

Chapter 7 随机方法

Author: 中山大学 17数据科学与计算机学院 YSY

<https://github.com/ysyisyourbrother>

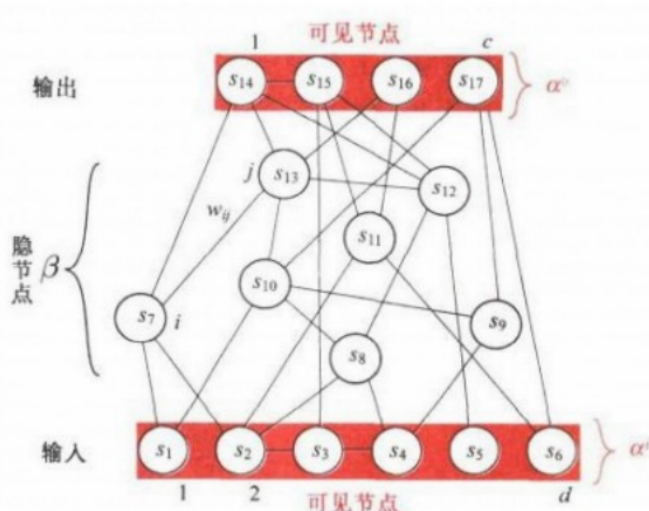
模拟退火

假设给定多个变量 $s_i, i = 1 \dots N$, 其中每个变量的数值都取两个离散值+1和-1. 优化问题是: 确定N个变量的合适取值, 使下面的代价函数或能量函数最小:

$$E = -\frac{1}{2} \sum_{i,j=1}^N w_{ij} s_i s_j$$

用网络表示这个优化问题, 其中连线表示权重:

图 7-7 当图 7-1 那样的网络用于学习时, 区别两类可见单元是很重要的。即 d 个输入单元和 c 个输出单元(它们的作用是接收外部特征和类别信息), 当然还包括对隐单元的区分。整个网络的状态用整数 γ 标记。因为这里有 17 个二值节点, 所以 γ 限制在 $0 \leq \gamma < 2^{17}$ 间。输入可见节点的状态用 α^i 表示, 而输出可见节点用 α^o (上标并不是值数, 仅分别表示输入或输出)。在图中的情况下, α^i 的范围是 $0 \leq \alpha^i < 2^d$ 。 α^o 的范围是 $0 \leq \alpha^o < 2^c$ 。隐节点的状态用 β 标记, 其范围为 $0 \leq \beta < 2^{N-c-d}$ 。



在物理上, 让一个金属多原子系统到达最低能量的方法称为退火。在物理学上的退火, 系统首先被加热, 保证每一组份具有充分随机性, 系统因此可以跃迁到较高能量的状态。退火过程中, 系统温度缓慢逐步降低, 直到零温度态, 此时没有随机变动, 系统被松弛到一个很低的能量构型。

这种退火相当有效, 系统发现最优构型的可能性很大, 随着温度的降低, 发现全局最小能量的概率在增加。

Boltzmann因子

系统位于一个特定构型 γ 具有能量 E 的概率分布为:

$$P(\gamma) = \frac{e^{-E/T}}{Z(T)}$$

Z 是一个归一化常数, 相当于对多有构型的求和, 为了保证上述公式是一个真的概率。分子称为 Boltzmann 因子, 归一化常数在模拟退火算法中并不需要计算。

对于一个互相无交互的磁体系统, 有一个磁体指向上方, $s_i = +1$, 则贡献一点正的能量。于是这个磁体系统的总能量就正比于上方向的磁体个数。系统具有特定能量的概率和具有这个能量的构型的个数有关。

这其实是一个二项分布的问题。考虑能量最高的构型, 所有磁体同向, 具有这种能量的只有一种构型。而第二高的是一个反向其他同向, 于是有 N 个构型。再低一个就是 2 个反向, 其他同向, 此时有 $N(N-1)/2$ 种, 依次类推。

可见随着能量的降低, 可能构型数量指数上升。

而公式还和温度 T 有关，直接分析公式可以看出：

1. 固定 E ，随着 T 升高，允许的构型数目以指数减少。
2. 而且，高温给了系统更多的能量，使得出现高能量构型的概率增大
3. 高温时，所有构型分布均匀（exp函数曲线平稳），而低温时，系统大部分聚集在低温构型周围（exp函数在低温处值大）

模拟退火算法

1. 将网络随机初始化，并设定一个高的初始温度 T （一个控制随机性的超参数）
2. 接着随机选择一个节点 i ，将其状态反转，计算原本总能量 E_a 和反转后系统总能量 E_b 。如果能量更低，就接受这个状态改变。如果能量升高，就按照下列概率接受这个状态改变：

$$e^{-\Delta E_{ab}/T}$$

这个步骤的关键在于，它使系统有可能从局部最小值跳出，这个偶尔接受能量增加的改变是退火算法成功的关键。

当温度很高的时候，随机性强，上述表达式的值接近 1，也就是每个构型的出现概率都接近相同。

3. 反复随机轮询节点，按照上述的方法进行状态改变，系统的能量会逐渐降低。此时**同时降低温度**，此时，接受能量增加的状态跃迁概率也下降。最终模拟退火算法终止于温度很低的情况，如果冷却过程充分的慢，那么系统落在最低能量状态的概率也会非常大并且有望是全局最小点。

