1 Introduction

In the last lesson we discussed the issue of using protein networks to study properties of proteins, by using Markov Network:

\[ P(x_1 \ldots x_n) = \frac{1}{z} e^{\sum_i \eta_i x_i + \sum_{i,j \in a} \theta \{x_i = x_j\}} \]

\[ z = \sum_{x_1} \ldots \sum_{x_n} e^{\sum_i \eta_i x_i + \theta N(x_1 \ldots x_n)} \]

We saw that given this setting, there are some things we'd like to do:

1. Find the most likely assignment - \( \text{argmax}_{x_1 \ldots x_n} P(x_1 \ldots x_n | \text{evidence}) \).

2. Inference - compute \( P(x_i | \text{evidence}) \).

3. Learning.

In the last lesson we saw how to solve the first two problems, today we will speak about the third problem - Learning. What do we want to learn? in this model there are two parameters - \( \eta, \theta \), as we already know the parameter \( \eta \) come from external considerations, so the problem is just to learn \( \theta \).

2 learn \( \theta \)

Today we will solve the easy problem - finding \( \theta \) when we got all the annotations. More complexity problem is to find \( \theta \) when we have just part of the annotations, and not all of them. At this case we will had also to find the other annotations, and not just \( \theta \).

2.1 Find The Maximum Point

We would like to find the value of \( \theta \) at the maximum point:

\( \text{argmax}_\theta P(x_1 \ldots x_n : \theta) \)

Define:

\[ N = \sum_{i,j \in a} 1\{x_i = x_j\} \]
Now, to find the maximum we compute the derivative for $\log P$:

$$
\frac{\partial}{\partial \theta} \log P(x_1 \ldots x_n : \theta) = \frac{\partial}{\partial \theta} \left[ \sum_i \eta_i x_i + \theta N(x_1 \ldots x_n) - \log z \right]
= 0 + N(x_1 \ldots x_n) - \frac{\partial}{\partial \theta} \log z
$$

Our problem is to find the diversion of $z$:

$$
\frac{\partial}{\partial \theta} \log z = \frac{1}{z} \frac{\partial}{\partial \theta} z
= \sum_{x_1} \ldots \sum_{x_n} N(x_1 \ldots x_n) e^{\sum_i \eta_i x_i + \theta N(x_1 \ldots x_n)}
= \sum_{x_1} \ldots \sum_{x_n} N(x_1 \ldots x_n) P(x_1 \ldots x_n : \theta)
= E [\mu : \theta]
$$

We can see that the derivative of the expression equal to the expected value. Define:

$$
\hat{\mu} = N(x_1 \ldots x_n)
$$

We have:

$$
\frac{\partial}{\partial \theta} \log P(x_1 \ldots x_n : \theta) = \hat{\mu} - E [\mu : \theta]
$$

Notice that the number of neighbors that agree is equal for chosen $\theta$ and for the expected value.

### 2.2 compute $\theta$

Our problem is that all we know about $\theta$ is the value at the maximum point and the derivative of $\log P$, we want to use this knowledge and compute $\theta$. What is the optional values of $\theta$? if all data points coming from the same annotation $\theta$ will be $(+\infty)$, if all data points disagree with each other $\theta$ will be $(-\infty)$, otherwise $\theta$ will be some finite value. The abstract problem - given function $l(\theta)$, we know how to compute $l'(\theta)$ and would like to find $\theta$. The algorithm that we will see solve this problem by searching along the derivative line

#### 2.2.1 The Algorithm:

1. Find the derivative, and move with the derivative direction (up is right, down is left), each step is twice larger from the previous one. Moving this way, we will capture $\theta$ between two numbers, or decide the it is $\pm \infty$.

2. Search in the area where we capture $\theta$, until we find it. For this search we can use some of the known search methods, for example Binary search. Other option is to use some of the function properties to find it, for example, we can use the square function properties and guess how the function look, using the function value at 3 points. We know that for square functions, if the points are close enough to the maximum, we can guess the maximum using this method.