

Лекция 10. EM-алгоритм

Сергей Лебедев

sergei.lebedev@jetbrains.com

16 апреля 2015 г.

- В прошлый раз на примере датасета *20 newsgroups*¹ мы научились предсказывать по тексту сообщения, в какой список рассылки оно будет отправлено.
- Пример сообщения из списка рассылки `comp.windows.x`:
Hi there,

```
I'm looking for tools that can make X programming
easy. I would like to have a tool that will enable
to create X motif GUI interactively. [...]. A package
that enables to create GUI with no coding at all
(but the callbacks).
Any help will be appreciated.
```

Thanks **Gabi**.

- Хочется научиться по тексту сообщения предсказывать его автора. Какие идеи?

¹<http://qwone.com/~jason/20Newsgroups/>

Известное

- Пусть $V = \{v_1, \dots, v_{|V|}\}$ – упорядоченное множество слов,
- тогда сообщение можно представить в виде вектора, в котором на d -той позиции стоит **количество раз**, которое слово v_d встречается в сообщении.
- Пример:
 - $V = \{\text{what, love, baby, hurt, more}\}$
 - Сообщение «What is love? Baby, don't hurt me, don't hurt me no more» будет векторизовано как $[1, 1, 1, 2, 1]$.

Вопрос

Как определить функцию правдоподобия $p(x|y)$ на множестве $X \equiv \mathbb{N}_0^{|V|}$?

- Мультиномиальное распределение – дискретное распределение с функцией вероятности

$$\text{Mul}(x; n, \vec{\theta}) = \binom{n}{x_1, \dots, x_{|V|}} \theta_1^{x_1} \dots \theta_{|V|}^{x_{|V|}}$$

- и двумя параметрами:

- $n = \sum_{d=1}^{|V|} x_d$ – количество слов в сообщении и
- $\vec{\theta} = \{\theta_1, \dots, \theta_{|V|}\} \in [0, 1]^{|V|}$ – вектор вероятностей каждого слова из словаря, удовлетворяющий $\sum_{d=1}^{|V|} \theta_d = 1$.

Вопрос

Какую комбинаторную интерпретацию можно дать функции вероятности мультиномиального распределения?

Мультиномиальный наивный Байесовский классификатор

- Найдём ОМП априорных вероятностей классов \hat{P}_y и параметров мультиномиального распределения $\hat{\theta}_{yd}$:

$$\hat{P}_y = \frac{\sum_{i=1}^l [y_i = y]}{l} \quad \hat{\theta}_{yd} = \frac{\sum_{i=1}^l [y_i = y] x_{id}}{\sum_{e=1}^{|V|} \sum_{i=1}^l [y_i = y] x_{ie}}$$

- Подставим оценки в оптимальный алгоритм классификации:

$$a(x) = \arg \max_{y \in Y} \lambda_y \hat{P}_y \prod_{d=1}^{|V|} \text{Mul}(x_{id}; \hat{\theta}_{yd})$$

Вопрос

Что делать, если наша обучающая выборка $X^l = (x, y)_{i=1}^l$ состоит из одного элемента?

Неизвестное

- Пусть $Y \equiv \{0, 1\}$ и $X^N = (x_i)_{i=1}^N$ – выборка и (x_l, y_l) – пример, для которого известна метка класса.
- На время забудем про (x_l, y_l) и сфокусируемся на оценивании параметров \hat{P}_y и $\hat{\theta}_{yd}$ по выборке X^N .
- Попытка 1: метод максимального правдоподобия:

$$\begin{aligned}
 L(\Theta) &= \ln \prod_{i=1}^N \sum_{y \in Y} p(x_i, y; \Theta) \\
 &= \sum_{i=1}^N \ln \sum_{y \in Y} p(x_i, y; \Theta) \rightarrow \max_{\Theta}
 \end{aligned}$$

- Попытка 2: введём N **независимых** случайных величин $y_i \in Y$ с функциями вероятности Q_j и запишем функцию **полного** правдоподобия:

$$L(\Theta, \vec{y}) = \sum_{i=1}^N \ln p(x_i, y_i; \Theta)$$

$$\begin{aligned}
 L(\Theta) &= \ln \sum_{\vec{y}} \exp L(\Theta, \vec{y}) = \ln \sum_{\vec{y}} \prod_{i=1}^N p(x_i, y_i; \Theta) \\
 &= \ln \sum_{\vec{y}} \prod_{i=1}^N Q_i(y_i) \frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} = \ln \mathbb{E}_{\vec{y}} \left[\prod_{i=1}^N \frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} \right] \\
 &= \sum_{i=1}^N \ln \mathbb{E}_{y_i} \left[\frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} \right] = \sum_{i=1}^N \ln \sum_{y \in Y} \mathbb{E}_{y_i} \left[\frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} \right] \rightarrow \max_{\Theta}
 \end{aligned}$$

Вопрос

Что произошло с мат. ожиданием $\mathbb{E}_{\vec{y}} [\cdot]$?

