

编译原理、数据库、人工智能

《机器学习原理》

gcc的编译过程？

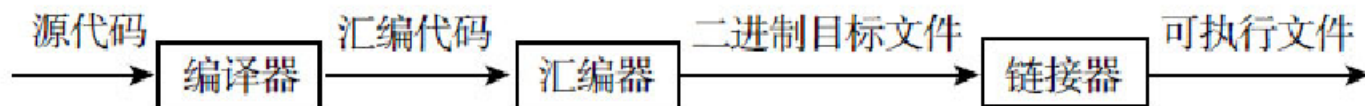


图1-1 编译的流程

预处理 (Processing)、编译(Compilation)、汇编(Assembly)、链接(Linking)。

1. 预处理/预编译：将源文件和头文件编译成.i文件。主要工作有：
 - 将所有的#define删除，并展开所有的宏定义
 - 处理预编译指令如#if #ifdef #endif
 - 处理#include指令，将被包含的文件插入到该预编译指令的位置。
 - 删除注释、添加行号以便出错时显示信息
2. 编译：对预处理后的.i文件进行词法分析、语法分析、语义分析以及优化，产生相应的汇编代码文件.s文件
3. 汇编：将汇编指令翻译成机器指令。.s文件变为.o文件。
4. 链接：将其他文件链接在一起，产生可执行文件。

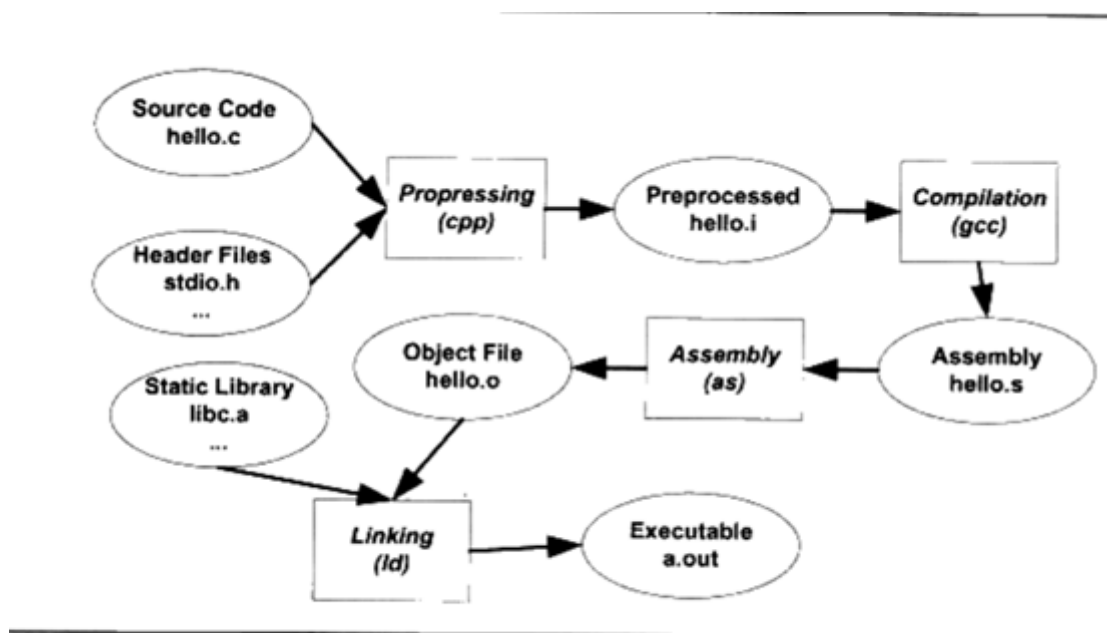


图 2-1 GCC 编译过程分解

编译阶段的过程?

