



# 机器学习及其MATLAB实现—从基础到实践 第11课

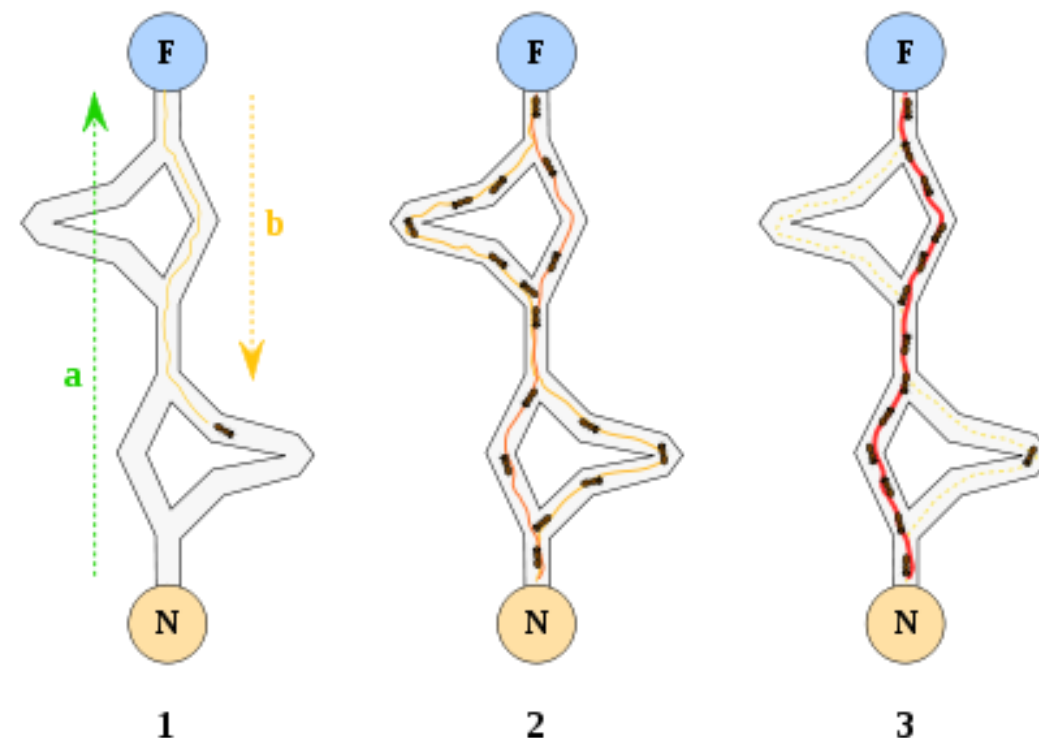
**【声明】** 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

