Batch-Incremental Triplet Sampling for Training Triplet Networks Using Bayesian Updating Theorem

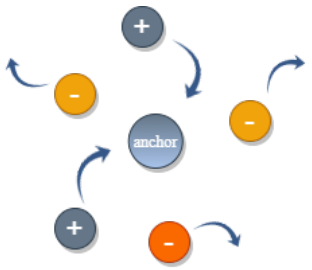
Milad Sikaroudi[1] Benyamin Ghojogh[2] Fakhri Karray[2] Mark Crowley[2] Hamid Reza Tizhoosh[1]

[1]Kimia Lab, University of Waterloo [2]Electrical and Computer Engineering Departemrnt, University of Waterloo

Explicit Metric Learning , Triplet/ NCA Losses

$$\text{Triplet loss: minimize} \sum_{i=1}^{b}\sum_{k=1}^{c-1}\sum_{\ell=1}^{c-1}\Big[m+\|\underline{x}_i-\underline{x}_k\|_2^2-\|\underline{x}_i-\underline{x}_\ell\|_2^2\Big]_+, \tag{1}$$

$$\text{NCA loss: minimize } -\sum_{i=1}^{b}\sum_{k=1}^{c-1}\ln\Big(\frac{\exp(-\|\underline{x}_i-\underline{x}_k\|_2^2)}{\sum_{\ell=1}^{c-1}\exp(-\|\underline{x}_i-\underline{x}_\ell\|_2^2)}\Big). \tag{2}$$

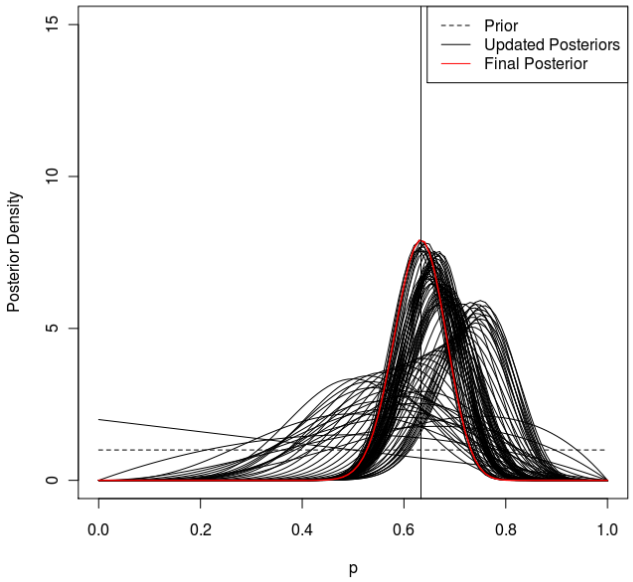


Triplet Sampling using Bayesian Updating Theorem

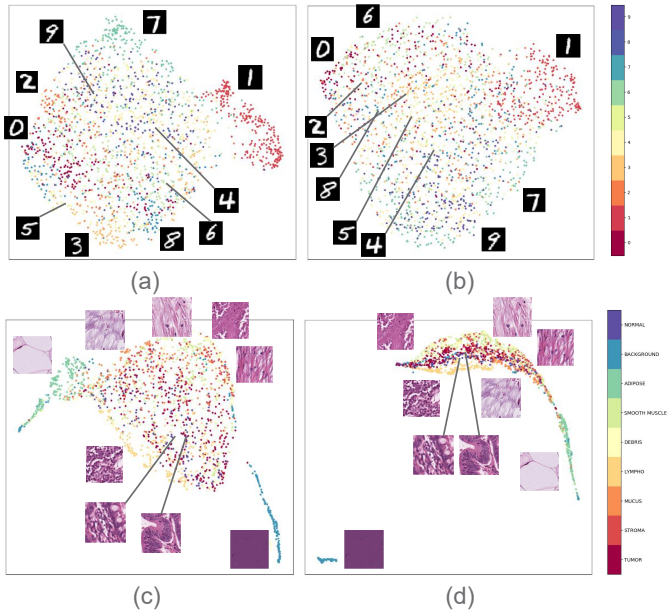
Procedure: TrainTripletNetwork($\{\underline{z}_i\}_{i=1}^n$, $\{y_i\}_{i=1}^n$)
Input: training data: $\{\underline{z}_i\}_{i=1}^n$, training labels: $\{y_i\}_{i=1}^n$
for *all required epochs* **do**
  for *all batches in epoch* **do**
    $\{\underline{x}_i\}_{i=1}^b \leftarrow$ Feed $\{\underline{z}_i\}_{i=1}^b$ to the triplet network
    for *class j from 1 to c* **do**
      if *it is first mini-batch* **then**
        $\underline{\mu}^{0,j} := (1/n')\sum_{i=1}^{n'} \underline{x}_i'^j$
        $\underline{\Sigma}^{0,j} := (1/n')\sum_{i=1}^{n'} (\underline{x}_i'^j - \underline{\mu}^{0,j})(\underline{x}_i'^j - \underline{\mu}^{0,j})^\top$
      else
        $\underline{\mu}'^j := (1/n')\sum_{i=1}^{n'} \underline{x}_i'^j$
        $\underline{\mu}^{0,j} := (n'\underline{\mu}'^j + n_0\underline{\mu}^{0,j})/(n'+n_0)$
        if $n'+n_0 > d+1$ **then**
          $\underline{\Upsilon} := n'\underline{\Sigma}'^j + n_0\underline{\Sigma}^{0,j} + \frac{n' n_0}{n'+n_0}(\underline{\mu}^{0,j} - \underline{\mu}'^j)(\underline{\mu}^{0,j} - \underline{\mu}'^j)^\top$
          $\underline{\Sigma}^{0,j} := \underline{\Upsilon}^{-1}/(n'+n_0-d-1)$
        else
          $\underline{\Sigma}^{0,j} := (1/n')\sum_{i=1}^{n'} (\underline{x}_i'^j - \underline{\mu}'^j)(\underline{x}_i'^j - \underline{\mu}'^j)^\top$
    for *instance i from 1 to b* **do**
      anchor $\leftarrow \underline{x}_i$
      for *class j from 1 to c* **do**
        if $j = y_i$ **then**
          Sample $(c-1)$ positive instances $\sim \mathcal{N}(\underline{\mu}^{0,j}, \underline{\Sigma}^{0,j})$
        else
          Sample a negative instance $\sim \mathcal{N}(\underline{\mu}^{0,j}, \underline{\Sigma}^{0,j})$
    Minimize the *triplet*/NCA loss with the $(b \times (c-1))$ triplets.

Algorithm 1: Dynamic Triplet Sampling with Bayesian Updating

Bayesian Updating Theorem



Results



2D visualization of test embeddings: (a) MNIST using BUT, (b) MNIST using BUNCA, (c) CRC using BUT, and (d) CRC using BUNCA.