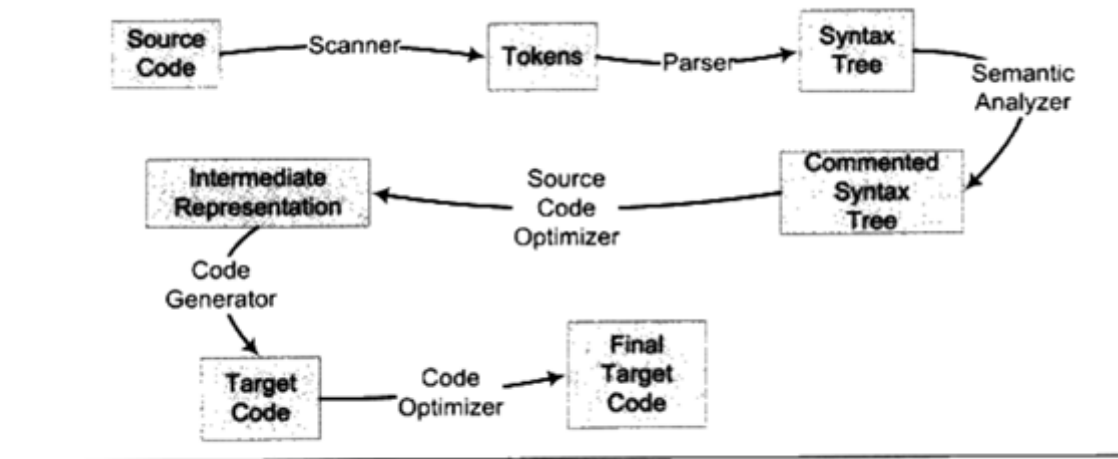
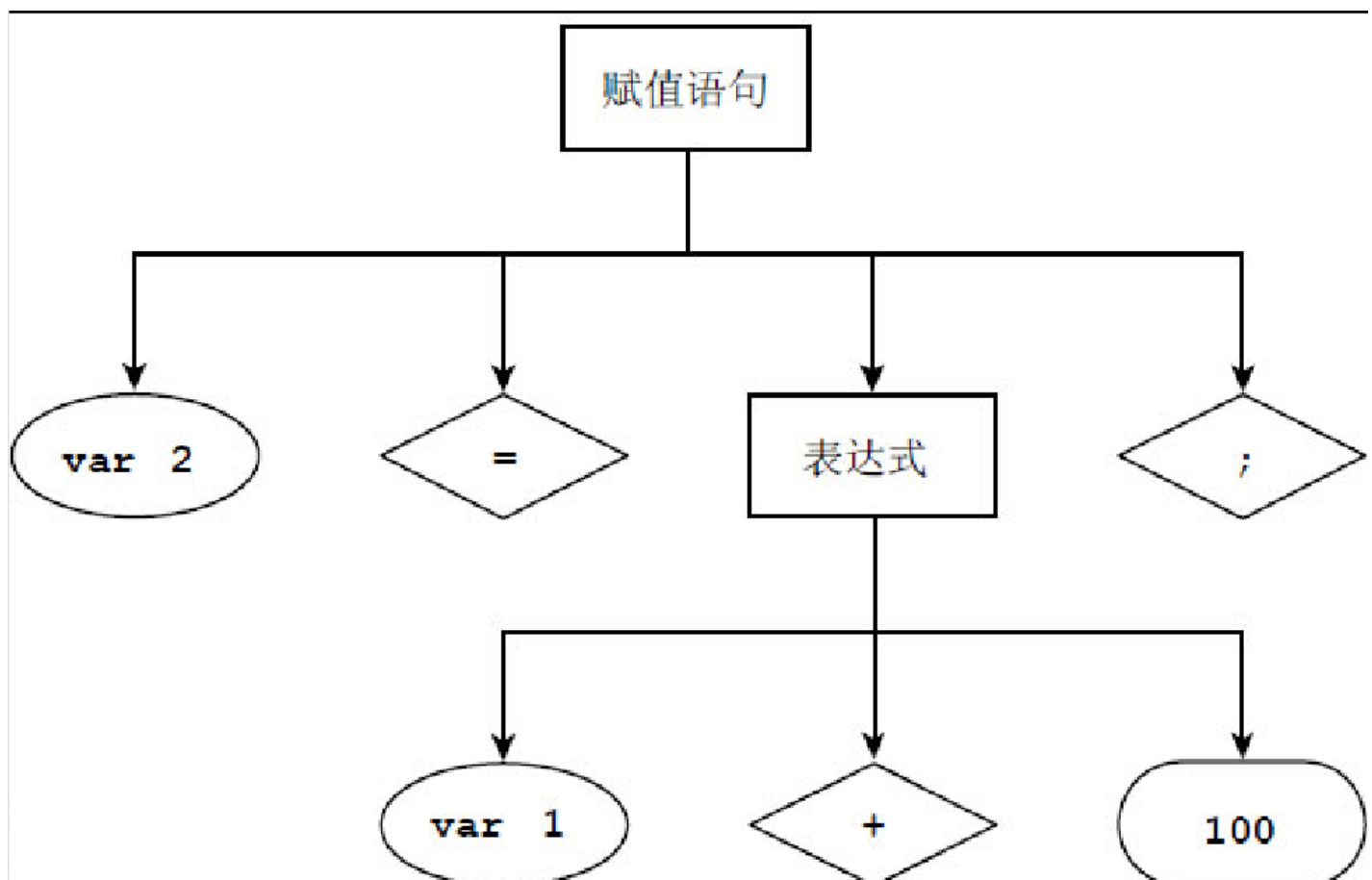


图 2-2 编译过程

- 1. 词法分析。扫描器Scanner通过有限状态机将源代码的字符分割成一系列的记号，例如关键字、标识符、数字、字符串、特殊符号（加号、等号）。扫描过程中扫描器会将标识符存放到符号表，将数字、字符串常量存放到文字表等，以便后续步骤使用。词法分析还能检测单词拼写错误和非法字符。
- 2. 语法分析：输入词法分析后的词法记号序列，根据LL(1)文法进行分析，输出抽象语法树：

```
'var2=var1+100 ;
```



3. 语义分析

语法分析将单词根据文法组成表达式或子程序，而语义分析对结构上正确的语法树进行上下文有关性质的审查，例如赋值语句左右两边数值类型不匹配、函数调用使用了错误的参数、函数声明了返回值类型却没有返回参数等错误。

4. 中间代码生成

5. 代码优化

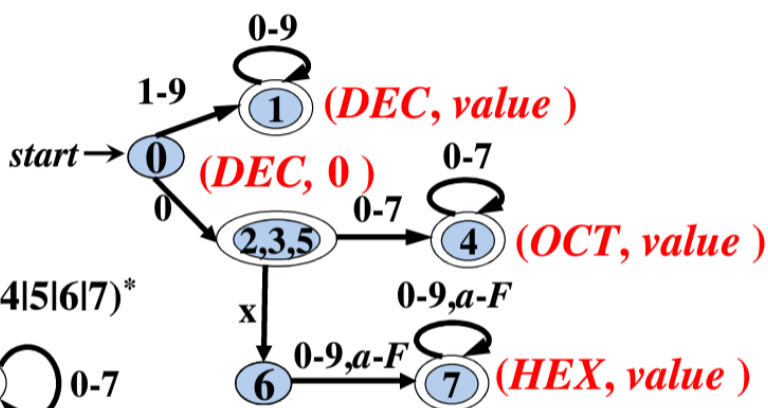
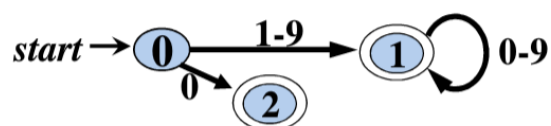
6. 目标代码生成

什么是有穷自动机？

有穷自动机 (Finite Automata, FA), 是对一类处理系统建立的数学模型。这类系统具有一系列离散的输入输出信息和有穷数目的内部状态，只需要根据当前所处的状态和当前面临的输入信息就可以决定系统的后继行为。例如电梯的状态为目前所在的楼层和运动方向，乘客的需求就是输入信息，改变电梯状态。

识别各进制无符号整数的DFA

$DEC \rightarrow (1|...|9)(0|...|9)^* | 0$



$OCT \rightarrow 0(0|1|2|3|4|5|6|7)(0|1|2|3|4|5|6|7)^*$



$HEX \rightarrow 0x(0|1|...|9|a|...|f|A|...|F)(0|...|9|a|...|f|A|...|F)^*$



什么是自顶向下分析

➤ 例

文法

① $E \rightarrow E + E$

② $E \rightarrow E * E$

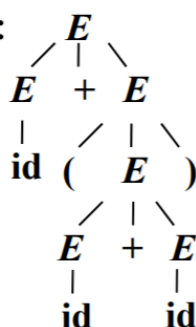
③ $E \rightarrow (E)$

④ $E \rightarrow id$

输入

$id + (id + id)$

分析树:



