We briefly describe the modified scoring function combined with a tensor component and the corresponding online learning update. For more details, see the code and Lei et al. [2014].

Scoring Function For a given sentence x, let $\phi(x, y)$ be the sparse vector representation of the 1st-, 3rd-order and global features (typically used by MST, Turbo, the sampling-based parsers, etc.). In addition, let ϕ_m be a smaller feature vector defined on the word m, and $\phi_{y(m),m}$ be a smaller feature vector defined on the arc $y(m) \rightarrow m$ (we suppressed their dependence on x). The combined scoring is defined as,

$$\begin{aligned} S(x,y) &= (1-\gamma)S_{tensor}(x,y) + \gamma S_{normal}(x,y) \\ &= (1-\gamma)\sum_{m}\sum_{i=r}^{r}\left[U\phi_{y(m)}\right]_i \left[V\phi_m\right]_i \left[W\phi_{y(m),m}\right]_i \\ &+ \gamma\theta^T\phi(x,y) \end{aligned}$$

where $\gamma \in [0,1]$ controls the weight of the two scoring components and $\theta \in \mathbb{R}^L$ is the traditional parameter vector, $U, V \in \mathbb{R}^{r\times k}$, $W \in \mathbb{R}^{r\times d}$ are three rank-r parameter matrices that factorize a 3-way low-rank tensor. The learning problem is to estimate parameter values for (θ, U, V, W).

Online Update The passive-aggressive online updates (tailored for the combined scoring) are carried in an alternative fashion, updating a pair of parameter sets (θ, U), (θ, V) and (θ, W) at a time. The updates (say for θ and U) are in the following form,

$$\begin{aligned} \theta^{(t+1)} &= \theta(t) + \alpha^{(t)}\gamma d\theta \\ U^{(t+1)} &= U^{(t)} + \alpha^{(t)}(1-\gamma)du \end{aligned}$$

Here $\alpha^{(t)}$ is the step size, $d\theta$ and du are the discrepancies between the gold tree $\hat{y}$ and the violation tree $\tilde{y}$:

$$\begin{aligned} d\theta &= \phi(\hat{x},\hat{y}) - \phi(\hat{x},\tilde{y}) \\ du &= \sum_m \left[(V\phi_m)\odot\left(W\phi_{\hat{y}(m),m}\right)\right]\otimes\phi_{\hat{y}(m)} \\ &\quad - \sum_m \left[(V\phi_m)\odot\left(W\phi_{\tilde{y}(m),m}\right)\right]\otimes\phi_{\tilde{y}(m)} \end{aligned}$$

where $\odot$ is the Hadamard (element-wise) product and $\otimes$ is the tensor product.

References

Tao Lei, Yu Xin, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. Low-rank tensors for scoring dependency structures. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2014.