- Пусть X – случайная величина и f – вогнутая функция, тогда справедливо

$$f(\mathbb{E}[X]) \geq \mathbb{E}[f(x)],$$

причём, если $\mathbb{E}[X] = X$, то есть X – константная случайная величина, то неравенство превращается в равенство.

- Почему нам это важно?

$$L(\Theta) = \sum_{i=1}^N \ln \mathbb{E}_{y_i} \left[\frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} \right] \geq \sum_{i=1}^N \mathbb{E}_{y_i} \left[\ln \frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} \right]$$

- Нижнюю оценку проще оптимизировать! Кстати, **почему?**

- В предыдущих рассуждениях мы не делали никаких предположений о распределении y_i , поэтому можем выбрать его удобным нам образом:

$$Q_i(y_i) \propto p(x_i, y_i; \Theta) = \frac{p(x_i, y_i; \Theta)}{\sum_{y \in Y} p(x_i, y; \Theta)} = P(y_i | x_i; \Theta)$$

- Получили, что

$$\frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} = \frac{p(x_i, y_i; \Theta) \sum_{y \in Y} p(x_i, z; \Theta)}{p(x_i, y_i; \Theta)} = \text{const}(y_i),$$

- а значит неравенство Йенсена стало равенством!

$$L(\Theta) = \sum_{i=1}^N \mathbb{E}_{y_i} \left[\ln \frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} \right]$$

- EM-алгоритм (англ. *expectation-maximization*) итеративно максимизирует нижнюю оценку на функцию правдоподобия.

$t \leftarrow 0$

repeat

for $i = 1 \dots N$ **do**

$$Q_i^{(t)}(y_i) \leftarrow P(y_i | x_i; \Theta^{(t)})$$

▷ E-шаг

end for

$$\Theta^{(t+1)} \leftarrow \arg \max_{\Theta} \sum_{i=1}^N \mathbb{E}_{y_i} \left[\ln \frac{p(x_i, y_i; \Theta^{(t)})}{Q_i^{(t)}(y_i)} \right]$$

▷ M-шаг

$t \leftarrow t + 1$

until convergence

- Для работы алгоритму требуется начальное приближение параметров $\Theta^{(0)}$. Как можно его построить?

- Хочется верить, что между итерациями EM-алгоритма правдоподобие $L(\Theta)$ не уменьшается.
- Пусть $\Theta^{(t)}$ и $\Theta^{(t+1)}$ – оптимальные параметры для t -й и $t + 1$ -й итерации соответственно, а $Q_i^{(t)}$ – распределение y_i , вычисленное на t -й итерации, тогда:

$$L(\Theta^{(t+1)}) \geq \sum_{i=1}^N \mathbb{E}_{y_i} \left[\ln \frac{p(x_i, y_i; \Theta^{(t+1)})}{Q_i^{(t)}(y_i)} \right] \quad (1)$$

$$\geq \sum_{i=1}^N \mathbb{E}_{y_i} \left[\ln \frac{p(x_i, y_i; \Theta^{(t)})}{Q_i^{(t)}(y_i)} \right] = L(\Theta^{(t)}) \quad (2)$$

- Неравенство (1) верно для любых Q_i и Θ , см. слайд 7, почему, а (2) верно в силу того, что $\Theta^{(t+1)} = \arg \max_{\Theta} L(\Theta^{(t)})$.

