2 i}$$

$$DCG = rel_1 + \sum_{i=2}^n \frac{rel_i}{\log_2 i}$$

$$nDCG = \frac{DCG}{IDCG}$$

42

Outline

- 大数据
- 大规模机器学习
- 大图挖掘, 谱图方法与理论
- 文本数据分析
- 知识工程与知识图谱
- 知识获取与知识计算
- 链接分析
- 大规模数据计算系统

43

通用知识图谱

- Google所提出的知识图谱是面向全领域的通用知识图谱
- 通用知识图谱主要应用于面向互联网的搜索、推荐、问答等业务场景
- 通用知识图谱强调的是知识的广度, 因而关注更多的是实体, 很难生成完整的、全局性的本体层, 也很难统一管理

44

行业知识图谱

- 行业知识图谱指面向**特定领域**的知识图谱
- 目标用户对象需要考虑行业中各种不同级别的人员，而不同人员的业务场景不同，因而**需要一定的深度与完备性**
- 行业知识图谱**对准确度要求非常高**，通常用于辅助各种**复杂的分析应用或决策支持**
- **有严格且丰富的数据模式**，行业知识图谱中的**实体通常属性比较多且具有行业意义**

45

行业知识图谱数据的特点

- **数据来源多**：内部数据、互联网数据、第三方数据
- **数据类型多**：包含结构化、半结构化、非结构化数据，且后两者越来越多
- **数据模式无法预先确定**：模式在数据出现之后才能确定；数据模式随数据增长不断演变
- **数据量大**：在大数据背景下，行业应用数据的体量通常都以亿级别计算，存在通常在TB、PB级别甚至更多

46

通用知识图谱 VS 行业知识图谱



- ✓ 面向通用领域
- ✓ 以常识性知识为主
- ✓ 结构化的百科知识
- ✓ 强调知识的广度
- ✓ 使用者是普通用户

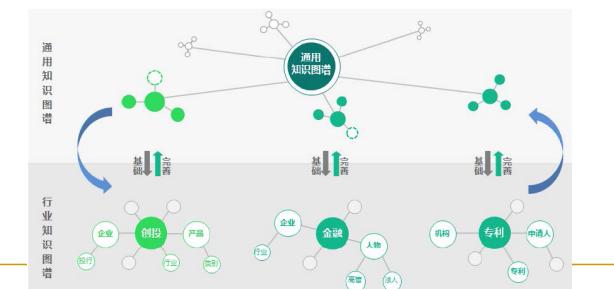


- ✓ 面向某一特定领域
- ✓ 基于行业数据构建
- ✓ 基于语义技术的行业知识库
- ✓ 强调知识的深度
- ✓ 潜在使用者是行业人

47

通用知识图谱 + 行业知识图谱

- 通用知识图谱的**广度**，行业知识图谱的**深度**，**相互补充**，形成更加完善的知识图谱
- 通用知识图谱中的知识，可以作为行业知识图谱构建的基础；而构建的行业知识图谱，再融合到通用知识图谱中



48

Outline

- 大数据
- 大规模机器学习
- 大图挖掘，谱图方法与理论
- 文本数据分析
- 知识工程与知识图谱
- 知识获取与知识计算
- 链接分析
- 大规模数据计算系统

实体抽取

■ 实体抽取定义

- 从原始语料中自动识别出**指定类型的命名实体**，主要包括**实体名**（如人名、地名、机构名、国家名等）、**缩略词**，以及一些**数学表达式**（如货币值、百分数、时间表达式等）

■ 示例

5月19日**下午**，**史密斯**教授做客**北京大学**海外名师讲堂。
时间 人名 机构名

50

基于机器学习的方法

■ 序列标注

- 实体标注一般使用**BIO模式**

(B-begin, I-inside, O-outside)

输入序列	小明	昨天	晚上	在	公园	遇到	了	小红	.
语块	B-NP	B-NP	I-NP	B-PP	B-NP	B-VP		B-NP	
标注序列	B-Agent	B-Time	I-Time	O	B-Location	B-Predicate	O	B-Patient	O
角色	Agent	Time	Time		Location	Predicate	O	Patient	

- 还有**BIOES标注模式**

(B-begin, I-inside, O-outside, E-end, S-single)