推导过程: $E \Rightarrow E + E$

$\Rightarrow E + (E)$

$\Rightarrow E + (E + E)$

$\Rightarrow E + (id + E)$

$\Rightarrow id + (id + E)$

$\Rightarrow id + (id + id)$

➤ 每一步推导中, 都需要做两个选择

➤ 替换当前句型中的哪个非终结符

➤ 用该非终结符的哪个候选式进行替换

自顶向下分析采用最左推导, 根据输入流中的下一个终结符, 选择最左非终结符的一个候选式

最左推导 (Left-most Derivation)

➤ 在**最左推导**中，总是选择每个句型的最左**非终结符**进行替换

➤ 例

文法
① $E \rightarrow E + E$
② $E \rightarrow E * E$
③ $E \rightarrow (E)$
④ $E \rightarrow id$
输入
$id + (id + id)$

推导过程

最左推导 ↓

$$\begin{aligned}
 E &\Rightarrow_{lm} E + E \\
 &\Rightarrow_{lm} id + E \\
 &\Rightarrow_{lm} id + (E) \\
 &\Rightarrow_{lm} id + (E + E) \\
 &\Rightarrow_{lm} id + (id + E) \\
 &\Rightarrow_{lm} id + (id + id)
 \end{aligned}$$

↑ 最右归约

每次都从最左边的子节点开始推导，类似DFS

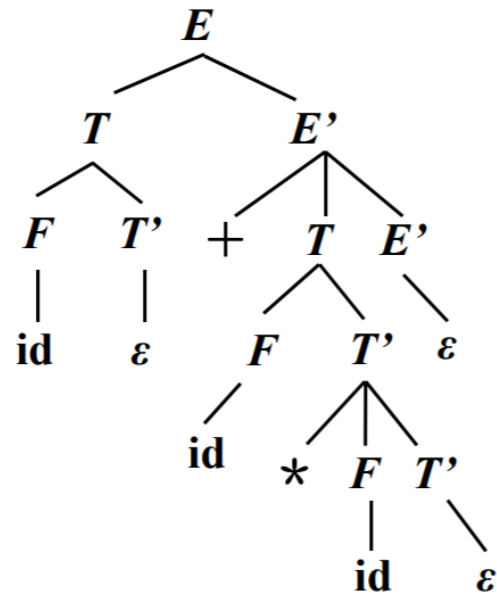
例

➤ 文法

- ① $E \rightarrow T E'$
- ② $E' \rightarrow + T E' \mid \varepsilon$
- ③ $T \rightarrow F T'$
- ④ $T' \rightarrow * F T' \mid \varepsilon$
- ⑤ $F \rightarrow (E) \mid id$

➤ 输入

$id + id * id$
 $\uparrow \uparrow \uparrow \uparrow \uparrow \uparrow$



自顶向下语法分析的通用形式

➤ 递归下降分析 (Recursive-Descent Parsing)

- 由一组过程组成，每个过程对应一个非终结符
- 从文法开始符号 S 对应的过程开始，其中递归调用文法中其它非终结符对应的过程。如果 S 对应的过程体恰好扫描了整个输入串，则成功完成语法分析

```
void A() {  
1)  选择一个 $A$ 产生式,  $A \rightarrow X_1 X_2 \dots X_k$  ;  
2)  for ( $i = 1$  to  $k$ ) {  
3)      if ( $X_i$ 是一个非终结符号)  
4)          调用过程  $X_i()$  ;  
5)      else if ( $X_i$ 等于当前的输入符号 $a$ )  
6)          读入下一个输入符号;  
7)      else /* 发生了一个错误 */;  
    }  
}
```

可能需要回溯(backtracking),
导致效率较低

解释：对每个左侧的非终结符，调用它的过程；如果是终结符，匹配当前输入的符号，依次循环。

事务

是指访问并可能更新数据库中各种数据项的一个程序执行单元(unit)。事务通常由高级数据库操纵语言或编程语言（如SQL，C++或Java）书写的用户程序的执行所引起，并用形如begin transaction和end transaction语句（或函数调用）来界定。事务由事务开始(begin transaction)和事务结束(end transaction)之间执行的全体操作组成。

例如：在关系数据库中，一个事务可以是一条SQL语句，一组SQL语句或整个程序。

事务是恢复和并发控制的基本单位。

事务应该具有4个属性：原子性、一致性、隔离性、持久性。这四个属性通常称为ACID特性。

数据库ACID

1. 原子性Atomicity：一个事务被视为一个最小单元，要么全部提交，要么全部回滚
2. 一致性Consistency：数据库事务只会是执行前的状态或是执行后的状态，不会出现执行中的状态。即如果一个事务执行了十秒，那么第一秒读到的结果和第九秒得到的应该是相同的。
3. 隔离性Isolation：一个事务的执行不会被另一个事务影响，互不干扰。
4. 持久性Durability：事务只要提交了，那么数据库中的数据也永久的发生了变化。

数据库中的锁

按锁级别划分：共享锁、排它锁、意向锁。

排它锁 (exclusive lock)

排他锁又叫写锁，如果事务T对A加上排它锁，则其他事务都不能对A加任何类型的锁。获准排它锁的事务既能读数据，又能写数据

共享锁 (share lock)

共享锁又叫读锁，如果事务T对A加上共享锁，则其他事务只能对A再加共享锁，不能加其他锁。共享锁的事务只能读数据，不能写数据。

按使用方式划分：乐观锁、悲观锁

乐观锁和悲观锁的区别在于拿数据的时候是否认为别人会不会修改，乐观锁认为别人不会修改，所以拿数据的时候没有上锁。悲观锁则认为别人会修改，所以直接上锁。

插入100个数据和100万个数据有何区别？

100数量级小，可以随意插入；100万数量级大，如果表里有索引，则索引更新代价很高，可以采取先删除索引再插入，插入完成后再建索引的策略。

数据库第一二三四范式是什么？



- 1NF：如果关系模式R的所有属性均为原子属性，即每个属性都是不可再分的，则称R属于第一范式。1NF是关系模式最基本的条件。
- 2NF：例如学号、课程号推导出分数，这叫完全函数依赖。学号、课程号推导出姓名，这叫部分函数依赖。2NF举例：{学号，课程号，分数，等级}。从1NF消除所有非主属性对主码的部分函数依

