Списочный подход

Распределение на перестановках

Большинство алгоритмов listwise ранжирования пытаются моделировать распределение перестановок $\pi : \{0, \ldots, n\} \to \{0, \ldots, n\}$, которое индуцировано набором скоров ранжирующей функции $(s_1, \ldots, s_n)$. Рассмотрим распределение Plackett-Luce. Пусть $\phi : \mathbb{R} \to \mathbb{R}_+$ строго возрастающая монотонная функция, которая принимает только положительные значения. Тогда вероятность перестановки π при скорах s задаётся как:

$$\mathbb{P}_s(\pi) = \prod_{j=1}^{n} \frac{\phi(s_{\pi(j)})}{\sum_{k=j}^{n} \phi(s_{\pi(k)})}$$

Это распределение моделирует процесс, в котором мы последовательно делаем выбор очередного элемента пропорционально весам оставшихся элементов $\phi(s_j)$. В качестве функции ϕ обычно берут $\phi(s) = e^s$:

$$\mathbb{P}_s(\pi) = \prod_{j=1}^{n} \frac{e^{s_{\pi(j)}}}{\sum_{k=j}^{n} e^{s_{\pi(k)}}}$$

Для наглядности рассмотрим пример с тремя объектами $[1, 2, 3]$, и соответствующими им скорами $s = (s_1, s_2, s_3)$. Вероятности перестановок $\pi = \langle 1, 2, 3 \rangle$и $\pi' = \langle 3, 1, 2 \rangle$будут считаться следующим образом:

$$\mathbb{P}_s(\pi) = \frac{e^{s_1}}{e^{s_1} + e^{s_2} + e^{s_3}} \cdot \frac{e^{s_2}}{e^{s_2} + e^{s_3}} \cdot \frac{e^{s_3}}{e^{s_3}}$$
$$\mathbb{P}_s(\pi') = \frac{e^{s_3}}{e^{s_1} + e^{s_2} + e^{s_3}} \cdot \frac{e^{s_1}}{e^{s_1} + e^{s_2}} \cdot \frac{e^{s_2}}{e^{s_2}}$$

Так, например, будут распределены перестановки, генерируемые функцией np.random.choice без возвращения:

In [60]:
```
scores = np.array([0.39, -0.95, 0.29, 0., -0.3, -0.97, -0.61, 0.82, -0.3, -0.77])
probs = np.exp(scores)
probs /= probs.sum()
print(probs)
```

```
[0.15817339 0.04141702 0.1431212  0.10709238 0.07933599 0.0405969
 0.05818874 0.24315323 0.07933599 0.04958517]
```

In [61]:
```
np.random.choice(probs.shape[0], size=probs.shape[0], replace=False, p=probs)
```

Out[61]:
```
array([7, 2, 0, 8, 5, 3, 1, 4, 9, 6])
```

Имея распределения, индуцированные известными оценками релевантности $\mathbb{P}_{y^{(i)}}(\pi)$, и распределения, индуцированные скорами модели $\mathbb{P}_{s^{(i)}}(\pi)$, естественно предложить кросс-энтропийную фукнцию потерь:

$$\mathcal{L}(\{s^{(i)}, y^{(i)}\}_{i=1}^{m}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{\pi\in\Omega_{n^{(i)}}} \mathbb{P}_{y^{(i)}}(\pi)\log \mathbb{P}_{s^{(i)}}(\pi)$$

Однако для каждого запроса $q^{(i)}$ в такой функции потерь будет $n^{(i)}!$ слагаемых, что практически неприменимо для любых разумных значений $n^{(i)}$.

Авторы алгоритма ListNet (https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2007-40.pdf) предлагают следующее упрощение модели. Давайте смотреть не на вероятность каждой перестановки, а только на вероятность каждого документа j быть на первом месте в ранжировании:

$$\mathbb{P}_s(j) = \sum_{\pi(j)=1,\ \pi\in\Omega_n} \mathbb{P}_s(\pi)$$

Оказывается, что такую вероятность легко посчитать на практике:

$$\mathbb{P}_s(j) = \frac{e^{s_j}}{\sum_{k=1}^{n} e^{s_k}}$$

Имея известное целевое распределение top-1 вероятностей $\mathbb{P}_{y^{(i)}}(j)$ и возвращаемое моделью $\mathbb{P}_{s^{(i)}}(j)$, функция потерь превращается в:

$$\mathcal{L}\big(\{s^{(i)}, y^{(i)}\}_{i=1}^m\big) = -\frac{1}{m}\sum_{i=1}^m \sum_{j=1}^{n^{(i)}} \mathbb{P}_{y^{(i)}}(j) \log \mathbb{P}_{s^{(i)}}(j) =$$

$$= \frac{1}{m}\sum_{i=1}^m \Big[\log\Big(\sum_{j=1}^{n^{(i)}} e^{s_j^{(i)}}\Big) - \frac{\sum_{j=1}^{n^{(i)}} s_j^{(i)} e^{y_j^{(i)}}}{\sum_{j=1}^{n^{(i)}} e^{y_j^{(i)}}}\Big]$$

Посчитаем её производные:

$$\frac{\partial \mathcal{L}}{\partial s_j^{(i)}} = \frac{1}{m}\Big(\frac{e^{s_j^{(i)}}}{\sum_{k=1}^{n^{(i)}} e^{s_k^{(i)}}} - \frac{e^{y_j^{(i)}}}{\sum_{k=1}^{n^{(i)}} e^{y_k^{(i)}}}\Big) = \frac{1}{m}(\mathbb{P}_{s^{(i)}}(j) - \mathbb{P}_{y^{(i)}}(j))$$

$$\frac{\partial^2 \mathcal{L}}{\partial (s_j^{(i)})^2} = \frac{1}{m}\mathbb{P}_{s^{(i)}}(j)(1 - \mathbb{P}_{s^{(i)}}(j))$$

$$\mathbb{P}(\pi^{*(i)}) = \prod_{j=1}^{n^{(i)}} \frac{e^{s^{(i)}(\pi^{*(i)}(j))}}{\sum_{k=j}^{n^{(i)}} e^{s^{(i)}(\pi^{*(i)}(k))}}$$

Другой вариант, предложенный авторами алгоритма ListMLE (http://icml2008.cs.helsinki.fi/papers/167.pdf) заключается в том, что для каждого запроса мы выбираем какую-нибудь перестановку $\pi^{(i)}$, и максимизируем её вероятность:

$$\mathcal{L}(\{s^{(i)}, \pi^{(i)}\}_{i=1}^m) = -\frac{1}{m}\sum_{i=1}^m \log \mathbb{P}_{s^{(i)}}(\pi^{(i)}) =$$

$$= -\frac{1}{m}\sum_{i=1}^m \Big(\big(\sum_{j=1}^{n^{(i)}} s_j^{(i)}\big) - \sum_{j=1}^{n^{(i)}} \log\big(\sum_{k=j}^{n^{(i)}} e^{s^{(i)}(\pi(k))}\big)\Big)$$

Если в качестве перестановки $\pi^{(i)}$ брать перестановку идеального ранжирования по таргетам $y^{(i)}$, то мы получим алгоритм ListMLE. Если же на каждой итерации сэмплировать эту перестановку из распределения Plackett-Luce $\pi^{(i)} \sim \mathbb{P}_{y^{(i)}}$, то мы получим алгоритм ListPL, предложенный в https://arxiv.org/abs/1707.07493.