Leveraging Disease Progression Learning for Medical Image Recognition

Qicheng Lao
Department of Computer Science and Software Engineering
Concordia University
Montréal, Québec, Canada
qi_lao@encs.concordia.ca

Thomas Fevens
Department of Computer Science and Software Engineering
Concordia University
Montréal, Québec, Canada
fevens@encs.concordia.ca

Boyu Wang
Princeton Neuroscience Institute
Princeton University
Princeton, New Jersey, USA
boyuw@princeton.edu

Abstract—Unlike natural images, medical images often have intrinsic characteristics that can be leveraged for neural network learning. For example, images that belong to different stages of a disease may continuously follow a certain progression pattern. In this paper, we propose a novel method that leverages disease progression learning for medical image recognition. In our method, sequences of images ordered by disease stages are learned by a neural network that consists of a shared vision model for feature extraction and a long short-term memory network for the learning of stage sequences. Auxiliary vision outputs are also included to capture stage features that tend to be discrete along the disease progression. Our proposed method is evaluated on a public diabetic retinopathy dataset, and achieves about 3.3% improvement in disease staging accuracy, compared to the baseline method that does not use disease progression learning.

Index Terms—disease progression, medical image, deep learning

I. INTRODUCTION

Deep learning has been widely applied to medical image recognition since its great success in natural image recognition, and has achieved state-of-the-art performance in various areas such as anatomical structure identification, lesion detection and classification [1], [2]. Unlike general object classification, medical images often have intrinsic characteristics that can be exploited to facilitate neural network learning for improved results. For example, medical recognition tasks on images acquired from computed tomography (CT) or magnetic resonance imaging (MRI) generally favor a 3D convolutional neural network (CNN) over a 2D CNN, due to the additional spatial information in three dimensions [3], [4]. Another example of neural networks that exploit medical intrinsic information is the BrainNetCNN, where special edge-to-edge, edge-to-node and node-to-graph convolutional filters are designed to leverage topological locality of brain networks for the prediction of neurodevelopmental outcomes [5].

In this paper, we propose a novel method that leverages disease progression learning for medical image recognition. Concretely, given any medical recognition problem that is associated with a disease progression, we use long short-term memory (LSTM) to model the disease progression, for example, to predict survival time based on brain tumor images (e.g. short-term survival, medium-term survival and long-term survival) [6], or a disease staging problem. As illustrated in Figure 1(a, b, c), the disease staging problem commonly exists across multiple modalities, including but not limited to classifying breast histopathological images into normal, benign, in-situ or invasive [7]; MRI images of white matter into mild, moderate or severe based on age-related changes [8]; and retinal fundus images into no-diabetic-retinopathy (NDR), simple-diabetic-retinopathy (SDR), pre-proliferative-diabetic-retinopathy (PPDR) or proliferative-diabetic-retinopathy (PDR) [9]. Rather than considering each different stage as an independent class as done in most previous research, we hypothesize that there is a stage difference memory driven by the disease progression that should also be represented in the neural network. By leveraging this memory information from the stage sequence, more robust and optimal results could be possible.

LSTM is a widely-used network that is powerful for sequential data learning, as it has memory units that efficiently remember previous steps [10]. There are previous publications using LSTM for medical data, but mostly based on diagnostic text reports [11], 3D image stacks [12], [13], or clinical measurements/admissions [14]–[16], among which, some also use the concept of disease progression modeling [15], [16]. However, all previous work either do not clearly define stage classes for a disease progression, or still consider each stage as an independent class and the sequence learning only occurs within each individual stage class. I.e., there is no explicit learning of the stage sequence itself, other than learning of temporal sequence (e.g. a series of clinical events), or spatial sequence (e.g. 3D MRI/CT images).

To the best of our knowledge, we are the first to use LSTM for the learning of stage sequence along the disease progression for medical image recognition, with each stage represented by feature vectors that are extracted from a well-established vision model (e.g. GoogleNet or ResNet). Auxiliary outputs from the vision model are also adopted in order to capture stage features that may not be continuous along the disease progression. Our proposed method is evaluated on a diabetic retinopathy dataset, where it shows a performance increase of around 3.3% in disease staging accuracy, compared
to the baseline method that is similar but without disease progression learning.

