INCREMENTAL TRANSFER LEARNING FOR OBJECT RECOGNITION IN STREAMING VIDEO

Jongdae Kim and John Collomosse

Centre for Vision Speech and Signal Processing,
University of Surrey,
Guildford, United Kingdom.
{jongdae.kim | j.collomosse}@surrey.ac.uk

ABSTRACT

We present a new incremental learning framework for real-time object recognition in video streams. ImageNet is used to bootstrap a set of one-vs-all incrementally trainable SVMs which are updated by user annotation events during streaming. We adopt an inductive transfer learning (ITL) approach to warp the video feature space to the ImageNet feature space, so enabling the incremental updates. Uniquely, the transformation used for the ITL warp is also learned incrementally using the same update events. We demonstrate a semi-automated video logging (SAVL) system using our incrementally learned ITL approach and show this to outperform existing SAVL which uses non-incremental transfer learning.

Index Terms— Object Recognition, Incremental Learning, Transfer Learning, Video Classification.

1. INTRODUCTION

Object recognition 'in the wild' remains an open challenge; it is difficult to train systems to classify objects correctly over diverse footage the nature of which is unknown *a priori*. Recently, significant progress has been made using 'transfer learning' approaches that train classifiers on object exemplars captured under one set of conditions (the *source domain*), and transform those classifiers to be applicable to the classifying content captured under another set of conditions (the *target* domain). One promising class of technique is *inductive transfer learning* (ITL) [1] that seeks to warp the feature space of the source domain to the target e.g. by estimating a linear transformation between the two based on class correspondences established between domains.

In this paper we apply the principles of ITL to the problem of incrementally learning classifiers for object recognition in streaming video. Our incrementally trainable classifiers improve accuracy as additional annotation is supplied piece-meal by the user, whilst watching the video stream. A practical use case is broadcast production where raw footage is 'logged' (annotated with metadata) on-set by a human operator. Due to the high volume of footage, and the need for expeditious logging for downstream production, this process is performed in real-time as footage is recorded. We envisage semi-automated video logging (SAVL) system in which occasional user annotations are used to incrementally train and improve classifiers until the user-desired degree of automated recognition is reached. For SAVL to be practical, interactive training should be minimal whilst retaining high accuracy. One strategy is to bootstrap classifiers through a pre-training step prior to the start of streaming annotation. This motivates transfer learning approaches, since the pre-training dataset may not resemble the video data to be annotated.

The technical contribution of this paper is the first use of ITL in a SAVL system. We bootstrap our classifiers by pre-training over relevant data sampled from ImageNet [2]. Our classifiers are incrementally trainable Support Vector Machines (SVMs) which are updated at each user-annotation event. Uniquely, the feature space warp used for ITL is *also learned incrementally* and updated with each user annotation event. We show that our incremental warping approach to SAVL outperforms existing state-of-the-art SAVL where this warping transform is estimated once during the pre-training and held constant subsequently.

2. RELATED WORK

Object recognition is a long-studied problem, yet classification methods achieving a high level of performance typically require hundreds of exemplar images per class and a trained in a batch process[3]. Recent algorithms can build upon batch training to learn novel categories, by generalising (transferring learning) from existing categories. Fei-Fei *et al.* described a 'one-shot' Bayesian learning framework for image classification in which a handful of examples are sufficient [4, 5]. Subsequently, alternative one-shot frameworks have been proposed e.g. using adaptive SVMs (A-SVMs) [6] to transfer learning from ImageNet [7]. Under Pan's taxonomy [1] such systems are ITL since exemplars are provided in the target domain. Zero-shot frameworks have also been proposed for constrained classes of object attribute [8] or activity [9].