如果关系模式 $R \in 1NF$ ，且每个非主属性都完全函数依赖于 R 的主码，则称 R 属于第二范式，简称2NF，记作 $R \in 2NF$ 。

如：关系模式 $TC(T, C)$

主码 $\rightarrow (T, C)$ ；主属性 $\rightarrow T, C$

不存在非主属性对主码的部分函数依赖，因此 $TC \in 2NF$ 。

- 3NF: {学号, 课程号, 分数, 等级}, 学号和课程号能推导分数, 分数又能推导等级, 所以非主属性等级传递函数依赖主码, 不属于第三范式。应该分解为{学号, 课程号, 分数}、{分数, 等级}

如果关系模式 $R \in 2NF$ ，且每个非主属性都不传递函数依赖于 R 的主码，则称 R 属于第三范式，简称3NF，记作 $R \in 3NF$ 。

如：SC(SNo, CNo, Score)

函数依赖为 $(SNo, CNo) \rightarrow Score$ ，非主属性Score不传递函数依赖于主码(SNo, CNo)，因此， $SC \in 3NF$ 。

又如：SD(SNo, SN, Age, Dept, MN)

$SNo \rightarrow Dept$ 和 $Dept \rightarrow MN \leftrightarrow SNo \xrightarrow{t} MN$

非主属性MN与主码SNo间存在着传递函数依赖，所以 $SD \notin 3NF$ 。

- BC范式：不能存在主属性对主码的部分函数依赖。如{学号, 姓名, 性别}

BC 范式的定义

如果关系模式 $R \in 1NF$ ，且所有的函数依赖 $X \rightarrow Y$ ($Y \notin X$)，决定因素 X 都包含了 R 的一个候选码，则称 R 属于 BC 范式，记作 $R \in BCNF$ 。

BCNF 具有如下性质：

如果 $R \in BCNF$ ，则 R 也是 $3NF$ 。

如果 $R \in 3NF$ ，则 R 不一定是 $BCNF$ 。

- 第四范式：一个表的主键只对应一个多值。例如{学号，学生选修课程，学生参与项目}，一个学生可能选修多门课程，参与多门项目，存在两个多值，不属于 $4NF$ 。修改为{学号，学生选修课程}、{学号，学生参与项目}即可。

学过哪些前沿课程？

智能系统？

梯度

在微积分里面，对多元函数的参数求 ∂ 偏导数，把求得的各个参数的偏导数以向量的形式写出来，就是梯度。

那么这个梯度向量求出来有什么意义呢？他的意义从几何意义上讲，就是函数变化增加最快的地方。具体来说，对于函数 $f(x,y)$ ，在点 (x_0, y_0) ，沿着梯度向量的方向就是 $\left(\frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial y_0}\right) T$ 的方向是 $f(x,y)$ 增加最快的地方。或者说，沿着梯度向量的方向，更加容易找到函数的最大值。反过来说，沿着梯度向量相反的方向，也就是 $-\left(\frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial y_0}\right) T$ 的方向，梯度减少最快，也就是更加容易找到函数的最小值。

最小二乘法

[看这篇文章](#)

损失函数表示为：

$$J(\theta_0, \theta_1, \dots, \theta_n) = \sum_{j=1}^m (h_{\theta}(x_0(j)), x_1(j), \dots, x_n(j)) - y(j))^2 = \sum_{j=1}^m m \left(\sum_{i=0}^n \theta_i x_i(j) - y(j) \right)^2$$

利用损失函数分别对 $\theta_i (i=0, 1, \dots, n)$ 求导,并令导数为0可得：

$$\sum_{j=1}^m m \left(\sum_{i=0}^n \theta_i x_i(j) - y_j \right) x_i(j) = 0 \quad (i=0, 1, \dots, n)$$

这样我们得到一个 $N+1$ 元一次方程组，这个方程组有 $N+1$ 个方程，求解这个方程，就可以得到所有的 $N+1$ 个未知的 θ 。

人工智能是什么？

人工智能收集真实世界中的数据，系统地提取这些数据的特征，对数据进行编程，实现根据环境的观测做出理性的行为。人工智能模仿人类大脑思考的方式，从而使计算机去做过去只有人才能做的工作，例如如何识别一只猫。

人工智能近几年火爆是因为计算机算力、存储力的提高以及互联网的崛起。

机器学习

监督式学习和非监督式学习

实际应用中的机器学习在大部分情况下我们都会使用监督式学习。

1. 监督式学习指的是你拥有一个输入变量和一个输出变量，使用某种算法去学习从输入到输出的映射函数这种学习方式就称之为监督式学习，因为算法学习从训练数据集学习的过程可以被看成类似于一名教师在监督学习学习的过程。我们已经知道了正确的答案，而算法不断迭代来对训练数据做出预测同时不断被一名教师修正。当算法达到一个可接受程度的表现时学习过程停止。

监督式学习问题可以进一步被分为回归和分类问题

- **分类**:分类问题指的是当输出变量属于一个范畴，比如“红色”和“蓝色”或者“生病”和“未生病”。
- **回归**:回归问题指的是输出变量是一个实值，比如“价格”和“重量”