II. METHODS

A. Disease progression learning

Given a disease progression with \( K \) sequential stages \( S = \{S_k, k = 0, 1, ..., K - 1\} \), with \( S_k > S_{k-1} \) for each \( k \in (0, K - 1) \), where the greater-than sign indicates a disease progression, meaning stage \( S_k \) is a subsequent stage of \( S_{k-1} \), e.g. \{NDR, SDR, PPDR, PDR\} for diabetic retinopathy and \{normal, benign, in-situ, malignant\} for epithelial cancers, we want the neural network to learn disease stage progression by presenting the network with a sequence of images ordered by disease stage as the input \( x = [I^{S_0}, I^{S_1}, ..., I^{S_{K-1}}] \), where \( I^{S_k} \) is a randomly selected image that belongs to stage \( S_k \), and the corresponding output is simply the ordered full sequence of all disease stages: \( y = [S_0, S_1, ..., S_{K-1}] \). However, with the above design, all the input samples would share the same output \( y \) that is fixed by the disease stage progression, making the network training meaningless. To overcome this problem, we artificially define \( S_0 > S_{K-1} \) to make the stage sequence cyclic (Figure 1(d)), so that multiple ordered full sequences of all disease stages can be generated. Therefore, \([S_1, S_2, ..., S_{K-1}, S_0]\), for example, is also considered as a valid ordered full sequence containing all stages for a certain disease. As a result, the notation of stage sequence for a training sample can be updated more formally by the introduction of a modulo operation as the following:

\[
(x, y) = ([I^{S_{i+0}} \pmod K], I^{S_{i+1}} \pmod K), ..., I^{S_{i+K-1}} \pmod K],
\]

\[
[S_{i+0} \pmod K], S_{i+1} \pmod K), ..., S_{i+K-1} \pmod K])
\]

where \( i \in [0, K - 1] \) can be considered as the step shift size, indicating the starting stage of a sequence, and \( K \) is the total number of disease stages. For simplicity of notation and explanation, we use the stage sequence of \( i = 0 \) as an illustration in the following.

As shown in Figure 2 the proposed method contains a vision model (e.g. GoogleNet or ResNet) for the feature extraction, followed by a LSTM model for the purpose of disease progression learning. The vision model extracts a feature vector \( z^{S_k} \in \mathbb{R}^C \) from its corresponding input image \( I^{S_k} \) for each disease stage \( S_k \), and then the concatenated feature vector sequence of all stages \( z = [z^{S_0}, z^{S_1}, ..., z^{S_{K-1}}] \in \mathbb{R}^{K \times C} \) is given as the input to LSTM. The LSTM maintains a hidden state \( h^{S_k} \in \mathbb{R}^C \) and a cell state \( c^{S_k} \in \mathbb{R}^C \), which are updated at each stage step \( k \in (0, K - 1) \):

\[
h^{S_k}, c^{S_k} = \text{LSTM}(z^{S_k}, h^{S_{k-1}}, c^{S_{k-1}})
\]

(2)

The hidden state sequence of all stages \( h = [h^{S_0}, h^{S_1}, ..., h^{S_{K-1}}] \in \mathbb{R}^{K \times C} \) is collected from the multi-output LSTM to learn a softmax classifier \( F \), which finally outputs the hypotheses \( \hat{y} \) of the true stage labels \( y \):

\[
\hat{y} = F(h; \theta)
\]

(3)

where \( \theta \) is the parameter for classifier \( F \). Note that \( \hat{y} = [\hat{y}^{S_0}, \hat{y}^{S_1}, ..., \hat{y}^{S_{K-1}}] \in \mathbb{R}^{K \times 1} \), where each sequence element represents a probability distribution over \( K \) stage labels for its corresponding stage. Also, it is worth noting that although for the simplicity of notation, we denote both \( h \) and \( y \) as vector sequences of all stages, the softmax classifier \( F \) is actually applied to each individual hidden state \( h^{S_k} \) independently to obtain its corresponding hypothesis \( \hat{y}^{S_k} \).