- Напоминание:

$$Q_i(y_i) = P(y_i|x_i; \Theta) = \frac{p(x_i, y_i; \Theta)}{\sum_{y \in Y} p(x_i, y; \Theta)} \propto p(x_i, y_i; \Theta)$$

- То есть для мультиномиального наивного Байесовского классификатора:

$$\begin{aligned} Q_i(y_i) &\propto P_{y_i} p(x_i|y_i; \Theta) = P_{y_i} \text{Mul}(x_i; n, \vec{\theta}_{y_i}) \\ &= P_{y_i} \binom{n_i}{x_{i1}, \dots, x_{i|V|}} \theta_{y_i1}^{x_{i1}} \dots \theta_{y_i|V|}^{x_{i|V|}} \\ &\propto P_{y_i} \theta_{y_i1}^{x_{i1}} \dots \theta_{y_i|V|}^{x_{i|V|}}, \end{aligned}$$

где $n_i = \sum_{d=1}^{|V|} x_{id}$ — количество слов в сообщении.

- Хочется $\arg \max_{\Theta} L(\Theta)$, но как?

$$\begin{aligned} L(\Theta) &= \sum_{i=1}^N \mathbb{E}_{y_i} \left[\ln \frac{p(x_i, y_i; \Theta)}{Q_i(y_i)} \right] \\ &= \sum_{i=1}^N \sum_{y_i \in Y} Q_i(y_i) \ln \frac{P_y \text{Mul}(x_i|y; n, \vec{\theta}_{y_i})}{Q_i(y_i)} \end{aligned}$$

- Теперь видно, что $L(\Theta)$ можно оптимизировать аналитически,
- но сперва воспользуемся методом множителей Лагранжа, чтобы учесть ограничение на сумму вероятностей θ_{yd} :

$$\mathcal{L}(\Theta) = L(\Theta) + \sum_{y \in Y} \lambda_y \left(1 - \sum_{d=1}^{|V|} \theta_{yd} \right) \rightarrow \max_{\Theta}$$

Назад к наивному Байесовскому классификатору: M-шаг

- В качестве примера оптимизируем $\mathcal{L}(\Theta)$ относительно θ_{yd} :

$$\frac{\partial}{\partial \theta_{yd}} \mathcal{L}(\Theta) = \frac{1}{\theta_{yd}} \sum_{i=1}^N Q_i(y) x_{id} - \lambda_y \equiv 0$$

$$\Rightarrow \theta_{yd} = \frac{1}{\lambda_y} \sum_{i=1}^N Q_i(y) x_{id}$$

$$\frac{\partial}{\partial \lambda_y} \mathcal{L}(\Theta) = 1 - \sum_{d=1}^{|V|} \theta_{yd} = 1 - \frac{1}{\lambda_y} \sum_{d=1}^{|V|} \sum_{i=1}^N Q_i(y) x_{id} \equiv 0$$

$$\Rightarrow \lambda_y = \sum_{d=1}^{|V|} \sum_{i=1}^N Q_i(y) x_{id}$$

- Получили:

$$\hat{\theta}_{yd} \propto \sum_{i=1}^N Q_i(y) x_{id}$$

EM-алгоритм для наивного Байесовского классификатора

- EM-алгоритм для мультиномиального варианта классификатора заключается в повторении до сходимости двух шагов:

- Е-шаг:

$$Q_i(y_i) \propto P_{y_i} \theta_{y_i1}^{x_{i1}} \dots \theta_{y_i|V|}^{x_{i|V|}}$$

- М-шаг:

$$\hat{P}_y = \frac{\sum_{i=1}^N Q_i(y)}{N} \quad \hat{\theta}_{yd} = \frac{\sum_{i=1}^N Q_i(y) x_{id}}{\sum_{d=1}^{|V|} \sum_{i=1}^N Q_i(y) x_{id}}$$

- Полученные оценки эквивалентны ОМП, если положить $Q_i(y) = [y_i = y]$, где y_i известны из обучающей выборки.

Вопрос

Какие проблемы, на ваш взгляд, есть у получившегося алгоритма?

- EM-алгоритм не гарантирует сходимость к ОМП :-)
- Сходимость алгоритма может зависеть от выбора начального приближения параметров. Какие варианты?
 - Делать несколько запусков со случайными параметрами, обучаться до сходимости и выбирать параметры, максимизирующие правдоподобие.
 - Аналогично предыдущему, но обучаться только фиксированное число итераций.
 - Кластеризовывать объекты непараметрическим алгоритмом и оценивать параметры по получившимся кластерам.
- Критерий сходимости можно определять по-разному и получать разные результаты, например:
 - по параметрам,
 - по изменению правдоподобия,
 - по количеству итераций алгоритма.

