Softmaxの最適化問題の解としての表現

あるベクトル $\boldsymbol{x} = (x_1, x_2, \dots, x_K)^\top$が与えられる。この時、

$$\max_{\boldsymbol{y} \in \Delta^{\mathrm{K}-1}} \boldsymbol{x}^\top \boldsymbol{y} - \tau \sum_{k=1}^{K} y_k \log y_k$$

の解は、softmax関数になる

$$y_k^* = \frac{\exp\left(\frac{x_k}{\tau}\right)}{\sum_k \exp\left(\frac{x_k}{\tau}\right)}$$

今欲しいもの：制約付きの確率ベクトル

やりたいこと

商品を割り当てる余地を表すBudgetベクトル $\boldsymbol{c} \in [0,1]^K$ が与えられたときに、
Budgetの範囲に入っている確率ベクトルを、数理的にそれっぽい方法で求めたい

$$y_k \in [0, c_k], \sum_k y_k = 1$$

アイデア

前述のsoftmaxの最適化問題に、Budgetベクトルの制約を付ける

$$\max_{\boldsymbol{y} \in \Delta^{K-1}, y_k \le c_k \ \forall k} \boldsymbol{x}^\top \boldsymbol{y} - \tau \sum_{k=1}^{K} y_k \log y_k$$

最適化問題の解

制約付きSoftmax（わかりやすさのため、minに変更）：

$$\min_{\boldsymbol{y}\in\Delta^{K-1}, y_k \le c_k\ \forall k} -\boldsymbol{x}^\top \boldsymbol{y} + \tau \sum_{k=1}^{K} y_k \log y_k$$

ラグランジュ関数：

$$\boldsymbol{L}(\boldsymbol{y}, \alpha, \boldsymbol{\beta}; \boldsymbol{x}, \tau) = -\boldsymbol{x}^\top \boldsymbol{y} + \tau \sum_{k=1}^{K} y_k \log y_k + \alpha\left(\sum_k y_k - 1\right) + \sum_k \beta_k (y_k - c_k)$$

$\boldsymbol{y}$に関して微分して最適性条件を考える

$$\frac{\partial L}{\partial y_k} = -x_k + \tau(1 + \log y_k) + \alpha + \beta_k$$

最適化問題の解：softmaxの亜種

$\frac{\partial L}{\partial y_k} = 0$の解：

$$-x_k + \tau(1 + \log y_k) + \alpha + \beta_k = 0$$
$$\to y_k = \exp([x_k - \alpha - \beta_k - 1]/\tau) = \exp\big((x_k - \beta_k)/\tau\big)/z$$

- $z \coloneqq \exp(-[-\alpha - 1]/\tau)$

確率ベクトルの条件 $\sum_k y_k = 1$より、zについては以下が成り立つ：

$$z = \sum_k \exp\big((x_k - \beta_k)/\tau\big)$$

よって、解は$\boldsymbol{x}$から何らかのベクトル$\boldsymbol{\beta}$を引いて、softmaxを噛ませることになる

$$y_k^* = \mathrm{softmax}(\boldsymbol{x} - \boldsymbol{\beta})$$

最適化問題の解：$\boldsymbol{\beta}$について

KKT条件より以下が成り立つ：

$$\beta_k(y_k - c_k) = 0$$
$$\rightarrow \beta_k = 0 \ \text{ or } (\beta_k > 0 \text{ and } y_k = c_k)$$

- つまり、β_kは (i)"0" または (ii) "引いてsoftmaxに噛ませた結果c_kになるような値"

$K_B \coloneqq \{k \in [K]: y_k^* = c_k\}$とすると

$$y_k^* = \frac{\exp(x_k/\tau)}{z} \quad \forall k \notin K_B\,, \qquad \sum_{k \notin K_B} y_k^* = 1 - \sum_{k \in K_B} c_k$$

$$\rightarrow z = \frac{\sum_{k \notin K_B} \exp\left(\frac{x_k}{\tau}\right)}{1 - \sum_{k \in K_B} c_k}$$

- y_k^*は (i) c_kに押し込める または (ii) c_kになった要素以外でsoftmaxを計算して定数$(1 - \sum_{k \in K_B} c_k)$倍することで求められる、ともいえる
- K_Bが分かれば計算できそう

最適化問題の解の性質：K_Bについて

命題1 $i \in K_B$ かつ $c_j / \exp\left(\frac{x_j}{\tau}\right) < c_i / \exp\left(\frac{x_i}{\tau}\right)$ならば、$j \in K_B$

(証明)$j \notin K_B$を仮定して矛盾を示す。$j \notin K_B$の仮定から $y_j^* = \exp(x_j/\tau)/z < c_j$

