The snowball principle for handwritten word-image retrieval

van Oosten, Jean-Paul

DOI:
10.33612/diss.160750597

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2021

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
Implementing an HMM framework from scratch is not trivial. The canonical paper by Rabiner (1989) contains all the theory necessary, but it may require some extra considerations to make implementation easier. We will give some of these considerations here. The approach used for implementing jpHMM is taken in part from A. Rahimi (2000).

Scaling forward and backward variables

The first issue to address is scaling the forward and backward variables $\alpha_t(j)$ and $\beta_t(j)$. The forward variable is the probability of the partial observation sequence up to time $t$ and being in state $S_j$ at time $t$, given the model $\lambda$: $\alpha_t(j) = P(O_1O_2\cdots O_t, q_t = S_j | \lambda)$. The backward variable is the probability of the partial observation sequence from time $t + 1$ to time $T$, given state $S_j$ at time $t$ and the model $\lambda$: $\beta_t(j) = P(O_{t+1}O_{t+2}\cdots O_T | q_t = S_j, \lambda)$.

These variables need to be scaled to avoid problems with floating point representations in code. Since the forward variable $\alpha_t(j)$ usually consists of many products of transition and observation probabilities, they tend to approach 0 quickly. On a computer, these variables are bound by a finite precision floating point representation.

A scaling can be applied to both $\alpha_t(j)$ and $\beta_t(j)$, to keep the calculations in range of a floating point representation. Rabiner proposes to use the scaling factor $c_t = \frac{1}{\sum_{i=1}^{N} \hat{a}_t(i)}$, which is independent of state. This means that $\sum_{i=1}^{N} \hat{a}_t(i) = 1$. Both $\alpha_t(j)$ and $\beta_t(j)$ are scaled with the same factor, $c_t$.

The recursion formulae defined by Rabiner are theoretically correct, but hard to use for implementation because it is unclear that one needs to use the scaled \( \hat{\alpha}_t(i) \) in the computation for \( c_{t+1} \). Rahimi therefore proposes the following computation steps:

\[
\begin{align*}
\bar{\alpha}_1(i) &= \alpha_1(i) \\
\bar{\alpha}_{t+1}(j) &= \sum_{i=1}^{N} \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \\
c_{t+1} &= \frac{1}{\sum_{i=1}^{N} \bar{\alpha}_{t+1}(i)} \\
\hat{\alpha}_{t+1}(i) &= c_{t+1} \bar{\alpha}_{t+1}(i)
\end{align*}
\]

Rabiner leaves out the full steps to compute \( \hat{\beta}_t(i) \). We can use the following (also from Rahimi):

\[
\begin{align*}
\bar{\beta}_T(i) &= \beta_T(i) \\
\bar{\beta}_t(j) &= \sum_{i=1}^{N} a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(i) \\
\hat{\beta}_t(i) &= c_t \bar{\beta}_t(i)
\end{align*}
\]

We can express the probability of a sequence given a model using \( P(O|\lambda) = \prod_{t=1}^{T} c_t \), but since this is also a product of probabilities, we are better off using the sum of log probabilities: \( \log[P(O|\lambda)] = -\sum_{t=1}^{T} \log c_t \).

**Multiple observation sequences of variable duration**

While implementing the reestimation formulae for multiple observation sequences of variable duration, we ran into the problem of requiring \( P(O^{(k)}|\lambda) \), where \( O^{(k)} \) is the \( k \)th observation sequence. We can no longer compute this, because we now use log-probabilities. However, we can rewrite these formula to no longer use \( P(O^{(k)}|\lambda) \). The full derivations are left out, but are essentially the same as those by Rahimi. We will also show the reestimation formula for \( \pi \), because both Rabiner and Rahimi do not mention it. They assume a strict left-right model, such as Bakis, where \( \pi_1 = 1 \) and \( \pi_i = 0 \) for \( i \neq 1 \).
We will use the following equalities:

\[
\prod_{s=1}^{t} c_s^k = C_t^k \\
\prod_{s=t+1}^{T_k} c_s^k = D_{t+1}^k \\
\prod_{s=1}^{T_k} c_s^k = C_t^k D_{t+1}^k = C_{T_k}^k \\
\frac{1}{\prod_{t=1}^{T_k} c_t^k} = \frac{1}{C_{T_k}^k} = P(O^{(k)}|\lambda)
\]

where \( c_t^k \) is the scaling factor \( \frac{1}{\sum_{j=1}^{N} \alpha_t^k(j)} \). Because we now have a new way of representing \( P(O^{(k)}|\lambda) \) as \( \frac{1}{C_{T_k}^k} \), we can substitute that into the reestimation equations, leading to the following equations after some rewriting:

\[
\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \hat{a}_t^k(i) a_{ij} (O_t^{(k)}) \hat{\beta}_t^k(j)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \hat{a}_t^k(i) \hat{\beta}_t^k(j) \frac{1}{c_t^k}} \\
\bar{b}_j(\ell) = \frac{\sum_{k=1}^{K} \sum_{t \in [1, T_k-1]} \hat{a}_t^k(i) \hat{\beta}_t^k(j)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k-1} \hat{a}_t^k(j) \hat{\beta}_t^k(j) \frac{1}{c_t^k}} \\
\bar{\pi}_i = \frac{\sum_{k=1}^{K} \hat{a}_1^k(i) \hat{\beta}_1^k(j) \frac{1}{c_1^k}}{\sum_{j=1}^{N} \sum_{k=1}^{K} \hat{a}_1^k(j) \hat{\beta}_1^k(j) \frac{1}{c_1^k}}
\]

For the full details and derivations of the reestimation equations, please see the explication by Rahimi or contact the authors of this study. The documented code for jpHMM will be published on-line soon.