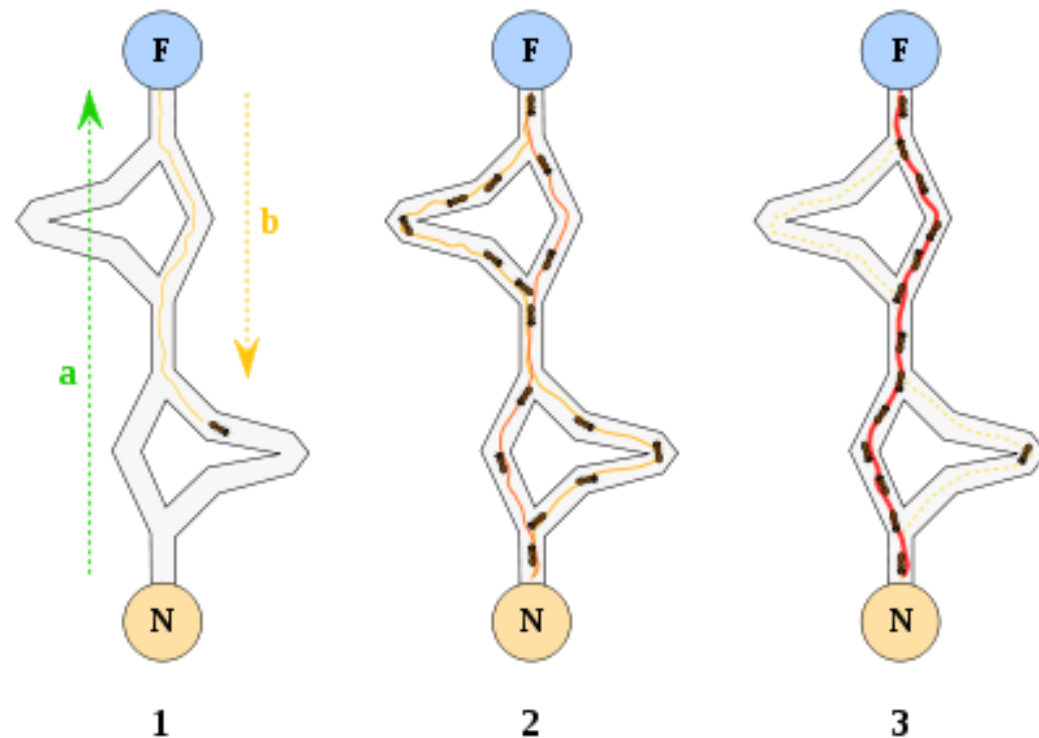
<http://edu.dataguru.cn>

- 第一课 MATLAB入门基础
- 第二课 MATLAB进阶与提高
- 第三课 BP神经网络
- 第四课 RBF、GRNN和PNN神经网络
- 第五课 竞争神经网络与SOM神经网络
- 第六课 支持向量机 ( Support Vector Machine, SVM )
- 第七课 极限学习机 ( Extreme Learning Machine, ELM )
- 第八课 决策树与随机森林
- 第九课 遗传算法 ( Genetic Algorithm, GA )
- 第十课 粒子群优化 ( Particle Swarm Optimization, PSO ) 算法
- **第十一课 蚁群算法 ( Ant Colony Algorithm, ACA )**
- 第十二课 模拟退火算法 ( Simulated Annealing, SA )
- 第十三课 降维与特征选择

- 蚁群算法（Ant Colony Algorithm, ACA）由**Marco Dorigo**于1992年在他的博士论文中首次提出，该算法模拟了自然界中蚂蚁的觅食行为。
- 蚂蚁在寻找食物源时，会在其经过的路径上释放一种信息素，并能够感知其它蚂蚁释放的信息素。**信息素浓度的大小表征路径的远近，信息素浓度越高，表示对应的路径距离越短。**
- 通常，蚂蚁会以较大的概率优先选择信息素浓度较高的路径，并释放一定量的信息素，以增强该条路径上的信息素浓度，这样，会形成一个**正反馈**。最终，蚂蚁能够找到一条从巢穴到食物源的最佳路径，即距离最短。
- 值得一提的是，生物学家同时发现，**路径上的信息素浓度会随着时间的推进而逐渐衰减。**



- 将蚁群算法应用于解决优化问题，其基本思路为：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。



不失一般性，设整个蚂蚁群体中蚂蚁的数量为 $m$ ，城市的数量为 $n$ ，城市 $i$ 与城市 $j$ 之间的相互距离为 $d_{ij}$  ( $i, j = 1, 2, \dots, n$ )， $t$ 时刻城市 $i$ 与城市 $j$ 连接路径上的信息素浓度为 $\tau_{ij}(t)$ 。初始时刻，各个城市间连接路径上的信息素浓度相同，不妨设为 $\tau_{ij}(0) = \tau_0$ 。

蚂蚁 $k$  ( $k = 1, 2, \dots, m$ ) 根据各个城市间连接路径上的信息素浓度决定其下一个访问城市，设 $P_{ij}^k(t)$ 表示 $t$ 时刻蚂蚁 $k$ 从城市 $i$ 转移到城市 $j$ 的概率，其计算公式如下：

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in allow_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & s \in allow_k \\ 0, & s \notin allow_k \end{cases} \quad (1)$$

其中， $\eta_{ij}(t)$ 为启发函数， $\eta_{ij}(t) = 1/d_{ij}$ ，表示蚂蚁从城市 $i$ 转移到城市 $j$ 的期望程度。 $allow_k$  ( $k = 1, 2, \dots, m$ )

