

## Lec1 Review

### 智慧城市定义 (Definition)

Wikipedia 定义: 利用电子方法和传感器收集数据的高科技城市区域。

IEEE 定义: 结合技术、政府和社会。核心要素包括: 智慧经济、智慧能源、智慧交通、智慧环境、智慧生活、智慧治理。

IBM 定义: 利用技术和数据收集来提高生活质量、可持续性和城市运营效率。

香港定义 (Hong Kong Blueprint): 包含六大支柱——智慧出行 (Mobility)、智慧生活 (Living)、智慧环境 (Environment)、智慧市民 (People)、智慧政府 (Government)、智慧经济 (Economy)。

核心应用领域 (Key Applications): 智慧交通 (Smart Mobility), 智慧能源与建筑 (Smart Energy & Building), 共享经济 (Sharing Economy)

### 技术支柱 I: 云计算 (Cloud Computing):

核心特征 (Characteristics): 按需自助服务 (On Demand Self-Service)、广泛的网络接入 (Broad Network Access)、资源池化 (Resource Pooling)、快速弹性 (Rapid Elasticity)、可计量服务 (Measured Service); 服务模型 (Service Models) - 常考对比题

SaaS (软件即服务): 提供应用软件, 用户无需维护 (如 Google Docs, Salesforce)

PaaS (平台即服务): 提供开发环境和工具, 部署客户创建的应用 (如 Google App Engine)。

IaaS (基础设施即服务): 租用处理能力、存储、网络等基础资源 (如 AWS EC2, Rackspace)。

虚拟化 (Virtualization): 优势: 资源隔离、动态配置

优缺点 (Pros & Cons): 优点: 降低硬件成本、即时软件更新、无限存储、通用文档访问、设备独立性; 缺点: 依赖网络连接 (断网无法工作)、低速网络体验差、功能可能受限、数据安全与隐私隐患

技术支柱 II: 物联网 (Internet of Things, IoT): Internet of People: 连接人。Internet of Things: 连接万物 (物体之间的无线网络, 自我配置) 三个维度 (New Dimension): 任何时间 (Any TIME)、任何地点 (Any PLACE)、任何物体 (Any THING) IoT 与 Cloud 的区别 (IoT vs. Cloud) : IoT: 真实世界、小物体、受限设备、不可靠性、边缘侧; Cloud: 虚拟世界、大算力、无限能力、高可用性、中心侧。

## Lec2 Review

流程建模方法论 (Process Modeling Methodology); IDEF 家族 (IDEF Family) [重要选择/填空题]IDEF0: 建模系统功能 (Function Modeling), 涉及硬件、软件和人。IDEF1x: 建模数据关系 (Data/Information Modeling)。IDEF3: 考点核心。用于建模活动序列 (Sequence of activities)。IDEF5: 建模领域本体 (Ontologies);

IDEF3 具体元素 (IDEF3 Elements)

UOBs: 行为单元 (Activities)

逻辑连接符 (Logic Junctions): & (AND): 同时发生。O (OR): 或者 (多选)。X (Exclusive OR):

异或 (单选) 流程分析 (Process Analysis): 观察分析法 (Observational Analysis); 计算分析法 (Computational Analysis) 关联矩阵 (Incidence Matrix): 如何把有向图 (Digraph) 转化为矩阵拓扑排序算法 (Topological Sorting Algorithm): 这是本章唯一的具体算法, 极大概率考操作步骤算法步骤 (Step-by-step): 找到矩阵中非空元素只有 1 个的行; 在该行画一条水平线; 在对应的列画一条垂直线; 标记该节点为序列的下一个; 删除该行该列, 重复上述步骤, 直到所有行列被划掉系统建模 (Systems Modeling): 物理模型 (Physics-based / White-box) 黑盒模型 (Black-box / Data-driven) 数据驱动建模 (Data-Driven Modeling) 核心概念: 假设数据是

物理世界的复刻 (Replication)，通过机器学习算法从输入输出数据中找数学关系。  
**难点 (Complexity Sources):** 系统参数维度高 (High dimension)。强非线性 (Strong nonlinearity)。领域知识有限 (Limited domain knowledge)。建模前的问题 (Prerequisites): 数据可用性、特征选择、算法选择、结果验证

## Lec3 Review

数据挖掘流程 (Process): Selection (选择) Preprocessing (预处理) Transformation (变换) Data mining (挖掘) Interpretation (解释)