In addition to efficient learning of novel categories, algorithms have been proposed to refine training on existing categories using incremental learning techniques such as multiple kernel learning for image classification [10]. In previous work [11] we described a SAVL system that combined the LASVM framework [12] to incrementally learn max-margins for object categories online, with the A-SVMs of [7] for ITL from ImageNet. LASVMs are combine Sequential Minimal Optimization and support vector removal yielding a system with low memory overhead (prior data and decision errors need not be retained between training iterations) with comparable performance to off-line SVM training [11] and fast update times. However A-SVMs and other prior ITL work for object recognition perform a once-only estimation of the transform mapping the source feature domain to the target. In a SAVL framework, new training annotations become incrementally available during playback and this one-off process may subsequently become sub-optimal for ITL.

In this paper we extend [11] to enable incremental learning of the ITL transform, yielding a fully incrementally trainable SAVL solution that combines both incremental classification (LASVM) and incremental ITL. For the latter we adapt the LORETA (Low rank retraction algorithm) algorithm [13].

3. VIDEO CLASSIFICATION OVERVIEW

Our SAVL framework operates upon streaming video in real-time. A set of object categories are named and a set of one-vs-all SVMs initialised prior to the commencement of streaming; specifically one LASVM [12] is initialised corresponding to each object category. During playback, each video frame is processed through the LASVM bank to detect the presence of the trained categories. Optionally, the user may manually indicate the presence of an object in the frame by touching the appropriate object category name in a video playback interface. This generates an update event. Subsequent frames are classified incorporating this new information.

3.1. Bootstrapping

ImageNet is used as an auxilliary data source for initial training. When an update event occurs, and training has yet to be received for all categories, a search of ImageNet is performed using both the pre-supplied object category name and the frame as a visual exemplar. Ferrari *et al.*'s ImageNet distance [14] is used to identify the closest visual exemplars (in our experiment, 40) which are used as positive examples to bootstrap the relevant LASVM, and negative examples for the others. Once training exemplars have been supplied for all categories the system transfers from the Bootstrapping to the Updating phase.

3.2. Updating

Domain adaptation is necessary to fuse training from the ImageNet and video feature spaces. This is because the distribution formed by visual exemplars e.g. for 'car' in ImageNet differs from the distribution formed by similar semantic objects in the video domain. In our ITL framework we build our LASVM classifiers in the ImageNet domain, i.e. the 'target' domain. Our video represents the 'source' domain from which we must transfer incrementally supplied training examples, in order that they be added to the LASVMs (Fig. 1).

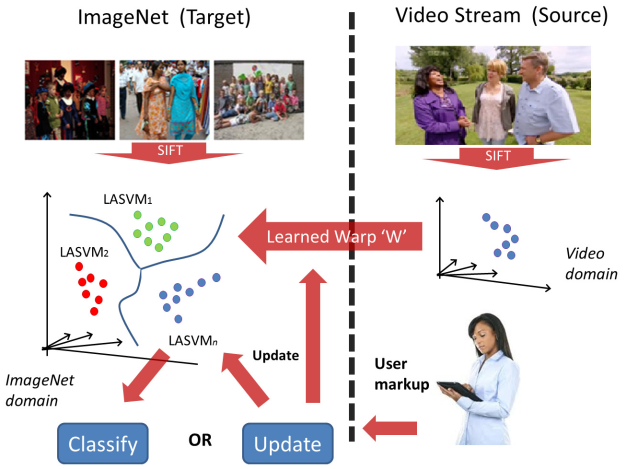


Fig. 1. Proposed framework. LASVM built in the target (ImageNet) domain are updated by transformed data from source (Video) domain using an incrementally learned transform.

When an update event occurs, the visual representation obtained from the video frame (source domain) is transferred to the target domain via a linear transformation (c.f. W in Sec. 4.2). The relevant LASVM is incrementally updated with this warped data as a positive exemplar; the others are updated with the new data as a negative exemplar. The LASVM update process is outlined in Sec. 4. W is also incrementally updated to improve the accuracy of the ITL. This is the novel step over [11] and is described in Sec. 4.2.