また、命題の仮定から$c_j < \exp\left(\frac{x_j}{\tau}\right) \cdot c_i / \exp\left(\frac{x_i}{\tau}\right)$
この二つを合わせると、

$$\frac{\exp\left(\frac{x_j}{\tau}\right)}{z} < \frac{\exp\left(\frac{x_j}{\tau}\right) \cdot c_i}{\exp\left(\frac{x_i}{\tau}\right)} \rightarrow 1 < \frac{c_i \cdot z}{\exp\left(\frac{x_i}{\tau}\right)} = \exp\left(-\frac{\beta_i}{\tau}\right)$$

を得る。$i \in K_B$ の仮定より、$\beta_i > 0 \rightarrow \exp\left(-\frac{\beta_i}{\tau}\right) < 1$なので、矛盾（証明了）

ゆえに、$c_k / \exp\left(\frac{x_k}{\tau}\right)$で昇順でソートされているとすると、ある$k' \geq 0$までは$y_k^* = c_k$とbudgetに押し込まれ、$k'$以降は$y_k^* = \exp(x_k/\tau)/z$の形式になる

最適化問題の解：アルゴリズム

結局、以下の手続きで求められる

- $O(K \log K)$。工夫すれば$O(K)$でも（＝ソートせず）できそうな気はするが、多分実用上は遅くなる

Input $\boldsymbol{x}, \boldsymbol{c}, \tau$
$c_k / \exp\left(\frac{x_k}{\tau}\right)$で昇順でソート;
$r \leftarrow \sum_k \exp(x_k/\tau)$, $s \leftarrow 1$; # 最終的に、$s := 1 - \sum_{k \in K_B} c_k$, $r := \sum_{k \notin K_B} \exp\left(\frac{x_k}{\tau}\right)$
For $k = 1, \ldots, K$:
　　$z \leftarrow r/s$;
　　If $\exp(x_k/\tau)\,/z > c_k$ # $k \notin K_B$と考えたら矛盾する場合
　　　　$y_k \leftarrow c_k$;
　　　　$r \leftarrow r - \exp\left(\frac{x_k}{\tau}\right)$;
　　　　$s \leftarrow s - c_k$;
　　else:
　　　　$y_k \leftarrow \exp(x_k/\tau)\,/z$; # 矛盾しない場合。一度でもこのelseに入れば残りはずっとここに入る
Output $\boldsymbol{y}$

最適化問題の勾配

ここまで考えてきた最適化問題の解を求める問題・手続きを、**bcsoftmax** (budget/box-constrained softmax)と呼ぶことにする：

$$\underset{\boldsymbol{y}\in\Delta^{K-1}, y_k\le c_k\ \forall k}{\operatorname{argmax}} \quad \boldsymbol{x}^\top \boldsymbol{y} - \tau \sum_{k=1}^{K} y_k \log y_k =: \mathrm{bcsoftmax}(\boldsymbol{x}, \boldsymbol{c}; \tau)$$

気になる点：bcsoftmaxの$\boldsymbol{x}$と$\boldsymbol{c}$に対する勾配

- NNで使う場合に勾配が計算できるとありがたい
- $\boldsymbol{c}$もNNの出力から計算される場合があるので

sparsemaxと似ていて、ほとんどいたるところで微分可能

- c_kに押し込まれるか否かが切り替わるような点では不可能

最適化問題のJacobian

Jacobianが計算できれば良い。まず、$\boldsymbol{x}$に関するJacobianを考える：

$$
\frac{\partial \mathrm{bcsoftmax}_i(\boldsymbol{x}, \boldsymbol{c}; \tau)}{\partial x_j}
= \begin{cases} 0, & j \in K_B \text{ or } i \in K_B \\ s\,\dfrac{\delta_{ij} \exp\left(\frac{x_i}{\tau}\right) r - \exp\left(\frac{x_i}{\tau}\right) \exp\left(\frac{x_j}{\tau}\right)}{\tau r^2}, & j \notin K_B \text{ and } i \notin K_B \end{cases}
$$

- $s \coloneqq 1 - \sum_{k \in K_B} c_k$, $r \coloneqq \sum_{k \notin K_B} \exp\left(\frac{x_k}{\tau}\right)$
- $i \in K_B$のときは$y_i^* = c_i$
- $j \in K_B$の時は分子分母にx_jは現れない

最適化問題のJacobian

次に、$\boldsymbol{c}$に関するJacobianを考える：

$$\frac{\partial \mathrm{bcsoftmax}_i(\boldsymbol{x}, \boldsymbol{c}; \tau)}{\partial c_j} = \begin{cases} 0, & j \notin K_B \\ \delta_{ij}, & j \in K_B \text{ and } i \in K_B \\ -\dfrac{\exp(\, x_i/\tau)}{r}, & j \in K_B \text{ and } i \notin K_B \end{cases}$$

- $j \notin K_B$の場合はc_jは現れない
- $i \in K_B$の場合は$y_i^* = c_i$
- $i \notin K_B$の場合は$y_i^* = s \cdot \exp(x_i/\tau)\,/r$