51

基于机器学习的方法

■ 隐马尔科夫模型

- 假定分词后的文档词语序列为 $W = (w_1, \dots, w_n)$ ， $T = (t_1, \dots, t_n)$ 为词序列的实体标注结果。模型旨在给定词语序列 W 的情况下，找出现概率最大的标注序列 T ，即，求使 $P(T|W)$ 最大的标注序列

$$T_{max} = arg_{T} max P(T|W)$$

根据贝叶斯公式，

$$P(T|W) = P(T)P(W|T)/P(W)$$

其中， $P(W)$ 可以看成一个常数，则有

$$T_{max} = arg_{T} max P(T)P(W|T)$$

其中， $P(T)P(W|T)$ 是引入隐马尔科夫模型来计算的参数。如果穷举序列 W 和 T 的所有可能情况，这个问题是NP难的。

52

基于机器学习的方法

■ 隐马尔科夫模型

- 按照马尔科夫假设，当前状态 t_i 只和其前一状态 t_{i-1} 有关，因此有

$$P(T)P(W|T) \approx \prod_{i=1}^n P(w_i|t_i)P(t_i|t_{i-1})$$

其中， $P(w_i|t_i)$ 表示隐状态为 t_i 的词语集合中出现 w_i 的概率， $P(t_i|t_{i-1})$ 表示上一词语标注为 t_{i-1} 时，当前词语标注为 t_i 的转移概率。进一步

$$T_{max} = arg_T max \prod_{i=1}^n P(w_i|t_i)P(t_i|t_{i-1})$$

$$T_{max} = -arg_T max \sum_{i=0}^n \{lnP(w_i|t_i) + lnP(t_i|t_{i-1})\}$$

训练时，取 $P(w_i|t_i) \approx Count(w_i, t_i)/Count(t_i)$ ，其中 $Count(w_i, t_i)$ 表示词语 w_i 被标注为 t_i 的次数， $Count(t_i)$ 表示隐状态 t_i 出现的总次数

53

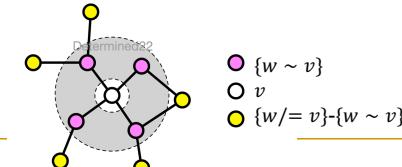
基于机器学习的方法

■ 条件随机场(Conditional Random Field, CRF)

- 设 $X = (X_1, \dots, X_n)$ 与 $Y = (Y_1, \dots, Y_n)$ 是联合随机变量。若在给定随机变量 X 的条件下，随机变量 Y 构成一个由无向图 $G = (V, E)$ 表示的马尔科夫模型，则条件概率分布 $P(Y|X)$ 称为条件随机场，即：

$$P(Y_v|X, Y_w, w/v) = P(Y_v|X, Y_w, w \sim v)$$

其中， w/v 表示图 $G = (V, E)$ 中节点 v 以外的所有节点， $w \sim v$ 表示与节点 v 有连边的所有节点



54

基于机器学习的方法

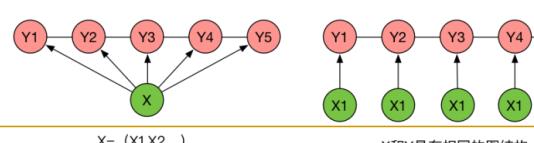
■ 线性链条件随机场(Linear Chain CRF)

- 设 $X = (X_1, \dots, X_n)$ 与 $Y = (Y_1, \dots, Y_n)$ 均为线性链表示的随机变量序列。若在给定的随机变量序列 X 的条件下，随机变量序列 Y 的条件概率分布 $P(Y|X)$ 构成条件随机场，且满足马尔科夫性，即：

$$P(Y_i|X, Y_1, \dots, Y_n) = P(Y_i|X, Y_{i-1}, Y_{i+1})$$

则称 $P(Y|X)$ 为线性链的条件随机场

- 线性链CRF不仅考虑了上一个状态 Y_{i-1} ，还考虑了后续的状态 Y_{i+1}



55

基于机器学习的方法

■ 线性链条件随机场模型

- 与隐马尔科夫模型相同，将CRF用于命名实体识别，其目标也是求 $T_{max} = arg_T max P(T|W)$ ，但是这里

$$P(T|W) = \frac{1}{Z(W)} exp \left(\sum_{i,k} \lambda_k \psi_k(t_{i-1}, t_i, W, i) + \sum_{i,l} \delta_l \phi_l(t_i, W, i) \right)$$

$$Z(W) = \sum_t exp \left(\sum_{i,k} \lambda_k \psi_k(t_{i-1}, t_i, W, i) + \sum_{i,l} \delta_l \phi_l(t_i, W, i) \right)$$