问题分类 (Taxonomy): 无监督学习 (Unsupervised): 没有标签 (No labeled targets)。典型算法: 聚类 (Clustering) 监督学习 (Supervised): 有标签 (Have labeled targets)。标签是分类变量 分类 (Classification)。标签是数值变量 回归 (Regression)

聚类算法 I: K-means (Partitional Clustering) : 评估指标: SSE (Sum of Squared Error) 误差平方和; 局限性 (Limitations): 难以处理: 不同大小 (Differing Sizes)、不同密度 (Differing Densities)、非球形形状 (Non-globular shapes) 的簇; 对噪声和离群点 (Outliers) 敏感; 聚类算法 II: 层次聚类 (Hierarchical Clustering): Agglomerative (凝聚): 自底向上, 从点合并成簇; Divisive (分裂): 自顶向下, 从大簇分裂; 簇间距离度量 (Linkage Criteria) - [核心考点]: MIN (Single Link / 单连接): 定义: 两个簇中最近点的距离。you 点: 能处理非椭圆形状。缺点: 对噪声敏感 (容易产生链式效应)。MAX (Complete Link / 全连接) 定义: 两个簇中最远点的距离。缺点: 倾向于把大簇打碎, 偏好球形簇。Group Average (组平均): 定义: 所有点对距离的平均值。特点: 折中方案, 对噪声较不敏感; 聚类算法 III: DBSCAN (Density-Based): 核心概念 (Point Types) - [必考定义]: 核心点 (Core point): 半径 Eps 内至少有 MinPts 个点; 边界点 (Border point): 在核心点的邻域内, 但自己不是核心点; 噪声点 (Noise point): 既不是核心也不是边界; 优点: 抗噪声, 能处理任意形状; 缺点: 处理密度变化大的数据效果不好, 高维数据效果不好。聚类有效性评估 (Cluster Validation): External (外部): 与已知标签对比 (如 Entropy); Internal (内部): 仅基于数据本身 (如 SSE); Relative (相对): 对比不同聚类结果; 内聚与分离 (Cohesion vs. Separation): Cohesion: 簇内越近越好 (SSE 越小越好); Separation: 簇间越远越好 (BSS 越大越好)

## Lec4 Re

classification: 1.TF: 分类与回归的区别: Classification (分类): 预测的是离散的 (Discrete) 或标称的 (Nominal) 类别标签 (如: Yes/No, High/Low); Regression (回归): 预测的是连续的 (Continuous) 数值; 决策树 (Decision Tree) 的特性: 不需要领域知识 (Domain knowledge) 也能构建; 不需要繁琐的数据预处理 (如归一化), 能处理高维数据; 贪心算法 (Greedy): 决策树采用自顶向下 (Top-down) 的递归分治策略, 每一步都只考虑当前最优, 不回溯; 模型评估原则: 独立性: 训练集 (Training set) 和测试集 (Test set) 的记录必须是互斥的 (Disjoint), 不能重叠; 过拟合 (Overfitting): 如果模型在训练集上表现极好, 但在测试集上表现很差, 这就是过拟合。树长得太深太复杂容易导致过拟合; 剪枝策略 (Pruning): Pre-pruning (预剪枝): 在完全生长之前停止。优点是快, 缺点是可能导致欠拟合 (Underfitting); Post-pruning (后剪枝): 先长成完全树, 再自底向上修剪。通常比预剪枝效果更好, 不易欠拟合;

**Part 2: 选择题考点 (Multiple Choice Focus): 分裂属性的选择标准 (Attribute Selection Measures): Information**

指标名称	核心公式	说明	来源
Gini Index	$1 - \sum_{j=1}^c p_j^2$	$p_j$ 是第 $j$ 类样本的占比。	
Entropy	$-\sum_{j=1}^c p_j \log_2(p_j)$	注意是负号, $\log_2$ 。	
Info Gain	$Entropy(P) - \sum \frac{N_i}{N} Entropy(i)$	父节点熵减去子节点加权熵。	
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	预测对的总数 / 总样本数。	
Error Rate	$\frac{FP+FN}{TP+TN+FP+FN}$	预测错的总数 / 总样本数。	
Precision	$\frac{TP}{TP+FP}$	查准率: 预测为正的中, 多少是真的?	
Recall	$\frac{TP}{TP+FN}$	查全率: 真实为正的中, 多少被找出来了?	
F1	$\frac{2 \cdot Precision \times Recall}{Precision + Recall}$	调和平均数。	