Fig. 1. Examples of disease progression with different stages: (a) Breast cancer with five stages: normal, benign proliferation, atypical hyperplasia, in-situ and invasive \[7\]; (b) Aging related changes in white matter with three severity grades: mild, moderate and severe \[8\]; (c) Diabetic retinopathy with four stages: non-diabetic retinopathy (NDR), simple-diabetic-retinopathy (SDR), pre- proliferative-diabetic-retinopathy (PPDR) and proliferative-diabetic-retinopathy (PDR) \[9\]; (d) Cyclic form of the stage sequence.

Fig. 2. (a) Example features for epithelial cancers, we artificially define the notation of stage sequence as the following:
Our loss function is a weighted summation of cross entropy losses at all stages:

$$\text{Loss}(\hat{y}, y) = \sum_{k=0}^{K-1} \alpha_k \cdot l(\hat{y}^{S_k}, S_k)$$  \hspace{1cm} (4)$$

where $\alpha_k$ is the loss weight for each stage output and

$$l(\hat{y}^{S_k}, S_k) = \sum_{j=0}^{K-1} -\log \hat{y}^{S_k}_j \cdot \delta(S_k = S_j)$$  \hspace{1cm} (5)$$

is the cross entropy loss function, with $\hat{y}^{S_k}_j$ denoting the probability value of image $I^{S_k}$ belonging to the stage label $S_j$.

B. Auxiliary vision outputs

In addition to disease progression learning, the network should also be able to capture stage-wise discriminative features that tend to be discrete among different stages along the disease progression, e.g. features that only appear in a certain disease stage. Given this thought, we also design auxiliary outputs directly from the vision model extracted features $z$ with another softmax classifier $F_v$ on top of the vision model:

$$\hat{y}_v = F_v(z; \theta_v)$$  \hspace{1cm} (6)$$

To distinguish the two classifiers on top of LSTM model ($F_l$) and vision model ($F_v$), Equation (3) is updated as:

$$\hat{y}_l = F_l(h; \theta_l)$$  \hspace{1cm} (7)$$

Similarly to LSTM outputs, the loss for auxiliary vision outputs is also defined as the weighted summation of cross entropy losses at all disease stages. Taking both losses into account, the final loss function for our proposed network is:

$$\text{Loss}(\hat{y}_l, \hat{y}_v, y) = \sum_{k=0}^{K-1} (\alpha_k \cdot l(\hat{y}^{S_k}_l, S_k) + \beta_k \cdot l(\hat{y}^{S_k}_v, S_k))$$  \hspace{1cm} (8)$$

The loss weights $\alpha_k$ and $\beta_k$ for each stage should be chosen depending on each individual disease progression. A heuristic choice is that more weight should be given to $\beta_k$ if the stage features tend to be more discrete.

C. Non-regression disease stage sequence

In the context of disease progression learning (Section II-A), we artificially define $S_0 > S_{K-1}$ to make the stage sequence cyclic, so that more variations of stage sequences can be generated for a particular disease to facilitate the neural network training. However, it is a bit counter-intuitive at the first thought to define $S_0 > S_{K-1}$, which violates the concept of monotonic disease progression. Alternatively, we could also use other approaches that do not require any similar assumptions, such as non-regression disease stage sequence, where the sequence can still start with any arbitrary disease stages and the disease simply stops progression when it reaches its final stage. In this way, the order of disease stages is still maintained without the assumption defined in the cyclic strategy. For example, $[S_1, S_2, \ldots, S_{K-2}, S_{K-1}, S_{K-1}]$ is also a valid non-regression stage sequence. Therefore, a training sample of non-regression disease stage sequence can be formulated as the following:

$$(x, y) = \left( [I_{\min(i+0,K-1)}, I_{\min(i+1,K-1)}, \ldots, I_{\min(i+K-1,K-1)}], \right. \left. [S_{\min(i+0,K-1)}, S_{\min(i+1,K-1)}, \ldots, S_{\min(i+K-1,K-1)}] \right)$$  \hspace{1cm} (9)$$
where \( i \in [0, K - 1] \) is the step shift size, indicating the starting stage of a sequence, and \( K \) is the total number of disease stages. Although a non-regression disease stage sequence is more intuitive as compared to a cyclic stage sequence, we will show in Section III-C that it does not outperform the cyclic stage sequence.