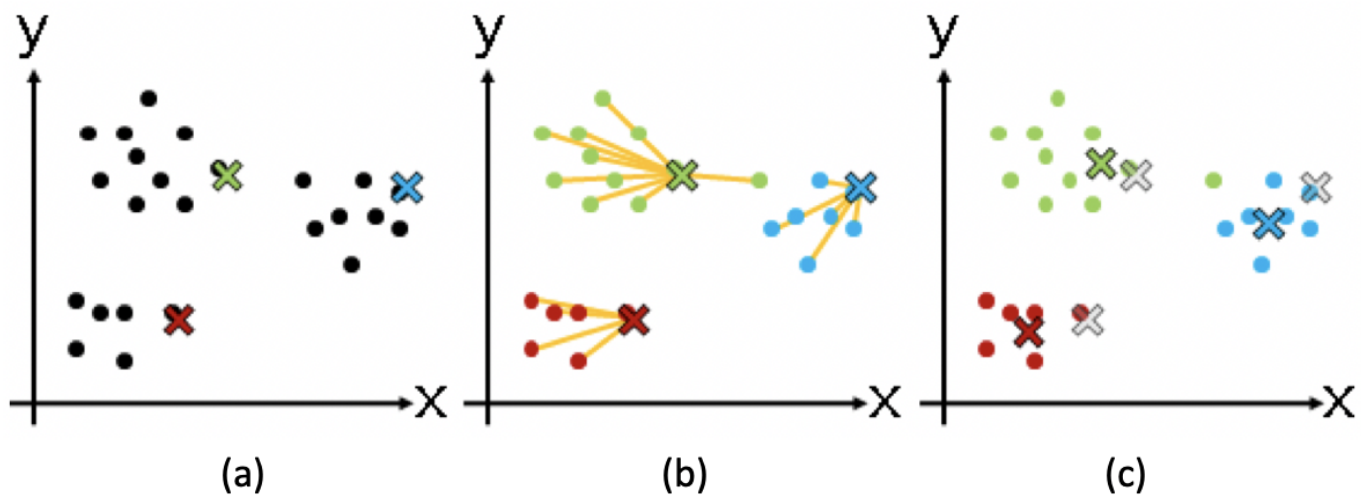
还有一些种类的问题建立在分类和回归之上，包括推荐问题和时序预测

2. 非监督式学习指的是我们只拥有输入变量但是没有相关的输出变量。非监督式学习的目标是对数据中潜在的结构和分布建模，以便对数据作更进一步的学习。这种学习方式就称为非监督式学习，因为其和监督式学习不同，对于学习并没有确切的答案和学习过程也没有教师监督。算法独自运行以发现和表达数据中的有意思的结构。

非监督式学习问题可以进一步分为聚类问题和关联问题

- **聚类问题**：聚类学习问题指的是我们想在数据中发现内在的分组，比如以购买行为对顾客进行分组。

这个算法的名称是 K 均值（K-Means）聚类算法，它让我们可以在一个任意多的数据上，得到一个事先定好群组数量（K）的聚类结果。这种算法的中心思想是：尽量最大化总的群组内相似度，同时尽量最小化群组之间的相似度。群组内或群组间的相似度，是通过各个成员和群组质心相比较来确定的。想法很简单，但是在样本数量达到一定规模后，希望通过排列组合所有的群组划分，来找到最大总群组内的相似度几乎是不可能的。于是人们提出如下的求近似解的方法。从 N 个数据对象中随机选取 k 个对象作为质心，这里每个群组的质心定义是，群组内所有成员对象的平均值。因为是第一轮，所以第 i 个群组的质心就是第 i 个对象，而且这时候我们只有这一个组员。对剩余的对象，测量它和每个质心的相似度，并把它归到最近的质心所属的群组。这里我们可以说距离，也可以说相似度，只是两者呈现反比关系。重新计算已经得到的各个群组的质心。这里质心的计算是关键，如果使用特征向量来表示的数据对象，那么最基本的方法是取群组内成员的特征向量，将它们的平均值作为质心的向量表示。迭代上面的第 2 步和第 3 步，直至新的质心与原质心相等或相差之值小于指定阈值，算法结束。我以二维空间为例子，画张图来展示一下数据对象聚类的过程。



在这张图中，(a)、(b)、(c) 三步分别展示了质心和群组逐步调整的过程。我们一一看。 (a) 步骤是选择初始质心，质心用不同颜色的 x 表示； (b) 步骤开始进行聚类，把点分配到最近的质心所在的组； (c) 步骤重新计算每个群组的质心，你会发现 x 的位置发生了变化。之后就是如此重复，进入下一轮聚类。总的来说，K 均值算法是通过不断迭代、调整 K 个聚类质心的算法。而质心或者群组的中心点，是通过求群组所包含的成员之平均值来计算的。

- **关联问题**：关联问题学习问题指的是我们想发现数据的各部分之间的联系和规则，例如购买 X 物品的顾客也喜欢购买 Y 物品。

高斯滤波

图像大多数噪声均属于高斯噪声，因此高斯滤波器应用也较广泛。高斯滤波是一种线性平滑滤波，适用于消除高斯噪声，广泛应用于图像去噪。

可以简单地理解为，高斯滤波去噪就是对整幅图像像素值进行加权平均，针对每一个像素点的值，都由其本身值和邻域内的其他像素值经过加权平均后得到。

高斯滤波的具体操作是：用一个用户指定的模板（或称卷积、掩膜）去扫描图像中的每一个像素，用模板确定的邻域内像素的加权平均灰度值去替代模板中心像素点的值。[2]

一维高斯分布：

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

蚁群算法

蚁群算法是一种用来寻找优化路径的概率型算法

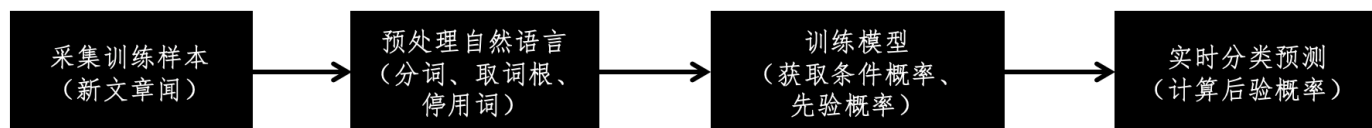
将蚁群算法应用于解决优化问题的基本思路为：用蚂蚁的行走路径表示待优化问题的可行解，整个蚂蚁群体的所有路径构成待优化问题的解空间。路径较短的蚂蚁释放的信息素量较多，随着时间的推进，较短的路径上累积的信息素浓度逐渐增高，选择该路径的蚂蚁个数也愈来愈多。最终，整个蚂蚁会在正反馈的作用下集中到最佳的路径上，此时对应的便是待优化问题的最优解。

遗传算法

遗传算法（Genetic Algorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法。

在一开始需要实现从表现型到基因型的映射即编码工作，如二进制编码。初代种群产生之后，按照适者生存和优胜劣汰的原理，逐代演化产生出越来越好的近似解，在每一代，根据问题域中个体的适应度大小选择个体，并借助于自然遗传学的遗传算子进行组合交叉和变异，产生出代表新的解集的种群。这个过程将导致种群像自然进化一样的后生代种群比前代更加适应于环境，末代种群中的最优个体经过解码，可以作为问题近似最优解。