3.3. Visual Representation

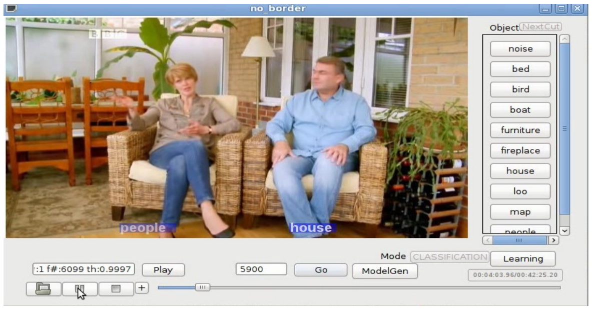
In both domains, the image data is distilled into a feature vector using the bag-of visual words (BoVW) pipeline. We extract a 128-dimensional SIFT [15] features from 20x20 overlapping patches densely sampled using the VLFeat [16] library. A codebook of 1000 words is generated using k-means and each visual descriptor is assigned, through approximate $k-NN$ to a single cluster center. The codebook is built *a priori* using randomly sampled ImageNet data. Despite using identical representations, the statistical distribution of exemplars in this space differs significantly between domains for the similar semantic objects necessitating use of ITL.

4. INCREMENTAL LEARNING FOR SAVL

4.1. Incremental Transfer Learning

Our incremental ITL method linearly warps visual features between domains via an incrementally learned transformation (or 'similarity') matrix estimated iteratively using a Riemannian manifold model.

We first describe our general domain adaptation model in linear setting. We assume that two domains A and B contain data $x_{A1},\dots,x_{An}$, and $x_{B1},\dots,x_{Bm}$ for which class labels are known (i.e. ITL). We adopt the linear framework of [17] to learn transformation W from *B* to *A*. A matrix $W \in \mathbb{R}^{n\times m}$



:1 f#:6099 th:0.9997

5900

Mode CLASSIFICATION

00:04:03.96/00:42:25.20

Fig. 2. Streaming video is classified and corrected in real-time by the user who indicates category presence via the UI.

of rank r can be factorized to a product of two r dimension matrices $W = AB^T$, $A \in \mathbb{R}^{n\times r}$,$B \in \mathbb{R}^{m\times r}$. Each product term is computed by applying gradient descent [18].

We use an online learning algorithm on the Riemannian manifold of low-rank matrices to learn transformation matrices (A and B). We start with a brief introduction of optimization of Riemannian manifold and retraction [19].

The basic assumption of manifold learning algorithm is that the input data exists on low dimensional manifold embedded in a space $\mathbb{R}^n$. Namely, a smooth subset of $\mathbb{R}^n$ is a embedded manifold. We focus on the Riemannian manifold of *r*-rank matrices of size $n \times m$ with $r \ll \min\{m, n\}$ and denote $M_r^{n,m}$. For incremental learning, a stochastic gradient descent is adopted to minimize a loss function *l(W)* for each point *W* over $M_r^{n,m}$,

$$\min_W l(W) \quad \text{s.t} \quad x \in M_r^{n,m}. \tag{1}$$

The stochastic gradient descent algorithm takes two steps at each step *t* to solve the above problem. The first step is to compute Riemannian gradient $\xi^t = \nabla l(W^t)$ which is a projection of the Euclidean gradient in ambient space onto the tangent space $T_W M$ associated with a point x. Then the Riemannian gradient ξ^t enables us to yield $W^{t+\frac{1}{2}} = W^t + \xi^t$. The second step is to map the ξ^t back onto the Riemannian manifold; the process of *retraction*. As the result of the process we have $W^{t+1} = R_{W^t}(-\eta^t \xi^t)$, where η is step size and R indicates retraction.

To describe an online algorithm to learn low-rank matrices, we define the tangent space $T_w M$ as following [13] ;

$$T_W M = \begin{bmatrix} A & A_\perp \end{bmatrix} \times \begin{bmatrix} M & N_1^T \\ N_2 & 0 \end{bmatrix} \times \begin{bmatrix} B_T \\ B_\perp^T \end{bmatrix}. \tag{2}$$

where $W \in M_r^{n,m}$ can be factorized into $W = AB^T$,and $A_\perp$ and $B_\perp$ are the orthogonal complements of A and B individually. Also M,N_1 and N_2 being $k \times k$, $(m - r) \times r$, and $(n - r) \times r$ matrix separately, the proof can be found appendix of [13]. angent vector ξ can be decomposed into three orthogonal elements;

$$\begin{aligned} &\xi = \xi^{AB} + \xi^{AB_\perp} + \xi^{A_\perp B}, \text{where} \\ &\xi^{AB} = AMB^T, \xi^{AB_\perp} = AN_1^T B_\perp^T, \text{and } \xi^{A_\perp B} = A_\perp N_2 B^T. \end{aligned} \tag{3}$$