D. Testing phase

In the testing phase, given an image \( I_{test} \), an artificial image sequence is generated by repeating \( I_{test} \) for \( K \) iterations, and then the sequence is fed into the trained network. Due to the design of our network, we have two options to predict its stage label based on either LSTM output \( \hat{y}_i \) or vision output \( \hat{y}_v \) (see Figure 2). In both cases, only the first stage output in the sequence is reported as the final predicted label. Note that the input image sequence can be started with any arbitrary stage as we described earlier in Section II-A and Section II-C. Therefore a faked sequence starting with \( I_{test} \) can still result in a reasonable prediction of stage label for \( I_{test} \) based on its corresponding first stage output, although the rest stage outputs are invalid since there is no disease progression in the input sequence.

E. Baseline network

The baseline network for this study is the same vision model followed by the same softmax classifier \( F_v \) as used in our proposed network, except that the input is a single image, similar to most previous research work on medical image classification.

For a fair comparison, we also make sure that both networks are trained on exactly the same amount of augmented data, which we will describe in detail in Section III-B to rule out the possibility that a superior performance could be gained simply due to larger training samples, given the fact that the input size of our proposed network is \( K \) times bigger than that of the baseline network.

III. EXPERIMENTS AND RESULTS

A. Dataset

To evaluate our proposed network, we choose to use a recently published dataset on diabetic retinopathy [9], which contains fundus photographs of four stages. The dataset has two sets of class labels based on whether to grade with wider retinal area: Davis grading of one figure and Davis grading of concatenated figures, and we use the former in our experiments. There are in total 9939 images, with 6561 of NDR, 2113 of SDR, 460 of PPDR and 805 of PDR.

Unlike previous work [9], we address the data imbalance problem by undersampling by a random selection of 460 images from each stage class for each independent experiment, and moreover, all the images are resized to 200 × 200 instead of 1272 × 1272 to speed up training, since it is not our purpose here to compete with the result in [9], but rather to investigate the advantage of disease progression learning in stage classification.

Out of the final resulting dataset, 10% of the data is randomly reserved for testing, and the rest is further split into two parts: 90% for training and 10% for validation. Data augmentation is performed on the training set by only using random rotations from \(-5^\circ\) to \(5^\circ\) without shifts or flips, as all the images share the same position.

B. Implementation details

We examine two types of vision models in our experiments: GoogleNet and ResNet-50, both of which were pretrained on ImageNet. The freeze layer is set to 64 for GoogleNet and 36 for ResNet-50. We use \( C = 256 \) for the dimension of extracted feature vectors, \( G = 256 \) for the LSTM model and \( \alpha = \beta = 1 \in \mathbb{R}^K \). All models are implemented in Keras with Theano backend, and trained using stochastic gradient descent, with the initial learning rate set to 0.001, decay by 1e-6 over each update, and Nestrov momentum is set to 0.9.

For each independent experiment, we evaluate the proposed network and baseline network on the same random split of the dataset. We repeat the above comparison experiment for 20 times, each with a different split. To make sure both networks are trained on the same amount of augmented data, the same number of iteration steps per epoch (i.e. 100 steps and 500 steps) is used in the training, until the validation loss does not drop for ten epochs (patience = 10, with the maximum number of epochs set to 100). In order to compensate the input size difference (i.e. one-image input against four-image input), the batch size for the baseline network training is set to 64 and decreased to 16 for our proposed network. Note that this is the only difference in hyper-parameter settings for the training of the two networks.

C. Results

Table 1 shows the performance comparison of our proposed method and the baseline method, with both results of GoogleNet Inception v3 and ResNet-50 as the vision model. As shown in the table, our proposed method outperforms the baseline across all experimental settings that are used in this paper (choices of vision model and training steps per epoch), and on average, there is about 3.3% accuracy gain (2.4%, 3.6%, 3.4% and 3.6% for each setting respectively). However, we do not observe a significant performance difference between vision outputs and LSTM outputs (59.4% vs 59.2%, 63.1% vs 62.9%, 60.7% vs 60.7% and 61.4% vs 61.7%) for our proposed method, probably due to the joint training of our model, so that the LSTM backpropagates the gradients to CNN to force it to learn features that are more stage-relevant, thus improving the vision outputs.