算法实现如下：

算法 1 （随机模拟退火）

```
1 begin initialize  $T(k), k_{\max}, s_i(1), w_{ij} \quad i, j = 1, \dots, N$ 
2    $k \leftarrow 0$ 
3   do  $k \leftarrow k + 1$ 
4     do 随机地选择节点  $i$ ; 假设它的状态为  $s_i$ 
5        $E_a \leftarrow -1/2 \sum_j^{N_i} w_{ij} s_i s_j$ 
6        $E_b \leftarrow -E_a$ 
7       if  $E_b < E_a$ 
8         then  $s_i \leftarrow -s_i$ 
9         else if  $e^{-(E_b - E_a)/T(k)} > \text{Rand}[0, 1)$ 
10          then  $s_i \leftarrow -s_i$ 
11       until 所有节点轮询多次
12     until  $k = k_{\max}$  或停止准则满足
13   return  $E, s_i, i = 1, \dots, N$ 
14 end
```

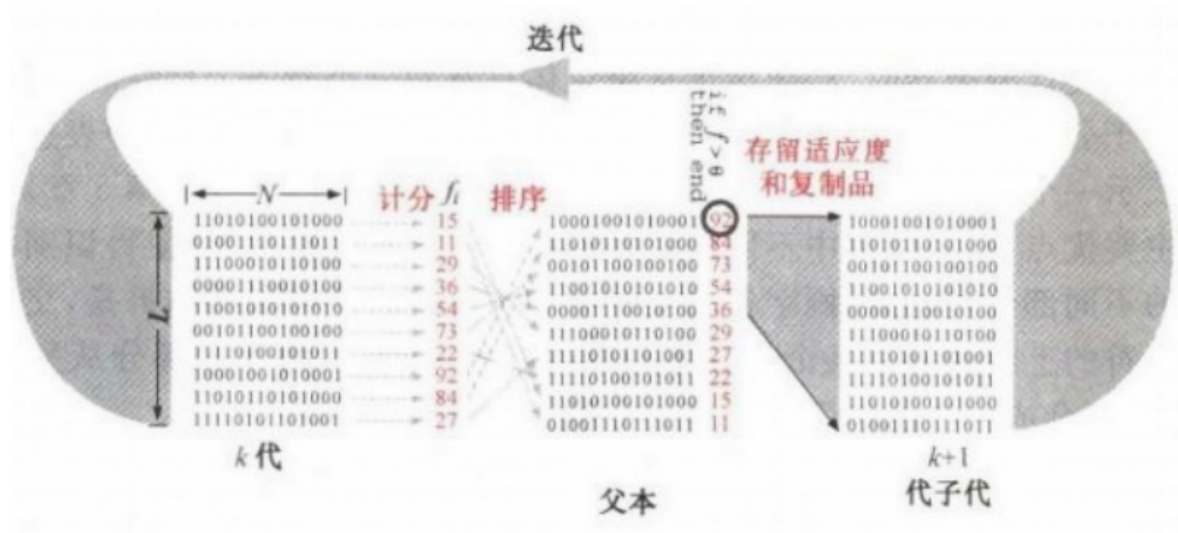
Boltzmann学习

进化方法

遗传算法

每个分类器的基本表达是一个二进制位串，称为染色体。

基本遗传算法是一个随机迭代搜索算法。在第k代的种群中存在L个分类器个体（分类器理解就是权重矩阵，他和样本的特征相乘求和，相当于线性分类器感知机算法）。每个分类器根据它在测试集上的正确率来评价分类器的好坏。这样可以得到L个分数f。染色体根据得分f来排序，按照得分由高到低顺序，对前几的染色体复制，交叉，变异的遗传运算，得到染色体的下一代。重复上述过程。



算法实现如下：

算法 4 （基本遗传算法）

```
1 begin initialize  $\theta, P_{co}, P_{mut}, L, N$ -位染色体
2   do 确定每个染色体的适应度,  $f_i, i=1, \dots, L$ 
3     染色体排序
4     do 选择得分最高的两个染色体
5       if Rand[0,1) <  $P_{co}$  then 交叉一对随机选择的位
6       else 以概率  $P_{mut}$  改变每一位; 删除父染色体
7     until  $N$  个子代被创建
8     until 任何染色体的得分  $f$  超过  $\theta$ 
9   return 最高适应度的染色体(最佳分类器)
10 end
```

遗传算子

1. 复制：染色体被原样复制一遍，不进行改变。
2. 交叉：把两条染色体混合或配对，得到两条新的染色体。在染色体上随机确定一个位置并截断，然后A的第一部分和B的第二部分连接，另一部分也一样。
3. 变异：允许每一位以一个相对较小的概率改变自身。