Algorithm 1 The r-rank Riemannian stochastic gradient descent algorithm of [19] instantiated for our SAVL framework.

```
Input: $x_1, x_2$
Init: $A$,$B$
Output: $Z_1$, $Z_2$
  Compute :                                              matrix dimension
  $A^+ = (A^T A)^{-1} AT, B^+ = (B^T B)^{-1} B^T$                     $r \times n, r \times m$
  $\alpha_1 = A^+ \cdot x_1, \beta_1 = B^+ \cdot x_2$                             $r \times 1, r \times 1$
  $\alpha_2 = A \cdot \alpha_1$                                              $n \times 1$
  $H = \beta_1^T \cdot \alpha_2$                                             $1 \times 1$
  $\alpha_3 = -\frac{1}{2}\alpha_2 + \frac{3}{8}\alpha_2 \cdot H + x_1 - \frac{1}{2}x_1 \cdot H$              $n \times 1$
  $Z_1 = A + \alpha_3 \cdot \beta_1^T$                                       $n \times r$
  $\beta_2 = (x_2^T B) \cdot B^+$                                       $1 \times m$
  $\beta_3 = -\frac{1}{2}\beta_2 + \frac{3}{8}H \cdot \beta_2 + x_2^T - \frac{1}{2}H \cdot x_2^T$             $1 \times m$
  $Z_2^T = B^T + \alpha_1 \cdot \beta_3$                                     $r \times m$
```

Then we define the three matrices M,N_1 and N_2 related to the projection of the gradient matrix denoted as $\hat{Z}$, such that $\hat{Z} = \xi$. If we assume that A is full rank matrix, then its pseudo-inverse A^+ is $(A^T A)^{-1} A^T$. With the assumption,the matrix projection P_A onto A' columns is exactly match to AA^+ and similar to $P_{A\perp}$, P_B and $P_{B\perp}$. For a given matrix Z, we have $M = A^+ Z B^{+T}$, $N_1 = B_\perp^T Z^T A^{+T}$ and $N_2 = A_\perp^T Z B^{+T}$ using (3). It enables us to represent $\xi = \xi^{AB} + \xi^{AB_\perp} + \xi^{A_\perp B}$ in terms of the projection.
The retraction is defined using the following theorem presented below (from [13]):

$$\begin{aligned} V_1 &= W + \frac{1}{2}\xi^{AB} + \xi^{A_\perp B} - \frac{1}{8}\xi^{AB} W^+ \xi^{AB} - \frac{1}{2}\xi^{A_\perp B} W^+ \xi^{AB}, \\ V_2 &= W + \frac{1}{2}\xi^{AB} + \xi^{AB_\perp} - \frac{1}{8}\xi^{AB} W^+ \xi^{AB} - \frac{1}{2}\xi^{AB} W^+ \xi^{AB_\perp}. \end{aligned} \tag{4}$$

The retraction is $R_W(\xi) = V_1 W^+ V_2$. The algorithm 1 shows the procedure of learning at t step. For the first time step, an example is expressed as a pair $x_1 \in A$ and $x_2 \in B$ and x_2 is produced by the difference of two examples (one's label is same as x_1 and the other's label is different from x_1). At $t = 1$, model A and B are initialized as identity matrix. For later t, the algorithm computes the updated model Z_1 and Z_2 which adapts to model A and B for the new data.