为蚂蚁 $k$ 待访问城市的集合。开始时， $allow_k$ 中有 $(n-1)$ 个元素，即包括除了蚂蚁 $k$ 出发城市的其它所有城市。随着时间的推进， $allow_k$ 中的元素不断减少，直至为空，即表示所有的城市均访问完毕。 $\alpha$ 为信息素重要程度因子，其值越大，表示信息素的浓度在转移中起的作用越大； $\beta$ 为启发函数重要程度因子，其值越大，表示启发函数在转移中的作用越大，即蚂蚁会以较大的概率转移到距离短的城市。

在蚂蚁释放信息素的同时，各个城市间连接路径上的信息素逐渐消失，设参数  $\rho (0 < \rho < 1)$  表示信息素的挥发程度。因此，当所有蚂蚁完成一次循环后，各个城市间连接路径上的信息素浓度需进行实时更新，具体公式如下：

$$\begin{cases} \tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \Delta \tau_{ij} \\ \Delta \tau_{ij} = \sum_{k=1}^n \Delta \tau_{ij}^k \end{cases}, 0 < \rho < 1 \quad (2)$$

其中， $\Delta \tau_{ij}^k$  表示第  $k$  只蚂蚁在城市  $i$  与城市  $j$  连接路径上释放的信息素浓度， $\Delta \tau_{ij}$  表示所有蚂蚁在城市  $i$  与城市  $j$  连接路径上释放的信息素浓度之和。

## 1. ant cycle system 模型

ant cycle system 模型中,  $\Delta\tau_{ij}^k$  的计算公式如式 (3) 所示。

$$\Delta\tau_{ij}^k = \begin{cases} Q / L_k, & \text{第 } k \text{ 只蚂蚁从城市 } i \text{ 访问城市 } j \\ 0, & \text{其它} \end{cases} \quad (3)$$

其中,  $Q$  为常数, 表示蚂蚁循环一次所释放的信息素总量;  $L_k$  为第  $k$  只蚂蚁经过路径的长度。

## 2. ant quantity system 模型

ant quantity system 模型中,  $\Delta\tau_{ij}^k$  的计算公式如式 (4) 所示。

$$\Delta\tau_{ij}^k = \begin{cases} Q / d_{ij}, & \text{第 } k \text{ 只蚂蚁从城市 } i \text{ 访问城市 } j \\ 0, & \text{其它} \end{cases} \quad (4)$$

## 3. ant density system 模型

ant density system 模型中,  $\Delta\tau_{ij}^k$  的计算公式如式 (5) 所示。

$$\Delta\tau_{ij}^k = \begin{cases} Q, & \text{第 } k \text{ 只蚂蚁从城市 } i \text{ 访问城市 } j \\ 0, & \text{其它} \end{cases} \quad (5)$$



## (1) 初始化参数

在计算之初，需对相关的参数进行初始化，如蚁群规模（蚂蚁数量） $m$ 、信息素重要程度因子 $\alpha$ 、启发函数重要程度因子 $\beta$ 、信息素挥发因子 $\rho$ 、信息素释放总量 $Q$ 、最大迭代次数 $iter\_max$ 、迭代次数初值 $iter=1$ 。