For both methods, accuracy performance can be improved with an increased number of training steps per epoch, which is within our expectation since more steps means more data augmentation. The best performance in our performed experiments is 63.1% for our proposed method and 59.5% for the baseline method, both of which are using GoogleNet Inception v3 pretrained on ImageNet with 500 training steps per epoch, and the confusion matrices are given in Figure 3. It is also
noted that most of the misclassified samples are located in NDR and SDR, which is reasonable since the two stages are often quite indistinguishable from a clinical perspective.

We believe that there is still room for performance improvement on this particular diabetic retinopathy problem by doing more data augmentation, using oversampling instead of undersampling (i.e. to make full use of images in the dataset), and using the original high resolution images. However, the purpose of this paper is to present and validate the idea of disease progression learning, and we leave the optimization for our next study.

To further test the alternative design of disease progression learning using non-regression stage sequences, we perform the comparison experiments using the same above settings, except that the model is trained on non-regression stage sequences (described in Section II-C), instead of cyclic stage sequences (described in Section II-A). As shown in Table II, the non-regression stage sequence fails to outperform cyclic stage sequence in all experiment settings, and in some case (GoogleNet and 100 training steps per epoch), it is even worse compared to the baseline method. The best result achieved by non-regression stage sequence is 61.8%, which is still worse than that of cyclic sequence 63.1%.

Based on the above results, we argue that despite being counter-intuitive at first thought, the cyclic strategy is merely to facilitate the training, which allows for cyclic variations of the training data that then de-emphasizes the first class as the first node, etc. This in turn allows the same disease progression sequence to be used for more training data. Since the change will be very abrupt from the last node to the first node, the learning will still retain the last node’s characteristics as being

![Confusion matrices](image)

**Fig. 3.** Confusion matrices of the baseline method (left) and our proposed method (right) using GoogleNet as the vision model with 500 training steps per epoch.

### TABLE I

**Performance comparisons of the baseline method and our proposed method trained on cyclic stage sequences (with both vision outputs and LSTM outputs).**

| Vision model | Steps | Method          | Accuracy  |
|--------------|-------|-----------------|-----------|
| **GoogleNet** |       |                 |           |
| Inception v3 | 100   | BASELINE        | 57.0 ± 3.2% |
|              | 500   | OURS(VISION OUTPUT) | 59.4 ± 3.1% |
|              |       | OURS(LSTM OUTPUT) | 59.2 ± 3.3% |
|              |       | BASELINE        | 59.5 ± 3.3% |
|              |       | OURS(VISION OUTPUT) | 63.1 ± 3.7% |
|              |       | OURS(LSTM OUTPUT) | 62.9 ± 3.0% |
| **ResNet-50** | 100   | BASELINE        | 57.3 ± 6.4% |
|              | 500   | OURS(VISION OUTPUT) | 60.7 ± 5.4% |
|              |       | OURS(LSTM OUTPUT) | 60.7 ± 5.7% |
|              |       | BASELINE        | 55.1 ± 4.8% |
|              |       | OURS(VISION OUTPUT) | 61.4 ± 3.8% |
|              |       | OURS(LSTM OUTPUT) | 61.7 ± 3.9% |

*Training steps per epoch.