4.2. Incremental Classifier Training

We adopt the LASVM algorithm for incremental training of SVMs, introduced by Bordes *et al.* [12]. The LASVM utilizes the Sequential Minimal Optimization (SMO) which offers fast training, using less memory and producing similar performance to batch SVM solvers. LASVM was employed in the SAVL system of Kim *et al.* [11], the principal difference here is the training of LASVM in the ImageNet domain, and the use of incremental ITL to pre-process the video features for compatibility with initial ImageNet-based training.

The LASVM maintains three important pieces of information during the online learning – *S* a set of potential support vectors, α_i coefficients of the current kernel expansion, g_i the partial gradient of the dual objective function **W** defined by

$$\mathbf{W} = \sum_{i=1}^{N} \alpha_i y_i - \frac{1}{2} \sum_{i=1,j=1}^{N} \alpha_i \alpha_j K(x_i, x_j). \quad (5)$$

where i and j are the indices of kernel expansion and $K(x, x)$ is the kernel function.

The LASVM update comprises two basic operations, PROCESS and REPROCESS, which rely on SMO algorithm calculating a τ approximate solution, which is an approximation of functions defined by the equations in [20]. The aim of the PROCESS operation is to recalculate the α and g after add a new vector(x) as a support vector. The operation first trains the new vector for the model, and then finds a corresponding support vector which consists of τ violating pairs with the new vector at maximal g. It finally updates the weight for all support vectors. In subsequent REPROCESS operation, support vectors are eliminated from the kernel expansion S where $\alpha = 0$. The removal of unnecessary vectors mitigates overfitting. The operation begin with finding the τ violating pairs in kernel expansion with maximum gradient as they are not candidates of support vectors anymore. After finding the pairs, it re-calculates the weight for all support vectors. Second, any support vectors whose weight $\alpha = 0$ are removed, and g updated. The final result S provides support vectors and coefficients for a new classifier $f(x)$. Our system only keeps the new classifier for a category and uses it for next incremental procedure — the training data itself is not retained and so avoids cumulative memory requirements for training successive updates.

5. RESULTS AND DISCUSSION

We evaluate our SAVL system on the dataset of [11] comprising $\sim$ 30 minutes of television broadcast footage. As per [11] we perform shot detection and sample a constant number of frames from each shot to prevent bias e.g. from longer 'easier' shots that contain self-similar frames. The data contains 8 object categories: $\{$Bed, Bird, Car, Fireplace, House, Bathroom, Map, People$\}$ and split into a training set (285 frames) and test set (282 frames) such that training and test frames are not drawn from the same shot. We compare classification performance of our method against [11] which uses a non-incrementally learned ITL (A-SVMs) for domain adaptation, and trains LASVMs in the source (video) domain. We also compare against a classical (batch-training) BoVW pipeline with no transfer learning (NTF). To learn W, r-rank is set empirically to 400. LASVMs were configured as per [11].

5.1. Comparative Evaluation

We compute the Mean Average Precision (MAP) i.e. AP averaged over all categories, after exposure to training from 2 (ImageNet only), 10, 50, 100 and 285 frames. Fig. 3 illustrates a performance increase of $\sim 5\%$ using the proposed method, over non-incremental ITL. Both ITL methods clearly outperform non-incremental training SAVL frameworks. The learning rate is approximately equal for all, plateauing after around 100 training examples. Table 1 shows improvement over non-incremental ITL is roughly uniform across categories. Both ITL outperform classical BoVW. The high AP of the Map class is caused by lack of visual variance in that category (a graphic of the UK).

Our C/C++ implementation ran on a 3.6Ghz Pentium 4 PC. Update events took $200ms$ being equally split between LASVM and ITL updates; slightly more expensive than $150ms$ for [11]. The cost is acceptable as occasional updating is unnecessary after classifiers mature, and can be multi-threaded to update in the background. Average classification time (transfer and LASVM prediction) was $600\mu s$ however SIFT extraction takes $400ms$. We work around this bottle-neck by predicting only every 20 frames.