- Функция правдоподобия $L(\Theta)$ инвариантна относительно перестановки меток классов, то есть, если $\sigma : Y \rightarrow Y$ – перестановка, то:

$$L(\Theta(Y)) = L(\Theta(\sigma(Y)))$$

- Чтобы разрешить неоднозначности, можно воспользоваться знаниями из предметной области, например: «Маша пишет чаще, чем Саша, поэтому нам нужна перестановка, в которой $P_Y > P_S$ »,
- или использовать небольшую контрольную выборку.
- Для случая двух классов хватит одного примера (x_l, y_l) .
Кстати, **почему?**

- EM-алгоритм – это не совсем алгоритм, а скорее способ находить ОМП параметров при наличии “отсутствующих” данных.
- Суть алгоритма в итеративной оптимизации нижней оценки на функцию правдоподобия.
- Мы поговорили
 - о том, почему EM-алгоритм “работает”,
 - применили его к мультиномиальному наивному Байесовскому классификатору,
 - обсудили основные проблемы, возникающие при использовании EM-алгоритма, и их возможные “решения”.

Последовательное

- Наивный Байесовский классификатор предполагает, что классы всех наблюдений **независимы**, то есть:

$$P(\vec{y}) = \prod_{i=1}^N P(y_i)$$

- Для задачи определения авторства естественно предположить, что между авторами последовательных писем есть зависимость.
- Удобной (с точки зрения обучения) формой зависимости является Марковская цепь порядка k :

$$P(\vec{y}) = \prod_{i=1}^N P(y_i | y_{i-1}, y_{i-2}, \dots, y_{i-k})$$

На практике чаще всего рассматривают случай $k = 1$.

- Чтобы сделать наш классификатор Марковским, необходимо параметризовать его вероятностями переходов P_{ys} и включить их в вероятность \vec{y} :

$$P(\vec{y}) = P_{y_1} \prod_{i=2}^N P_{y_{i-1}y_i}$$

- Таким образом, для обучения классификатора необходимо оценить:
 - P_y — априорные вероятности классов,
 - P_{ys} — вероятности перехода между классами,
 - θ_{yd} — вероятности слов для функции правдоподобия.

Вопрос

Как будем оценивать? И, кстати, почему **скрытый**?

Правдоподобие скрытого Марковского [...] классификатора

- Выпишем функцию **полного** правдоподобия

$$L(\Theta, \vec{y}) = \underbrace{\ln P_{y_1} + \sum_{i=2}^N \ln P_{y_{i-1}y_i}}_{\ln P(\vec{y})} + \underbrace{\sum_{i=1}^N p(x_i|y_i; \Theta)}_{\ln p(\vec{x}|\vec{y})}$$

- и подставим её в функцию правдоподобия:

$$\begin{aligned} L(\Theta) &= \mathbb{E}_{\vec{y}} [L(\Theta, \vec{y}) - \ln Q(\vec{y})] \\ &= \mathbb{E}_{\vec{y}} [\ln P_{y_1}] + \sum_{i=2}^N \mathbb{E}_{\vec{y}} [\ln P_{y_{i-1}y_i}] + \sum_{i=1}^N p(x_i|y_i; \Theta) \\ &\rightarrow \max_{\Theta} \end{aligned}$$

- Ура, $L(\Theta)$ выглядит как то, что несложно оптимизировать аналитически!

- Максимизировать $L(\Theta)$ напрямую нельзя, потому что не все значения параметров нам подходят:

$$\sum_{y \in Y} P_y = 1 \quad \sum_{s \in Y} P_{ys} = 1 \quad \sum_{d=1}^{|V|} \theta_{yd} = 1$$

- В очередной раз воспользуемся методом множителей Лагранжа:

$$\begin{aligned} \mathcal{L}(\Theta) &= L(\Theta) + \lambda(1 - \sum_{y \in Y} P_y) \\ &+ \sum_{y \in Y} \kappa_y(1 - \sum_{s \in Y} P_{ys}) \\ &+ \sum_{y \in Y} \delta_y(1 - \sum_{d=1}^{|V|} \theta_{yd}) \end{aligned}$$

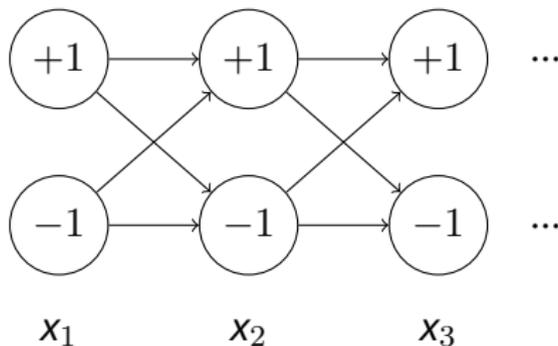
- Остальное – дело техники (и внимательности).