**Gain (信息增益):** 选择值最大 (Maximum) 的属性。偏向于取值较多 (多值) 的属性; **Gain Ratio (增益率):** 对信息增益的改进, 克服了偏向多值属性的弱点; **Gini Index (基尼指数):** 选择值最小 (Minimum) 的属性。衡量不纯度, 越小越纯; **Entropy (熵):** 值越小越好 (0 表示纯净, 1 表示最混乱); **模型评估方法的选择:** Holdout: 简单地按比例 (如 2/3 训练, 1/3 测试) 划分; Random Subsampling: 多次重复 Holdout; **Cross-validation (交叉验证):** 将数据分为  $k$  份, 轮流做测试集。 **$k$ -fold** 是最常用的方法; **Stratified Cross-validation (分层交叉验证):** 专门解决类别不平衡问题。它保证训练集和测试集中各类的比例与原始数据集一致; **评估指标的适用场景:** **Accuracy (准确率):** 仅适用于类别分布平衡的数据。如果类别极度不平衡 (如 99:1), 准确率会产生误导; **F1-measure (F1 值):** 综合了 Precision 和 Recall, 适用于类别不平衡场景; **Cost Matrix (代价矩阵):** 当将正类误判为负类的代价 (如漏诊癌症) 远高于误判为正类时使用; **关键术语 (Key Terms):** **ROC Curve:** 接收者操作特征曲线, 横轴是 FPR (False Positive Rate), 纵轴是 TPR (True Positive Rate/Recall); **AUC (Area Under Curve):** ROC 曲线下的面积, 值在 0.5 到 1 之间, 越接近 1 性能越好; 常见计算题型逻辑回归、朴素贝叶斯、决策树、随机森林等。

辑: 算 Gini: 给定一个节点有 6 个 Yes, 4 个 No;  $0.48 = \frac{1 - (0.6^2 + 0.4^2)}{1} = 1 - (0.36 + 0.16) = 1 - 0.52 = 0.48$ 。  
算 Accuracy:  
混淆矩阵中对角线 (TP+TN) 之和除以总数

Lec5

感知机 (Perceptron): 单层感知机 (Single layer perceptron) 只能解决线性可分 (linearly separable) 的问题; 单层感知机无法解决 XOR (异或) 问题, 因为 XOR 是非线性的; 神经网络 (ANN) 的特性: 多层神经网络是通用逼近器 (universal approximators), 理论上可以逼近任何函数; 如果网络太大 (节点太多), 很容易出现过拟合 (overfitting) 现象; 梯度下降算法 (Gradient Descent) 不保证收敛到全局最优解, 可能会陷入局部极小值 (local minimum); 神经网络对训练数据中的噪声 (noise) 非常敏感; 回归方法对比: Ridge Regression (岭回归) 的惩罚项是系数的平方和; LASSO 的惩罚项是系数的绝对值和; k-NN 回归是通过取  $k$  个最近邻居的  $y$  值的平均值 (average) 来预测的; Weka: 在 Weka 中, 分类 (Classification) 和回归 (Regression) 任务都是在 "Classify" 标签页下进行的。识别激活函数 (Activation Functions): 能够根据图像或公式识别常见的激活函数: Sigmoid: S 形曲线, 值域 (0, 1); Tanh: S 形曲线, 值域 (-1, 1); Sign (符号函数): 阶跃函数, 输出 -1 或 1; Linear (线性函数); 感知机输出计算:

感知机权重更新规则 (Weight Update Rule): 预测值偏小增大权重 偏大减小权重; 反向传播 (Backpropagation): 核心数学原理是链式法则 (Chain Rule); 流程是: 先进行前向传播 (Forward pass) 计算输出和误差, 再进行后向传播 (Backward pass) 更新权重; SVM 回归: 目标是找到一个函数, 使其范数 ( $\beta^* \beta$ ) 最小化; 引入松弛变量 (slack variable) 来处理不能完美拟合的情况; 使用核函数 (Kernel) 将数据投影到高维空间以解决非线性问题; 用于训练多层神经网络 (MLP) 的主要算法是 梯度下降 (Gradient Descent) 或 反向传播 (Backpropagation); 在 Weka 中, 实现支持向量机 (SVM) 的算法名称通常显示为 SMO (Sequential Minimal Optimization); 位于输入层和输出层之间的层被称为 隐藏层 (Hidden Layers); 神经网络中处理加权求和信号的函数  $g(S_i)$  被称为激活 (Activation) 函数;