### TABLE II

**Performance comparisons of the baseline method and our proposed method trained on non-regression stage sequences (with both vision outputs and LSTM outputs).**

| Vision model | Steps | Method          | Accuracy  |
|--------------|-------|-----------------|-----------|
| **GoogleNet** |       |                 |           |
| Inception v3 | 100   | BASELINE        | 57.5 ± 2.9% |
|              | 500   | OURS(VISION OUTPUT) | 56.2 ± 3.8% |
|              |       | OURS(LSTM OUTPUT) | 56.7 ± 4.3% |
|              |       | BASELINE        | 59.1 ± 3.5% |
|              |       | OURS(VISION OUTPUT) | 61.8 ± 2.7% |
|              |       | OURS(LSTM OUTPUT) | 60.9 ± 2.5% |
|              |       | BASELINE        | 58.5 ± 3.8% |
|              |       | OURS(VISION OUTPUT) | 59.6 ± 4.3% |
|              |       | OURS(LSTM OUTPUT) | 59.6 ± 4.3% |
|              |       | BASELINE        | 58.2 ± 3.5% |
|              |       | OURS(VISION OUTPUT) | 60.5 ± 2.5% |
|              |       | OURS(LSTM OUTPUT) | 60.1 ± 2.8% |

*Training steps per epoch.
substantially distinct from the first node of the sequence.

IV. Conclusion

In this paper, we present a novel method for medical image recognition by leveraging disease progression learning, where stage sequences are learned by LSTM after feature extraction with a shared vision model for the images from each stage. Compared to the baseline method that is a pure vision model, our proposed method has an average of 3.3% accuracy increase based on our performed experiments, when evaluated for the problem of disease staging on a diabetic retinopathy dataset.

References

[1] Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., Van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Medical image analysis 42, 60–88 (2017)

[2] Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. Annual review of biomedical engineering 19, 221–248 (2017)

[3] Payan, A., Montana, G.: Predicting alzheimer’s disease: a neuroimaging study with 3d convolutional neural networks. arXiv preprint arXiv:1502.02506 (2015)

[4] Huang, X., Shan, J., Vaidya, V.: Lung nodule detection in ct using 3d convolutional neural networks. In: Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on. pp. 379–383. IEEE (2017)

[5] Kawahara, J., Brown, C.J., Miller, S.P., Booth, B.G., Chau, V., Grunau, R.E., Zwicker, J.G., Hamarneh, G.: Brainmetcnn: convolutional neural networks for brain networks; towards predicting neurodevelopment. NeuroImage 146, 1038–1049 (2017)

[6] Nie, D., Zhang, H., Adeli, E., Liu, L., Shen, D.: 3d deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 212–220. Springer (2016)

[7] Burstein, H.J., Polyak, K., Wong, J.S., Lester, S.C., Kaelin, C.M.: Ductal carcinoma in situ of the breast. New England Journal of Medicine 350(14), 1430–1441 (2004)

[8] Inzitari, D., Pracucci, G., Poggesi, A., Carlucci, G., Barkhof, F., Chabrier, H., Erkinjuntti, T., Fazekas, F., Ferro, J.M., Hennerici, M., et al.: Changes in white matter as determinant of global functional decline in older independent outpatients: three year follow-up of ladis (leukoaraiosis and disability) study cohort. Bmj 339, b2477 (2009)

[9] Takahashi, H., Tampo, H., Arai, Y., Inoue, Y., Kawashima, H.: Applying artificial intelligence to disease staging: Deep learning for improved staging of diabetic retinopathy. PloS one 12(6), e0179790 (2017)

[10] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)

[11] Zhang, Z., Xie, Y., Xing, F., McGough, M., Yang, L.: Mdnet: A semantically and visually interpretable medical image diagnosis network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6428–6436 (2017)

[12] Chen, J., Yang, L., Zhang, Y., Alber, M., Chen, D.Z.: Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In: Advances in Neural Information Processing Systems. pp. 3036–3044 (2016)

[13] Xue, W., Lum, A., Mercado, A., Landis, M., Warrington, J., Li, S.: Full quantification of left ventricle via deep multitask learning network respecting intra-and inter-task relatedness. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 276–284. Springer (2017)

[14] Lipton, Z.C., Kale, D.C., Elkan, C., Wetzel, R.: Learning to diagnose with lstm recurrent neural networks. arXiv preprint arXiv:1511.03677 (2015)

[15] Pham, T., Tran, T., Phung, D., Venkatesh, S.: Deepcare: A deep dynamic memory model for predictive medicine. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 30–41. Springer (2016)

[16] Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F., Sun, J.: Doctor ai: Predicting clinical events via recurrent neural networks. In: Machine Learning for Healthcare Conference. pp. 301–318 (2016)