Table 1. Comparison of A-SVM (A) vs. proposed (P)

AP(%)	2		10		50		100		285	
	A	P	A	P	A	P	A	P	A	P
Bed	9	14	16	19	16	20	17	25	14	26
Bird	31	30	33	35	30	41	34	49	55	61
Car	5	6	12	17	13	18	14	20	20	21
Fireplace	11	10	19	20	20	22	19	24	14	20
House	58	57	59	63	61	67	63	68	65	67
Bathroom	31	28	37	39	41	43	40	43	39	43
Map	100	100	100	100	100	100	99	100	100	100
People	54	52	64	67	67	70	67	71	68	73

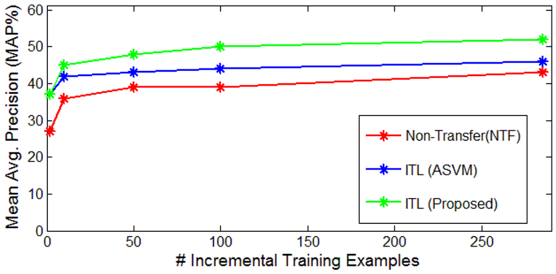


Fig. 3. Relative accuracy of the proposed method (green) vs. non-incremental ITL (A-SVM, blue) and classical non-transfer learning BoVW (NTF, green).

6. CONCLUSION

We have introduced an incremental ITL framework for SAVL and shown this to out-perform non-incremental ITL (and classical BoVW recognition) by $\sim$ 5% with similar computational expense. To the best of our knowledge an incrementally estimated domain transformation is novel to ITL for object classification in images or video. Our learning and classification framework comfortably exceeds real-time speeds though the implementation requires temporal sub-sampling due to the expense of our CPU based SIFT extraction. This engineering consideration could be addressed in future work.

7. REFERENCES

[1] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[3] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computuer Vision and Image Understanding (CVIU)*, vol. 106, no. 1, pp. 59–70, Apr. 2007.

[4] L. Fei-Fei, R. Fergus, and P. Perona, "A bayesian approach to unsupervised one-shot learning of object categories," in *Proc. International Conference on Computer Vision (ICCV)*, 2003, pp. 1134–1141.

[5] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 28(4), pp. 594–611, 2006.

[6] J. Yang, R. Yan, and A. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *Proc. ACM Multimedia*, 2007, pp. 188–197.

[7] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *Proc. International Conference on Computer Vision (ICCV)*, 2011, pp. 2252–2259.

[8] C. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.

[9] N. FarajiDavar, T. De Campos, J. Kittler, and F. Yan, "Transductive transfer learning for action recognition in tennis games," in *Proc. VECTaR Workshop, at International Conference on Computer Vision (ICCV Workshops)*, 2011, pp. 1548–1553.

[10] A. Kembhavi, B. Siddiquie, R. Miezianko, S. McCloskey, and L. Davis, "Incremental multiple kernel learning for object recognition," in *Proc. International Conference on Computer Vision (ICCV)*, 2009, pp. 638–645.

[11] J. Kim and J. Collomosse, "Semi-automated video logging by incremental and transfer learning," in *Proc. Intl. Workshop on Image and Audio Analysis for Multimedia Interactive Services (WIAMIS 2013)*, Paris, July 2013.

[12] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, pp. 1579–1619, September 2005.

[13] U. Shalit, D. Weinshall, and G. Chechik, "Online learning in the embedded manifold of low-rank matrices.," *Journal of Machine Learning Research*, vol. 13, pp. 429–458, 2012.

[14] T. Deselaers and V. Ferrari, "Visual and semantic similarity in imagenet," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1777–1784.

[15] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. International Conference on Computer Vision (ICCV)*, 1999, vol. 2, pp. 1150–1157 vol.2.

[16] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[17] Z. Liu, Y. Wang, and T. Chen, "Audio feature extraction and analysis for scene segmentation and classification," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 20, no. 1-2, pp. 61–79, 1998.

[18] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proc. International Conference on Machine Learning (ICML)*, 2005, pp. 89–96.

[19] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, USA, 2007.

[20] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Tech. Rep., Microsoft Research (MSR-TR-98-14), 1998.