Do Lessons from Metric Learning Generalize to Image-Caption Retrieval?

Maurits Bleeker and Maarten de Rijke

University of Amsterdam, Amsterdam, The Netherlands
{m.j.r.bleeker, m.derijke}@uva.nl

This appendix has two sections, one devoted to a derivation of the gradient of SmoothAP w.r.t. q (Appendix A), and one devoted to reproducibility (Appendix B).

A Derivative of the gradient of SmoothAP w.r.t. q

A.1 Explanation of SmoothAP

The Average Precision metric represents the area under the precision-recall curve. Average Precision is a discrete metric and therefore can not be used directly as loss function for optimizing retrieval methods. The main intuition behind the SmoothAP [1] is to have a smooth, and therefore differentiable, approximation of the Average Precision metric. Using the notation introduced in Section **??**, the Average Precision metric is defined as follows:

$$AP_{\mathbf{q}} = \frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \frac{\mathcal{R}(i, \mathcal{S}_{\mathcal{P}}^{\mathbf{q}})}{\mathcal{R}(i, \mathcal{S}_{\Omega}^{\mathbf{q}})}, \tag{1}$$

where $\mathcal{R}(i,\mathcal{S})$ is defined as:

$$\mathcal{R}(i,\mathcal{S}) = 1 + \sum_{j \in \mathcal{S}, i \neq j} \mathbb{1}\{s_i - s_j < 0\}. \tag{2}$$

$\mathcal{R}(i,\mathcal{S})$ returns the rank of candidate i in a ranking over a set with candidates $\mathcal{S}$, given query $\mathbf{q}$. s_i is the similarity score between the query $\mathbf{q}$ and candidate $\mathbf{v}_i$. Similarly, s_j is similarity score between the query $\mathbf{q}$ and candidate $\mathbf{v}_j$. If $s_i - s_j$ is lower than 0, this indicates that candidate j is ranked higher than candidate i. By counting how many times $\mathbb{1}\{s_i - s_j < 0\}$ is true, the ranking of candidate i can be determined.

To simplify the computation and notation, we can introduce a matrix D, where D is defined as:

$$D = \begin{bmatrix} s_1 & \dots & s_m \\ \vdots & \ddots & \vdots \\ s_1 & \dots & s_m \end{bmatrix} - \begin{bmatrix} s_1 & \dots & s_1 \\ \vdots & \ddots & \vdots \\ s_m & \dots & s_m \end{bmatrix}, \tag{3}$$

where $D_{ij} = s_i - s_j$. In this case we have a candidate set $\mathcal{S}$ with m candidates. By using matrix D, we can rewrite Eq. 2 as follows:

$$\mathcal{R}(i,\mathcal{S}) = 1 + \sum_{j \in \mathcal{S}, i \neq j} \mathbb{1}\{D_{ij} > 0\}. \tag{4}$$

To make $\mathcal{R}(i,\mathcal{S})$, and thereby $AP_{\mathbf{q}}$, differentiable, the indicator function $\mathbb{1}$ is replaced by the smooth sigmoid function $\mathcal{G}(\cdot,\tau)$.

A.2 Derivative of the gradient of SmoothAP w.r.t. q

In this section, we give an analyses and derivation of the gradient of SmoothAP [1] w.r.t. query $\mathbf{q}$. We start with Eq. **??**, the definition of SmoothAP:

$$AP_{\mathbf{q}} = \frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|} \sum_{i \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}} \frac{1 + \sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} \mathcal{G}(D_{ij};\tau)}{1 + \sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} \mathcal{G}(D_{ij};\tau) + \sum_{j \in \mathcal{S}_{\mathcal{N}}^{\mathbf{q}}} \mathcal{G}(D_{ij};\tau)}. \tag{5}$$

Here, $\mathcal{G}$ is a smooth approximation of an indicator/step function:

$$\mathcal{G}(f(x);\tau) = \frac{1}{1+e^{-\frac{f(x)}{\tau}}}. \tag{6}$$

The derivative of $\mathcal{G}$ w.r.t. a function $f(x)$ has the following form:

$$\frac{\partial \mathcal{G}(f(x);\tau)}{\partial x} = \mathcal{G}(f(x);\tau)(1-\mathcal{G}(f(x);\tau)) \frac{1}{\tau} \frac{\partial f(x)}{\partial x}. \tag{7}$$

Note that $f(x)$ in the case of SmoothAP is D_{ij}:

$$D_{ij} = s_i - s_j = \mathbf{q}\mathbf{v}_i - \mathbf{q}\mathbf{v}_j, \tag{8}$$

where both $\mathbf{v}_i$, $\mathbf{v}_j$ and $\mathbf{q}$ are normalized on the unit-sphere. The gradient of D_{ij} w.r.t. query $\mathbf{q}$ has the following form:

$$\frac{\partial D_{ij}}{\partial \mathbf{q}} = \mathbf{v}_i - \mathbf{v}_j. \tag{9}$$