用朴素贝叶斯文本分类



用词袋模型将语言转换为向量

求文本“中国航母”属于“政治”类话题的概率：我们首先要知道，“中国”“政治”“航母”这些分类的先验概率，其次要知道 $P(\text{中国}|\text{政治})$ 意思为文本“中国”属于政治类话题的先验概率，运用朴素贝叶斯求得后验概率。

$$\begin{aligned}P(\text{政治}|\text{中国航母}) &= P(\text{政治}|\text{中国}, \text{航母}) \\&= P(\text{政治}|\text{中国}) \times P(\text{政治}|\text{航母}) \\&= \frac{P(\text{中国}|\text{政治}) \times P(\text{政治})}{P(\text{中国})} \times \frac{P(\text{航母}|\text{政治}) \times P(\text{政治})}{P(\text{航母})} \\&= \frac{0.0110\% \times 20\%}{0.0106\%} \times \frac{0.0034\% \times 20\%}{0.0028\%} \\&= 20.7547\% \times 24.2857\% \\&= 5.04\%\end{aligned}$$

主成分分析PCA

<https://time.geekbang.org/column/article/87097>

主成分分析 PCA（Principal Component Analysis）和奇异值分解 SVD（Singular Value Decomposition）。这两种方法是从矩阵分析的角度出发，找出数据分布之间的关系，从而达到降低维度的目的

PCA 主成分分析法，它是一种针对数值型特征、较为通用的降维方法。和特征选择不同，它并不需要监督式学习中的样本标签，而是从不同维度特征之间的关系出发，进行了一系列的操作和分析。主要步骤包括，标准化原始的数据矩阵、构建协方差矩阵、计算这种协方差矩阵的特征值和特征向量、挑选较大特征值所对应的特征向量、进行原始特征数据的转换。如果排名靠前的特征向量，或者说主成分，已经包括了足够的信息量，那么我们就可以通过选择较少的主成分，对原始的样本矩阵进行转换，从而达到降维的目的。

例子：

ID	特征1	特征2	特征3	...	特征n-1	特征n
1	1	3	-7	...	-10.5	-8.2
2	2	5	-14	...	2.7	4
...
m	-3	-7	2	...	55	13.6

这个矩阵是 $m \times n$ 维的，其中每一行表示一个样本，而每一列表示一维特征。让我们把这个矩阵称作样本矩阵，现在，我们的问题是，能不能通过某种方法，找到一种变换，可以降低这个矩阵的列数，也就是特征的维数，并且尽可能的保留原始数据中有用的信息

步骤：

1. 标准化样本矩阵中的原始数据（使数据分布在零点附近）
2. 获取标准化数据的协方差矩阵；

$$\text{cov}(x, y) = \frac{\sum_{k=1}^m (x_k - \bar{x})(y_k - \bar{y})}{m - 1}$$

$$COV = \begin{bmatrix} \text{cov}(X_{,1}, X_{,1}) & \text{cov}(X_{,1}, X_{,2}) & \cdots & \text{cov}(X_{,1}, X_{,n-1}) & \text{cov}(X_{,1}, X_{,n}) \\ \text{cov}(X_{,2}, X_{,1}) & \text{cov}(X_{,2}, X_{,2}) & \cdots & \text{cov}(X_{,2}, X_{,n-1}) & \text{cov}(X_{,2}, X_{,n}) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \text{cov}(X_{,n}, X_{,1}) & \text{cov}(X_{,n}, X_{,2}) & \cdots & \text{cov}(X_{,n}, X_{,n-1}) & \text{cov}(X_{,n}, X_{,n}) \end{bmatrix}$$

3. 计算协方差矩阵的特征值和特征向量；
4. 依照特征值的大小，挑选k个主要的特征向量，生成新的矩阵D（n*k），而原始矩阵是m*n，两者相乘即可

假设我们获得了 k 个特征值和对应的特征向量，那么我们就有：

$$Xv_1 = \lambda_1 v_1$$

$$Xv_2 = \lambda_2 v_2$$

...

$$Xv_k = \lambda_k v_k$$

按照所对应的λ数值的大小，对这 k 组的 v 排序。排名靠前的 v 就是最重要的特征向量。假设我们只取前 k1 个最重要的特征，那么我们使用这 k1 个特征向量，组成一个 n×k1 维的矩阵 D。把包含原始数

据的 $m \times n$ 维矩阵 X 左乘矩阵 D ，就能重新获得一个 $m \times k1$ 维的矩阵，达到了降维的目的。

有的时候，我们无法确定 $k1$ 取多少合适。一种常见的做法是，看前 $k1$ 个特征值的和占有所有特征值总和的百分比。假设一共有 10 个特征值，总和是 100，最大的特征值是 80，那么第一大特征值占整个特征值之和的 80%，我们认为它能表示 80% 的信息量，还不够多。那我们就继续看第二大的特征值，它是 15，前两个特征值之和有 95，占比达到了 95%，如果我们认为足够了，那么就可以只选前两大特征值，把原始数据的特征维度从 10 维降到 2 维。

奇异值分解

PCA 是通过分析不同纬度特征之间的协方差，找到包含最多信息量的特征向量，从而实现降维。而 SVD 这种方法试图通过样本矩阵本身的分解，找到一些“潜在的因素”，然后通过把原始的特征维度映射到较少的潜在因素之上，达到降维的目的。

群智感知 (Crowd Sensing)

你一定好奇，这个神秘的“群智感知”究竟是什么？它对我们未来的生活将会产生哪些重要影响？比如——

如何了解城市交通路网的畅通和拥堵情况，并对车流和人群进行引导？

如何低成本、高效率地获知遍布城市的公共设施出现了损坏？

如果发生了安全事故或者突发事件，该如何取证？

城市的空气、水体等的检测如何能更有效？

城市中某一地区的消费者是否更偏爱某些特定的产品？

.....