- M-шаг *aka* алгоритм Баума-Велша:

$$\hat{P}_y = \frac{\sum_{i=1}^N Q_i(y)}{N} \quad \hat{P}_{ys} = \frac{\sum_{i=1}^N Q_i(y_{i-1}, y_i)}{\sum_{i=1}^N Q_i(y)} \quad \hat{\theta}_{yd} \propto \sum_{i=1}^N Q_i(y) x_{id},$$

- Таким образом, в E-шаге нам необходимо вычислить не только $Q_i(y)$, но и $Q_i(y_{i-1}, y_i)$.
- Идея: представим $Q_i(y_i)$ в виде произведения двух вспомогательных переменных:

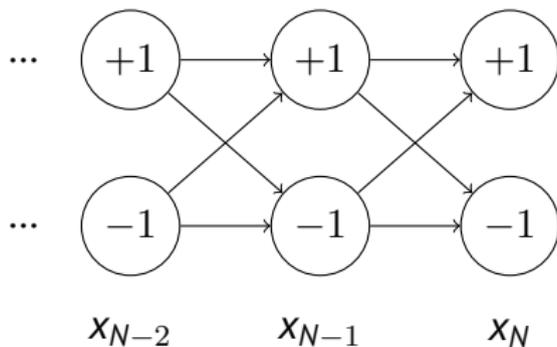
$$Q_i(y_i) \propto \alpha_y(i) \beta_y(i) \quad \begin{aligned} \alpha_y(i) &\doteq P(x_1, \dots, x_i, y_i = y) \\ \beta_y(i) &\doteq P(x_{i+1}, \dots, x_N | y_i = y) \end{aligned}$$



$$\alpha_y(i) \doteq P(x_1, \dots, x_i, y_i = y)$$

$$\alpha_y(1) = P_y p(x_1 | y; \Theta)$$

$$\alpha_y(i) = \left(\sum_{s \in Y} \alpha_s(i-1) P_{sy} \right) p(x_i | y; \Theta)$$



$$\beta_{\mathbf{y}}(i) \doteq P(x_{i+1}, \dots, x_N | y_i = y)$$

$$\beta_{\mathbf{y}}(N) = 1$$

$$\beta_{\mathbf{y}}(i) = \sum_{s \in Y} A_{ys} p(x_{i+1} | s; \Theta) \beta_s(i+1)$$

- $Q_i(y_i)$ можно посчитать с помощью алгоритмов прямого и обратного хода:

$$Q_i(y_i = y) = \frac{\alpha_y(i)\beta_y(i)}{\sum_{s \in Y} \alpha_y(i)\beta_y(i)}$$

- Оставшуюся вероятность $Q_i(y_{i-1}, y_i)$ тоже можно выразить в терминах $\alpha_y(i)$ и $\beta_y(i)$:

$$Q_i(y_{i-1} = y, y_i = s) \propto \alpha_y(i-1)A_{ys}p(x_i|s; \Theta)\beta_s(i)$$

Вопрос

Какая константа нормализации в выражении для $Q_i(y_{i-1}, y_i)$?

- До сих пор мы обсуждали классификацию методом максимума апостериорной вероятности

$$y_i^{\text{MAP}} = \arg \max_{y \in Y} P(y|x_i; \Theta),$$

- но существует и альтернативный подход, который ищет вектор \vec{y} , максимизирующий правдоподобие:

$$\vec{y}^{\text{MLE}} = \arg \max_{\vec{y} \in Y^N} L(\vec{x}, \vec{y}; \Theta)$$

- Для наивного Байесовского классификатора оба подхода эквивалентны, но после перехода к скрытому Марковскому [...] классификатору эквивалентность теряется.

Вопрос

Какой из методов лучше подходит для задачи определения авторства?

- Марковское предположение позволяет добавить в классификатор удобную для вычисления форму зависимости.
- Мы обсудили
 - применение EM-алгоритма для обучение скрытого Марковского [...] классификатора,
 - алгоритмы прямого и обратного хода, вычисляющие необходимые для M-шага значения $Q_i(y_i)$ и $Q_i(y_{i-1}, y_i)$,
 - два подхода к классификации при наличии Марковского предположения.