其中 $Z(W)$ 是归一化因子，在所有可能的输出序列上求和； λ_k 和 δ_l 为权重因子

56

基于机器学习的方法

■ 线性链条件随机场模型

$$P(T|W) = \frac{1}{Z(W)} \exp \left(\sum_{i,k} \lambda_k \psi_k(t_{i-1}, t_i, W, i) + \sum_{i,l} \delta_l \phi_l(t_i, W, i) \right)$$

□ $\psi_k(t_{i-1}, t_i, W, i)$ 是转移函数，依赖于当前和前一位置，表示从标注序列中位置*i*–1的标记*t_i*–1转移到位置*i*上的标记为*t_i*的概率

$$\varphi_k(t_{i-1}, t_i, W, i) = \begin{cases} 1, & \text{满足条件} \\ 0, & \text{其他} \end{cases}$$

□ $\phi_l(t_i, W, i)$ 是状态函数，表示标记序列在位置*i*上标记为*t_i*的概率

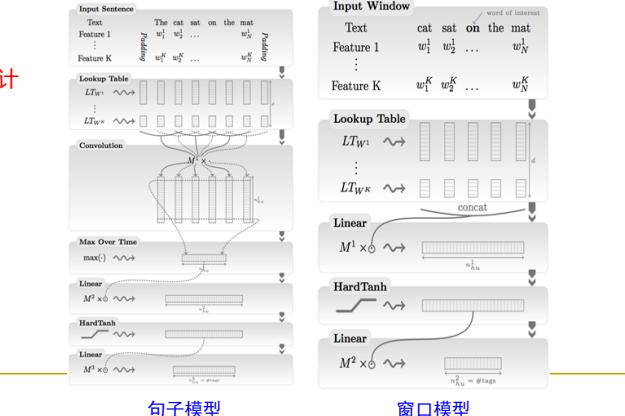
$$\phi_l(t_i, W, i) = \begin{cases} 1, & \text{满足条件} \\ 0, & \text{其他} \end{cases}$$

57

考试，会设计

基于机器学习的方法

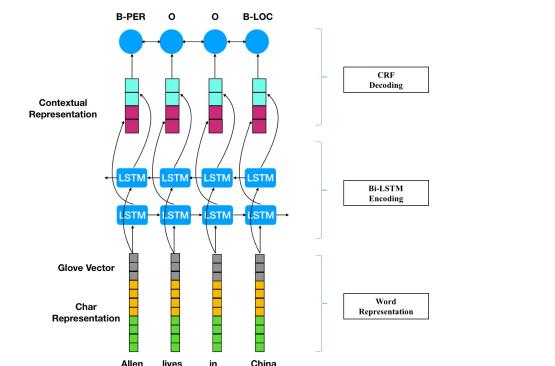
■ 基于深度学习的模型



58

基于机器学习的方法

■ 基于深度学习的模型—LSTM-CRF模型



59

关系抽取

■ 关系抽取示例



60

■ 抽取方法分类

- 有监督关系抽取
- 半监督关系抽取
- 远程监督关系抽取
- 无监督关系抽取

远程监督关系抽取

- **初始动机**: 通过外部知识库代替人对语料进行标注，从而低成本地获取大量有标注数据 [Mintz et al., 2009]
- **核心思想**: 如果知识库中存在三元组 (e_1, R, e_2) ，那么语料中所有出现实体对 (e_1, e_2) 的语句，都标注为表达了关系R
- 根据这一假设，对每个三元组 (e_1, R, e_2) ，将所有 (e_1, e_2) 共现的句子都标注标签R，用**分类方法**解决关系抽取问题

61

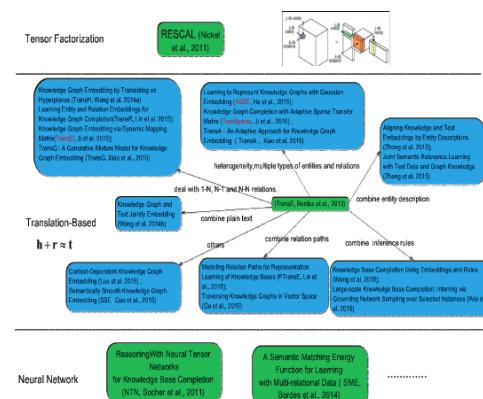
远程监督关系抽取

- Riedel等[Riedel et al., 2010]认为Mintz的假设过强，可能引入噪声模式，因而提出“at-least-once”假设：
 - 如果存在三元组 (e_1, R, e_2) ，那么所有 (e_1, e_2) 实体对共现的语句中，至少有一句体现了关系R在这两个实体上成立的事实
- 引入了**多实例学习机制**，将所有 (e_1, e_2) 共现的句子聚成一个句袋，并将任务由对**句子分类**变为对**句袋分类**