这些与我们衣食住行息息相关的问题，都可以通过“群智感知”技术来有效解决。

如何“感知”一座城市？

就像人需要通过视觉、听觉、嗅觉、味觉、触觉等来感知周边环境一样，城市管理者也需要通过各种工具或终端，来感知城市安全、交通、环境等的变化，这样才能更好地维护城市安全，开展城市管理。

城市感知任务具有范围广、规模大、任务重等特点。而目前我们的城市感知系统还主要依赖于安装在城市各处的专业传感设施，如摄像头、空气检测装置等，覆盖范围受限，投资和维护成本高，这些都导致“智能感知”效果不佳。

随着城市的快速发展，人们对城市精细化管理的需求，以及对精准化商业服务的需求越来越高。传统的城市感知系统，已经无法满足社会发展需求。

西北工业大学计算机学院院长於志文教授说，近年来，随着计算机科学的突飞猛进，特别是大数据、人工智能、物联网技术的不断突破，群智感知计算逐渐成为计算机学科一个新兴研究热点。

群智感知，是物联网和人工智能相结合的前沿研究方向，它利用大量普通用户使用的移动设备作为基本感知单元，通过物联网和移动互联网进行协作，实现感知任务分发与感知数据收集利用，最终完成大规模、复杂的城市与社会感知任务。

移动群智感知与计算（Mobile Crowd Sensing and Computing, MCSC）

A formal definition of MCSC is as follows: a new sensing paradigm that empowers ordinary citizens to contribute data sensed or generated from their mobile devices and aggregates and fuses the data in the cloud for crowd intelligence extraction and human-centric service delivery

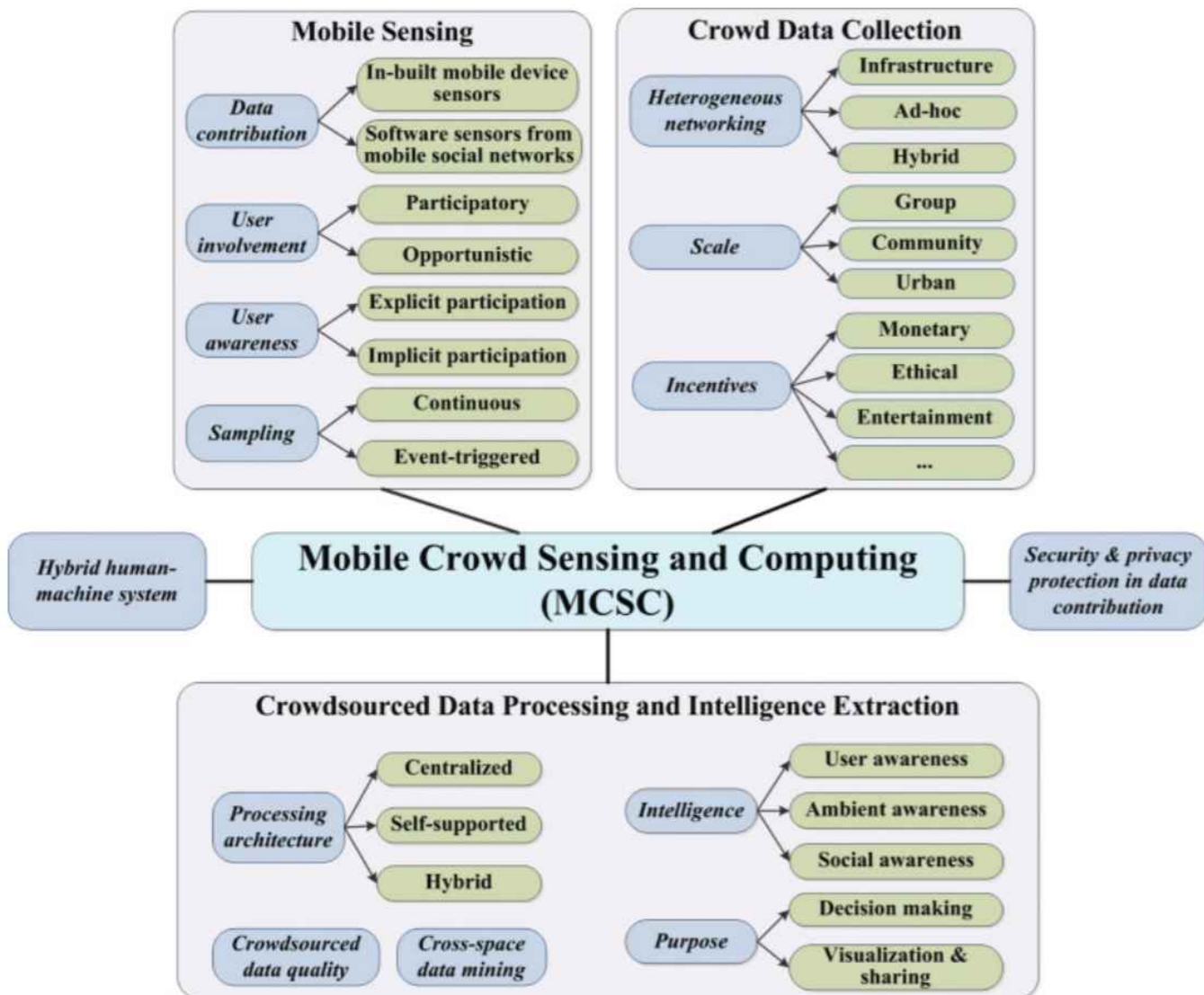


Fig. 1. A taxonomy graph of MCSC.