Lec6 深度学习基础与优化 (Deep Learning Fundamentals); 浅层 vs 深层; 浅层模型面临

“选择性-不变性困境” (Selectivity-Invariance dilemma), 即难以同时做到识别物体（如区分不同种类的狗）又忽略无关变化（如姿态、背景）；深度学习通过多层次抽象自动进行表示

学习 (Representation Learning), 无需人工设计特征; 优化算法: 随机梯度下降 (SGD): 深度学习中最常用。通过小批量样本计算平均梯度, 虽然有噪声 (noisy), 但计算效率高; 鞍点 (Saddle Points): 在高维空间中, 优化算法更容易遇到鞍点 (梯度为 0 但不是极值点), 而非局部极小值; 卷积神经网络 (CNN): 四大核心机制 (Key Ideas): 局部连接 (Local connections); 权值共享 (Shared weights); 池化以及多层结构; 通常由卷积层 (Convolution) -> 非线性激活 (ReLU) -> 池化层 (Pooling)交替组成; 循环神经网络 (RNN): 应用模式: 一对多 (One-to-Many): 图像 -> 序列 (如图像描述 Image Captioning); 多对一 (Many-to-One): 序列 -> 向量 (如情感分析 Sentiment Classification); 多对多 (Many-to-Many): 序列 -> 序列 (如机器翻译 Machine Translation); 问题: 普通 RNN 面临 梯度消失 (Gradient vanishing) 和梯度爆炸问题, 难以处理长序列; 变体: LSTM (长短期记忆网络) 和 GRU; 迁移学习 (Transfer Learning): 将在大规模数据集 (源域 Source Domain) 上训练好的模型 (Backbone), 通过微调 (Fine-tuning) 应用到数据量较小的特定任务 (目标域 Target Domain); 策略: 固定部分层参数, 只训练剩余层

## Lec7-8

Wind Energy Basics & Data (风能基础与数据): SCADA System (SCADA 系统): SCADA 代表数据采集与监视控制系统。它设计用于监控风力涡轮机状态, 同时也收集大量数据。虽然采样频率很高, 但由于存储限制, 数据通常以 10 分钟为间隔 进行存储; 风能系统涉及作为燃料的不确定风力, 在恶劣环境中运行, 包含昂贵的资产, 且能量转换过程具有高度非线性; Wind Statistics & Physics: Weibull Distribution (威布尔分布)风速的变化通常用威布尔分布来描述, 该分布由 尺度参数 (lambda) 和 形状参数 (k) 定义; 如果 k=1, 它变为指数分布; 如果 k=3/4, 它看起来类似于正态分布; The Average Bottle Fallacy (平均瓶谬误): 你不能仅使用平均风速来计算风的平均能量含量, 因为功率与 风速的立方 成正比。必须使用威布尔分布进行计算 Betz' Law (贝兹定律)风力涡轮机从风中提取动能的理论最大值为 16/27 (约 59%)。现代转子的功率系数 ( $C_p$ ) 通常达到 0.4 – 0.5 (理论极限的 70% – 80%) ;

Aerodynamics Model (空气动力学模型); Chinese: 机械功率  $P_a$  计算公式为  $P_a = \frac{1}{2} \rho \pi R^2 C_p(\lambda, \beta) v^3$ , 其中  $\lambda$  是叶尖速比 ( $\lambda = \frac{\omega R}{v}$ )。 Wind

Turbine Operation & Metrics (风力涡轮机运行与指标): Operational Speeds (运行风速)切入风速 (Cut-in): 涡轮机开始发电 (通常为 3 – 5 m/s)。额定风速 (Rated): 涡轮机达到最大效率/功率 。切出风速 (Cut-out): 涡轮机停机以避免损坏 (通常 >25 m/s) ; Performance Metrics (性能指标):容量系数 (Capacity Factor): 实际年发电量 与 理论最大发电量 (即全天候以额定功率运行) 之比。通常为 20%–70%; 可用性系数 (Availability Factor): 涡轮机可用时间与一年总时间之比; Machine Learning Applications (机器学习应用): Prediction vs. Modeling (预测与建模)建模: 使用当前输入  $X_t$  估计当前输出  $Y_t$  预测: 使用当前/过去输入  $X_t$  估计未来输出  $Y_{t+n}$ ; Feature Selection (特征选择):方法包括领域知识、试错法、LASSO (线性模型)、重要性排序 (Boosting 树) 和全局敏感性分析 (神经网络); Data-Driven Approaches (数据驱动方法):用于风电功率预测/建模的算法包括 k-NN (k-近邻算法)、神经网络、支持向量机 (SVM) 和 Boosting 树.