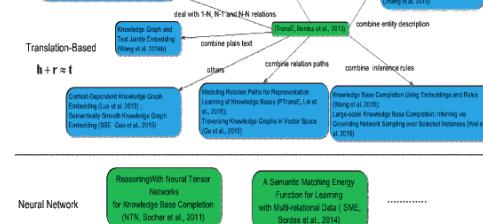
62

基于分布式表达的知识计算

张量分解方法



基于翻译的方法



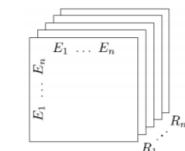
神经网络方法



63

用张量表示知识图谱

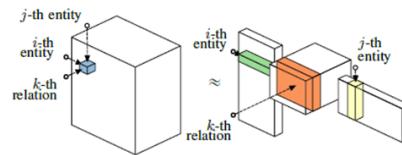
知识图谱中三元组的结构是（头部实体 h ，关系 r ，尾部实体 t ），其中 r 连接头尾实体。以 E_1, E_2, \dots, E_n 表示知识图谱中的实体，以 R_1, R_2, \dots, R_m 表示知识图谱中的关系，则可以使用一个**三维矩阵**（张量）表示知识图谱



Nickel et al. (2011). A three-way model for collective learning on multi-relational data. In Proceedings of the 28th international conference on machine learning (ICML-11).

64

张量分解得到实体、关系表示



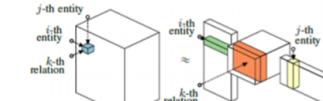
65

张量分解的目标函数

- 表示知识图谱的张量记为 Y ，其第 k 个矩阵记为 Y_k ，是第 k 种关系的矩阵，表示该种关系在向量空间中与头尾部实体相互作用
- 对 Y_k 可以进行如下的低秩分解：

$$Y_k = AR_kA^T \quad k = 1, 2, \dots, m$$

其中， $A \in R^{n \times r}$, $Y_k \in R^{n \times n}$, $R_k \in R^{r \times r}$, r 表示矩阵 A 的秩； A 是实体向量矩阵，每一行表示一个实体的向量，转置后其每一列表示一个实体的向量



66

张量分解的目标函数

- 由上述内容可知， A 和 R_k 均是待求解的变量。因此目标函数是：

$$\min_{A, R_k} (f(A, R_k) + g(A, R_k))$$

其中 $f(A, R_k)$ 是目标函数

$$f(A, R_k) = \frac{1}{2} \left(\sum_k \|Y_k - AR_kA^T\|_F^2 \right)$$

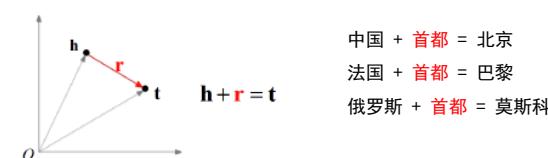
$g(A, R_k)$ 是正则化项

$$g(A, R_k) = \frac{1}{2} \gamma \left(\|A\|_F^2 + \sum_k \|R_k\|_F^2 \right)$$

67

基于翻译的模型：TransE

- 关系事实 $(\text{head}, \text{relation}, \text{tail})$ 简写为 (h, r, t) ，其对应的向量表示为 $(\mathbf{h}, \mathbf{r}, \mathbf{t})$



Bordes, et al. Translating embeddings for modeling multi-relational data. In Advances in Neural Information Processing Systems, 2013 (pp. 2787-2795).

68

翻译模型的学习

势能函数

- 对于真实事实的三元组 (h, r, t) ，要求 $\mathbf{h} + \mathbf{r} = \mathbf{t}$ ；而对于错误的三元组则不满足该条件

$$f(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2$$

$f(\text{姚明, 出生于, 北京}) > f(\text{姚明, 出生于, 上海})$

损失函数

$$L = \sum_{(h, r, t) \in \Delta} \sum_{(h', r', t') \in \Delta'} \max(0, f_r(h, t) + M_{opt} - f_r(h', t'))$$

正例三元组集 负例三元组集 最优Margin超参

69

张量分解的目标函数

将目标函数写成分量形式

$$f(\mathbf{A}, \mathbf{R}_k) = \frac{1}{2} \left(\sum_k \|Y_k - \mathbf{A} \mathbf{R}_k \mathbf{A}^T\|_F^2 \right) \Rightarrow f(\mathbf{A}, \mathbf{R}_k) = \frac{1}{2} \sum_{i,j,k} (y_{ijk} - \mathbf{a}_i^T \mathbf{R}_k \mathbf{a}_j)^2$$