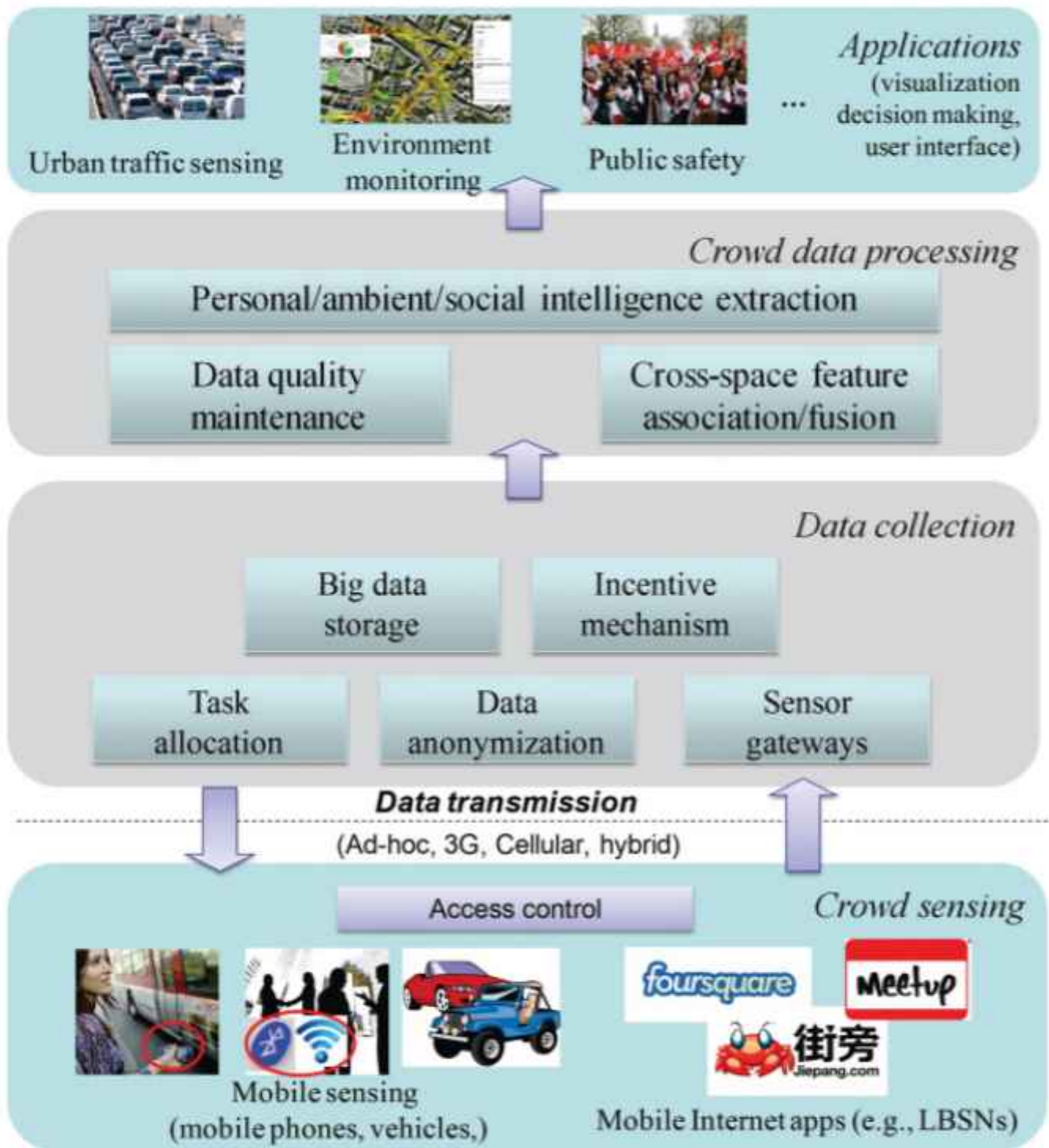


Fig. 2. A reference architecture for MCSC.

分布式计算

随着计算技术的发展，有些应用需要非常巨大的计算能力才能完成，如果采用集中式计算，需要耗费相当长的时间来完成。

分布式计算将该应用分解成许多小的部分，分配给多台计算机进行处理。这样可以节约整体计算时间，大大提高计算效率。

普适计算 (Ubiquitous Computing)

普适计算 (Ubiquitous computing (ubicom)、pervasive computing)，又称**普存计算、普及计算、遍布式计算、泛在计算**，是一个强调和环境融为一体的计算概念，而**计算机**本身则从人们的视线里消失。在普适计算的模式下，人们能够在任何时间、任何地点、以任何方式进行信息的获取与处理。

科学家表示，普适计算的核心思想是小型、便宜、网络化的处理设备广泛分布在日常生活的各个场所，计算设备将不只依赖命令行、图形界面进行人机交互，而更依赖“自然”的交互方式，计算设备的尺寸将缩小到毫米甚至**纳米级**。

在普适计算的环境中，无线传感器网络将广泛普及，在环保、交通等领域发挥作用；人体传感器网络会大大促进健康监控以及人机交互等的发展。各种新型交互技术（如触觉显示、**OLED**等）将使交互更容易、更方便。

普适计算的目的是建立一个充满计算和通信能力的环境,同时使这个环境与人们逐渐地融合在一起。在这个融合空间中人们可以随时随地、透明地获得数字化服务。在普适计算环境下,整个世界是一个网络的世界,数不清的为不同目的服务的计算和通信设备都连接在网络中,在不同的服务环境中自由移动。普适计算是一个世界，在这个世界中，计算机遍布在我们周围，但我们意识不到它们的存在。而且，随着数字设备润物细无声地遍布在我们身边，并被赋予如此强大的计算能力，普适计算也使数据收集达到前所未有的规模。

如果有了普适计算，所有现代汽车都配备的计算机芯片就会向中央程序提交有关实时路况的信息，为其他驾驶员所享用。

云计算

云计算 (cloud computing) 是分布式计算的一种，指的是通过网络“云”将巨大的数据计算处理程序分解成无数个小程序，然后，通过多部服务器组成的系统进行处理和分析这些小程序得到结果并返回给用户。特点有：**虚拟化技术、按需部署、可靠性高**

通常，它的服务类型分为三类，即基础设施即服务(**IaaS**)、平台即服务(**PaaS**)和软件即服务(**SaaS**)。这3种云计算服务有时称为云计算堆栈，因为它们构建堆栈，它们位于彼此之上，以下是这三种服务的概述：[6]

1、基础设施即服务IaaS)

基础设施即服务是主要的服务类别之一，它向云计算提供商的个人或组织提供虚拟化计算资源，如虚拟机、存储、网络和操作系统。[6]

2、平台即服务(PaaS)

平台即服务是一种服务类别，为开发人员提供通过全球互联网构建应用程序和服务的平台。Paas为开发、测试和管理软件应用程序提供按需开发环境。[6]

3、软件即服务(SaaS)

软件即服务也是其服务的一类，通过互联网提供按需软件付费应用程序，云计算提供商托管和管理软件应用程序，并允许其用户连接到应用程序并通过全球互联网访问应用程序。 [6]

复习打卡

- [illegible]