Очевидное

- Обучение вероятностных классификаторов подразумевает операции с большим количеством вероятностей,
- большинство существующих языков программирования реализуют вещественные числа в соответствии со стандартом IEEE-754²,
- поэтому, чтобы не потерять точность, рекомендуется проводить все операции в логарифмах:
 - умножение, возведение в степень, деление

$$\ln(ab) = \ln a + \ln b \quad \ln(a^b) = b \ln a \quad \ln \frac{a}{b} = \ln a - \ln b$$

- сложение

$$\ln(a + b) = ??? \quad \ln \sum_{i=1}^N a_i = ???$$

²Кстати, чем отличаются стандартные функции \log и \log_{1p} ?

- Пусть $a > b$, тогда посчитать логарифм суммы можно так:

$$\ln(a + b) = \ln\left(a\left(1 + \frac{b}{a}\right)\right) = \ln a + \ln\left(1 + \frac{b}{a}\right),$$

- а если a и b представимы в виде e^z :

$$\ln(e^x + e^y) = x + \ln(1 + e^{y-x})$$

- и для произвольного количества слагаемых:

$$x_m \doteq \max_{i=1}^N x_i \quad \ln \sum_{i=1}^N e^{x_i} = x_m + \ln \sum_{i=1}^N e^{x_i - x_m}$$

- На первый взгляд может показаться, что логарифмы сумм – вещь бесполезная, но вспомним алгоритм прямого хода:

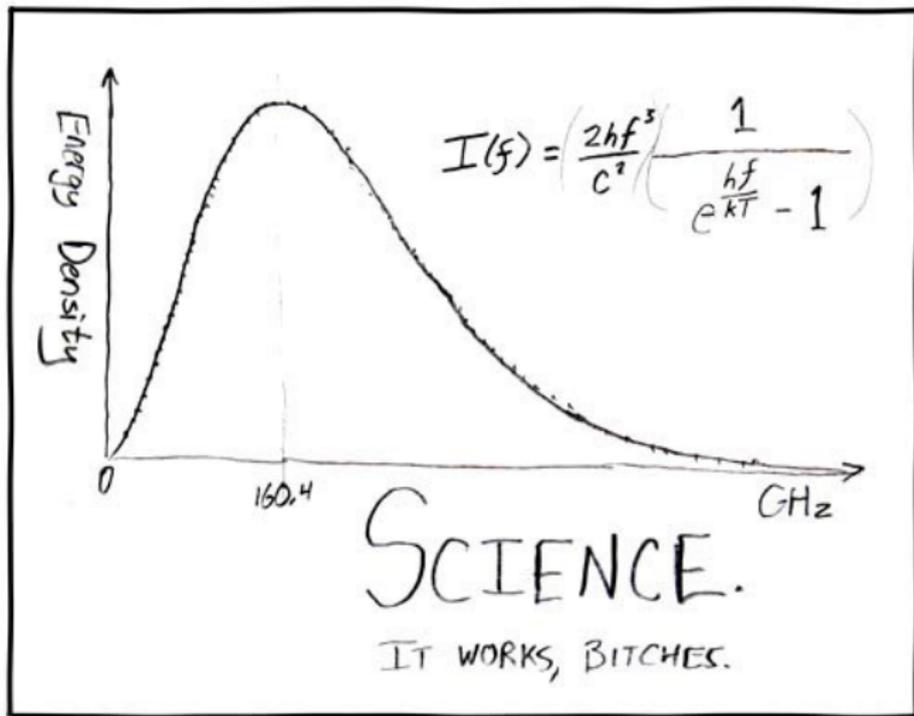
$$\alpha_y(1) = P_y p(x_1|y; \Theta)$$

$$\alpha_y(i) = \left(\sum_{s \in Y} \alpha_s(i-1) P_{sy} \right) p(x_i|y; \Theta)$$

- и прологарифмируем $\alpha_y(i)$:

$$\begin{aligned} \ln \alpha_y(i) &= \ln \sum_{s \in Y} \alpha_s(i-1) P_{sy} + \ln p(x_i|y; \Theta) \\ &= \ln \sum_{s \in Y} e^{\ln \alpha_s(i-1) + \ln P_{sy}} + \ln p(x_i|y; \Theta) \end{aligned}$$

- QED.



<https://xkcd.com/54>