If we plug in D_{ij} into Eq. 7 and take the gradient w.r.t. query $\mathbf{q}$, we get

$$\begin{aligned} \frac{\partial \mathcal{G}(D_{ij};\tau)}{\partial \mathbf{q}} &= \mathcal{G}(D_{ij};\tau)(1-\mathcal{G}(D_{ij};\tau)) \frac{1}{\tau} (\mathbf{v}_i - \mathbf{v}_j) \\ &= sim(D_{ij},\tau)(\mathbf{v}_i - \mathbf{v}_j), \end{aligned} \tag{10}$$

where $sim(D_{ij},\tau)$ is a function that gives an indication of how close the similarity scores are of candidate i and j are w.r.t. query $\mathbf{q}$, scaled by τ:

$$sim(D_{ij},\tau) = \mathcal{G}(D_{ij};\tau)(1-\mathcal{G}(D_{ij};\tau)) \frac{1}{\tau}. \tag{11}$$

Now we define $\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})$ and $\mathcal{R}(i,\mathcal{S}_{\mathcal{P}}^{\mathbf{q}})$. $\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})$ gives the ranking of candidate i within the full candidate set $\mathcal{S}_{\Omega}^{\mathbf{q}}$. $\mathcal{R}(i,\mathcal{S}_{\mathcal{P}}^{\mathbf{q}})$. gives the rank of candidate i within the positive candidate set $\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}$:

$$\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}}) = \left(\overbrace{1 + \sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} \mathcal{G}(D_{ij};\tau)}^{A} + \overbrace{\sum_{j \in \mathcal{S}_{\mathcal{N}}^{\mathbf{q}}} \mathcal{G}(D_{ij};\tau)}^{C} \right) \tag{12}$$

$$\mathcal{R}(i,\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}) = \left(\overbrace{1 + \sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} \mathcal{G}(D_{ij};\tau)}^{A} \right). \tag{13}$$

The gradient of $\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})$ w.r.t. to $\mathbf{q}$ has the following form:

$$\frac{\partial \mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})}{\partial \mathbf{q}} = \left(\overbrace{\sum_{j \in \mathcal{S}_{\mathcal{P}}^{\mathbf{q}}, j \neq i} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j)}^{B} + \overbrace{\sum_{j \in \mathcal{S}_{\mathcal{N}}^{\mathbf{q}}} sim(D_{ij})(\mathbf{v}_i - \mathbf{v}_j)}^{D} \right). \tag{14}$$

Using all the definitions above, we can write the full gradient of $AP_{\mathbf{q}}$ w.r.t. $\mathbf{q}$:

$$\frac{\partial AP_{\mathbf{q}}}{\partial \mathbf{q}}$$

$$=\frac{1}{|\mathcal{S}_{P}^{\mathbf{q}}|}\sum_{i\in\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}}\frac{\mathcal{R}(i,\mathcal{S}_{\mathcal{P}}^{\mathbf{q}})\frac{\partial\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})}{\partial\mathbf{q}}-\mathcal{R}(i,\mathcal{S}_{\Omega}^{q})\left(\sum_{j\in\mathcal{S}_{\mathcal{P}}^{\mathbf{q}},j\neq i}sim(D_{ij})(\mathbf{v}_i-\mathbf{v}_j)\right)}{\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})^2} \tag{15}$$

$$=\frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|}\sum_{i\in\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}}\frac{1}{\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})^2}\left(\left(\overbrace{\mathcal{R}(i,\mathcal{S}_{\mathcal{P}}^{\mathbf{q}})}^{A}\overbrace{\frac{\partial\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})}{\partial\mathbf{q}}}^{B+D}\right)-\right.$$
$$\left.\left(\overbrace{\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})}^{A+C}\right)\left(\overbrace{\sum_{j\in\mathcal{S}_{\mathcal{P}}^{\mathbf{q}},j\neq i}sim(D_{ij})(\mathbf{v}_i-\mathbf{v}_j)}^{D}\right)\right). \tag{16}$$

When looking at Eq. 16 it becomes clear that we have a function in the following form $A(B+D)-(A+C)B$. This can be rewritten to: $AB+AD-AB-CB=AD-CB$. If we apply this to Eq. 16, we end up with the following form:

$$\frac{\partial AP_{\mathbf{q}}}{\partial\mathbf{q}}$$

$$=\frac{1}{|\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}|}\sum_{i\in\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}}\frac{1}{\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})^2}\left(\overbrace{\mathcal{R}(i,\mathcal{S}_{\mathcal{P}}^{\mathbf{q}})}^{A}\left(\overbrace{\sum_{j\in\mathcal{S}_{\mathcal{N}}^{\mathbf{q}}}sim(D_{ij})(\mathbf{v}_i-\mathbf{v}_j)}^{D}\right)-\right.$$
$$\left.\overbrace{\sum_{j\in\mathcal{S}_{\mathcal{N}}^{\mathbf{q}}}\mathcal{G}(D_{ij};\tau)}^{C}\left(\overbrace{\sum_{j\in\mathcal{S}_{\mathcal{P}}^{\mathbf{q}},j\neq i}sim(D_{ij})(\mathbf{v}_i-\mathbf{v}_j)}^{B}\right)\right) \tag{17}$$