## (2) 构建解空间

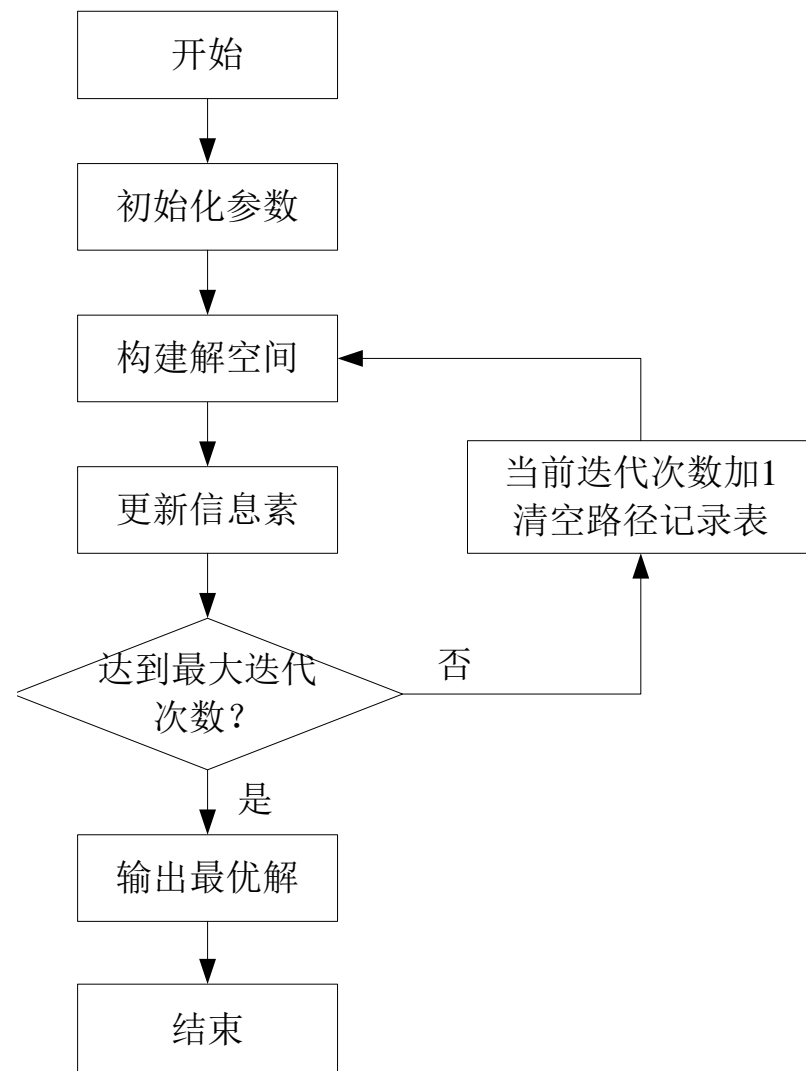
将各个蚂蚁随机地置于不同出发点，对每个蚂蚁 $k(k=1,2,...,m)$ ，计算其下一个待访问的城市，直到所有蚂蚁访问完所有的城市。

## (3) 更新信息素

计算各个蚂蚁经过的路径长度 $L_k(k=1,2,...,m)$ ，记录当前迭代次数中的最优解（最短路径）。同时，对各个城市连接路径上的信息素浓度进行更新。

## (4) 判断是否终止

若 $iter < iter\_max$ ，则令 $iter = iter + 1$ ，清空蚂蚁经过路径的记录表，并返回步骤（2）；否则，终止计算，输出最优解。



- 采用正反馈机制，使得搜索过程不断收敛，最终逼近于最优解；
- 每个个体可以通过释放信息素来改变周围的环境，且每个个体能够感知周围环境的实时变化，个体间通过环境进行间接地通讯；
- 搜索过程采用分布式计算方式，多个个体同时进行并行计算，大大提高了算法的计算能力和运行效率；
- 启发式的概率搜索方式，不容易陷入局部最优，易于寻找到全局最优解。

# 旅行商问题 (TSP) 概述

- Traveling Salesman Problem, TSP
- 在N个城市中各经历一次后再回到出发点，使所经过的路程最短。
- 若不考虑方向性和周期性，在给定N的条件下，可能存在的闭合路径数目为 $1/2 \cdot (N-1)!$ 。当N较大时，枚举法的计算量之大难以想象。（按照枚举法，我国31个直辖市、省会和自治区首府（未包括港、澳、台）的巡回路径应有约 $1.326 \cdot 10^{32}$ 种）
- TSP问题经常被视为验证优化算法性能的一个“金标准”。



## Chinese TSP Optimization

- **ismember, ~**

`tf = ismember(A, S)` returns a vector the same length as `A`, containing logical 1 (true) where the elements of `A` are in the set `S`, and logical 0 (false) elsewhere.

`~expr` represents a logical NOT operation applied to expression `expr`.

- **cumsum**

`B = cumsum(A)` returns the cumulative sum along different dimensions of an array.

- **num2str**

`str = num2str(A)` converts array `A` into a string representation `str` with roughly four digits of precision and an exponent if required.

- **text**

`text(x,y,'string')` adds the string in quotes to the location specified by the point `(x,y)` `x` and `y` must be numbers of class `double`.

- Dataguru ( 炼数成金 ) 是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 <http://edu.dataguru.cn>

# Thanks

**FAQ时间**