其中， y_{ijk} 是张量中的一个元素， \mathbf{a}_i 表示 \mathbf{A} 的第*i*行，即

$$[\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] = \mathbf{A}$$

70

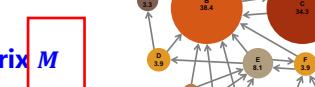
Outline

- 大数据
- 大规模机器学习
- 大图挖掘
- 文本数据分析
- 知识工程与知识图谱
- 知识获取与知识计算
- 链接分析
- 大规模数据计算系统

PageRank

Stochastic adjacency matrix M

- Let page i has d_i out-links
- If $i \rightarrow j$, then $M_{ji} = \frac{1}{d_i}$ else $M_{ji} = 0$
- M is a column stochastic matrix
- Columns sum to 1



Rank vector r : vector with an entry per page

- r_i is the importance score of page i
- $\sum_i r_i = 1$

- The flow equations can be written

$$\mathbf{r} = \mathbf{M} \cdot \mathbf{r}$$

72

Power Iteration

- Power iteration:

A method for finding dominant eigenvector (the vector corresponding to the largest eigenvalue)

- $r^{(1)} = M \cdot r^{(0)}$
- $r^{(2)} = M \cdot r^{(1)} = M(Mr^{(0)}) = M^2 \cdot r^{(0)}$
- $r^{(3)} = M \cdot r^{(2)} = M(M^2r^{(0)}) = M^3 \cdot r^{(0)}$

- Approximate dominant eigenvector:

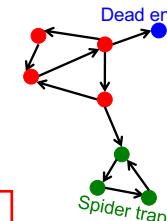
Sequence $M \cdot r^{(0)}, M^2 \cdot r^{(0)}, \dots, M^k \cdot r^{(0)}, \dots$ approaches the dominant eigenvector of M

73

Google PageRank

- Teleports

- With prob. β , follow a link at random
- With prob. $1-\beta$, jump to some random page



解决的两个问题

- Dead ends: no out-links

- not "column stochastic" (zero-column)
- teleport with probability 1.0 from dead-ends

- Spider traps: all out-links are within the group

- get stuck in a spider trap
- teleporting out of it in a finite number of steps

Outline

- 大数据
- 大规模机器学习
- 大图挖掘
- 文本数据分析
- 知识工程与知识图谱
- 知识获取与知识计算
- 谱图方法与理论
- 链接分析
- 大规模数据计算系统

MapReduce: Word Counting

Provided by the programmer

MAP:
Read input and produces a set of key-value pairs

The crew of the space shuttle Endeavor recently returned to Earth as ambassadors, harbingers of a new era of space exploration. Scientists at NASA are saying that the recent assembly of the Dextre bot is the first step in a long-term space-based man/machine partnership. "The work we're doing now - the robotics we're doing - is what we're going to need...."

Big document

Group by key:
Collect all pairs with same key

(The, 1)
(crew, 1)
(of, 1)
(the, 1)
(space, 1)
(shuttle, 1)
(Endeavor, 1)
(recently, 1)
....

(key, value)

Provided by the programmer

Reduce:
Collect all values belonging to the key and output

(crew, 2)
(space, 1)
(the, 3)
(shuttle, 1)
(recently, 1)
...

(key, value)

Sequentially read the data

76

Word Count Using MapReduce

```

map(key, value):
    // key: document name; value: text of the document
    for each word // new key - value
        emit(w, 1)

reduce(key, values):
    // key: a word; value: an iterator over counts
    result = 0
    for each count v in values:
        result += v
    emit(key, result) // new value for input key

```

77

Cost Measures for Algorithms

- In MapReduce we quantify the cost of an algorithm using
 1. **Communication cost** = total I/O of all processes
 2. **Elapsed communication cost** = max of I/O along any path
 3. (**Elapsed**) **computation cost** analogous, but count only running time of processes

Note that here the big-O notation is not the most useful
(adding more machines is always an option)

78

Example: Cost Measures

- For a map-reduce algorithm:
 - **(Total) communication cost** = input file size + 2 map写一次, reduce读一次
× (sum of the sizes of all files passed from Map processes to Reduce processes) + the sum of the output sizes of the Reduce processes.
 - **Elapsed communication cost** is the sum of the largest input + output for any map process, plus the same for any reduce process

79
题型
练习题

Good Luck!

Thank you.