$$=\frac{1}{|\mathcal{S}_{P}^{\mathbf{q}}|}\sum_{i\in\mathcal{S}_{\mathcal{P}}^{\mathbf{q}}}\frac{1}{\mathcal{R}(i,\mathcal{S}_{\Omega}^{\mathbf{q}})^2}\left(\mathcal{R}(i,\mathcal{S}_{\mathcal{P}}^{\mathbf{q}})\left(\sum_{j\in\mathcal{S}_{\mathcal{N}}^{q}}sim(D_{ij})(\mathbf{v}_i-\mathbf{v}_j)\right)-\right.$$
$$\left.(\mathcal{R}(i,\mathcal{S}_{\mathcal{N}}^{q})-1)\left(\sum_{j\in\mathcal{S}_{P}^{\mathbf{q}},j\neq i}sim(D_{ij})(\mathbf{v}_i-\mathbf{v}_j)\right)\right). \tag{18}$$

B Reproducibility

B.1 VSE++

The VSE++ [12] is an image-caption retrieval (ICR) method that uses two encoders that do not share parameters: an image and a caption encoder. For the image encoder, two CNN networks have been used in [12]: ResNet-152 [15] and VGG19 [37], where ResNet-152 yields the best evaluation performances. To reduce the computation time of the training process, we have decided to use ResNet-50 instead of ResNet-152. Some preliminary experiments have shown that the differences in evaluation score between ResNet-152 and 50 is relatively small, while ResNet-50 is faster to optimize. The ResNet network functions as a so-called backbone or feature extractor. On top of the ResNet network, a fully-connected layer is placed to map the extracted features to a multi-modal latent space. Only the weights of this fully-connected layer are optimized during training.

The caption encoder consists of a unidirectional GRU [10] encoder. The word embeddings for the text encoder are trained end-to-end (from scratch) with the rest of text encoder. The output of the last encoding step is the representation for the input caption. The output representations of both encoders are normalized on the unit sphere.

B.2 VSRN

VSRN [25] also consists of separate text and image encoder. For the image encoder VSRN [25] uses pre-computed features as input. These features have been generated by a Faster R-CNN [36] model, which uses ResNet-101 [15] as backbone, trained on the Visual Genomes dataset [23]. The feature map of the last convolutional layer serves as input representation for the next layer, each vector in this feature map represents a region in the input image. Next, a GCN [21] is used to enhance the input feature vectors with relation information between each region in the input image. Finally, a GRU is used to compute the global representation of the different region vectors. This is done by feeding the region representations one by one into the GRU encoder as a sequence. VSRN uses the same text encoder as VSE++.

To generate an extra training signal, a caption generator is added to the training process. This generator is optimized to reconstruct the input caption based on the visual region feature representations. We have decided to remove this caption decoder from the learning algorithm. The reason for this is that we focus on ICR only and we want to exclude any additional learning signal from the training process.

B.3 Implementation and optimization details

Both VSE++ and VSRN are optimized for 30 epochs on the same datasets [27, 42]. For VSE++, after 30 epochs of training, 15 epochs of additional fine-tune are applied where the backbone of the image encoder is also optimized. We do not apply this additional fine-tuning step due to the following two reasons: (1) We want to optimize VSE++ and VSRN for the same number of epochs, and (2) Our goal is not to have the best performing model, but rather to evaluate the impact of a loss function. Therefore, the weights of the feature extractor for the VSE++ image encoder are frozen during the entire training process in this work. All the other remaining implementation details and hyper-parameters in this work are similar to [12, 25].

Bibliography

[1] Brown A, Xie W, Kalogeiton V, Zisserman A (2020) Smooth-AP: Smoothing the path towards large-scale image retrieval. In: European Conference on Computer Vision (ECCV), Springer, pp 677–694

[10] Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:14091259

[12] Faghri F, Fleet DJ, Kiros JR, Fidler S (2018) VSE++: Improving visual-semantic embeddings with hard negatives. In: Proceedings of the British Machine Vision Conference (BMVC)

[15] He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778

[21] Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907

[23] Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li LJ, Shamma DA, et al. (2016) Visual genome: Connecting language and vision using crowdsourced dense image annotations. arXiv preprint arXiv:160207332

[25] Li K, Zhang Y, Li K, Li Y, Fu Y (2019) Visual semantic reasoning for image-text matching. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp 4654–4662

[27] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: Common objects in context. In: European Conference on Computer Vision, Springer, pp 740–755

[36] Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, vol 28, pp 91–99

[37] Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

[42] Young P, Lai A, Hodosh M, Hockenmaier J (2014) From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics 2:67–78