Video event detection using subclass discriminant analysis and linear support vector machines

Nikolaos Gkalelis, Damianos Galanopoulos, Vasileios Mezaris

Information Technologies Institute / Centre for Research and Technology Hellas

TRECVID 2014 Workshop, Orlando, FL, USA, November 2014

Overview

- Introduction
- Machine learning for MED
  - Proposed method outline
  - SVMs & their time complexity
  - Proposed solution: SRKSDA+LSVM
- Experimental evaluation
  - On older datasets: TRECVID MED 2010
  - On older datasets: TRECVID MED 2012 (Habibian subset)
  - TRECVID MED 2014 Runs
- Conclusions – Future Work

Introduction

- Video understanding is a very important technology for many application domains, e.g., surveillance, entertainment, WWW
- The explosive increase of video content has brought new challenges on how to effectively organize these resources
- One major problem is that conventional classifiers are difficult to scale on this vast amount of features resulted from video data
- More efficient computational approaches are necessary to speed up current approaches

Proposed method - outline

- Method outline and innovation
  - Video representation in a high-dimensional feature space (Fisher Vectors of dense trajectories, and more)
  - Learn a very low dimensional subspace of the original high dimensional space using a Kernel DA method
  - Learn the separating hyperplane in the new subspace using LSVM
  - A new fast SRKSDA algorithm and an SRKSDA-LSVM combination are proposed for event detection
- Advantages
  - Proposed SRKSDA is much faster than traditional kernel subclass DA
  - SRKSDA projects data to a lower dimensional subspace where classes are expected to be linearly separable
  - LSVM is applied in the resulting subspace, providing faster responses and improved event detection performance

Support vector machines

- Training set

$$U = \{(\mathbf{x}_i, y_i), i = 1,\ldots,N\}, \mathbf{x}_i \in R^F, y_i \in \{-1,+1\}$$

- Primal formulation

$$\min_{\mathbf{w},b} ||\mathbf{w}||^2 + C \Sigma_i \xi_i \quad \text{s.t.} \quad y_i(w^T \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

- Dual formulation

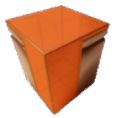
$$\max_{\mathbf{a}} \mathbf{1}^T \mathbf{a} - 0.5\, \mathbf{a}^T \mathbf{H} \mathbf{a} \quad \text{s.t.} \quad \mathbf{y}^T \mathbf{a} = \mathbf{0}, \mathbf{a} - C\mathbf{1} \leq 0, \mathbf{a} \geq 0$$

  where $\mathbf{a} \in R^N$ are the dual variables, and matrix $H = [H_{i,j}]$ is defined as $H_{i,j} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$

- Classification

$$f(\mathbf{x}) = \text{sgn}(\Sigma_p a_p y_p \mathbf{x}^T \mathbf{x}_p + b)$$

  where $U_{SV} = \{(\mathbf{x}_p, y_p), p = 1,\ldots,N_{SV}\}$ is the set of support vector (SVs) - the subset of the training set that actively participates in classifier's definition



SVM time complexity

- Both primal and dual formulations are quadratic programming (QP) problems with F or N variables respectively (F = feature vector dimensionality, N = training observations)
- Thus, SVM training time complexity with traditional QP solvers is $O(NF^2 + F^3)$ or $O(FN^2 + N^3)$ using the primal or dual formulation respectively
- As shown in [1] exploiting the relation between the primal and dual formulation for both cases the complexity is reduced to $O(\max(N,F) \min(N,F)^2)$
- Training time in typical SVM problems is very large, e.g., in MED, $F > 100000$, $N > 5000$, and thus, $FN^2 > 0.25\ 10^{13}$

[1] O. Chapelle, “Training a support vector machine in the primal”, Neural Comput., vol. 19, no. 5, pp. 1155–1178, May 2007.

SVM time complexity

- The special structure of SVM formulation is usually exploited in order to devise efficient algorithms, e.g., LIBSVM uses a SMO type algorithm
- In these implementations the number of SVs play a critical role in training time complexity (and of course in testing time as they are used to define the classifier) [2]
- The SVM training procedure yields many SVs when:
  - Data classes are non-linearly separable
  - High dimensional feature vectors are used (curse of dimensionality: phenomena described in high dimensional spaces require more parameters (in our case SVs) to capture their properties)

[2] D. Decoste and B. Scholkopf, “Training invariant support vector machines”, Mach. Learn., vol. 46, no. 1-3, pp. 161–190, Mar. 2002.

Proposed solution: Nonlinear subclass Discriminant Analysis (DA) plus LSVM

- Apply nonlinear subclass DA
  - A low-dimensional subspace of the original high-dimensional space is derived, discarding noise or irrelevant (w.r.t. classification) features
  - Data nonlinearities are (to the greatest possible extend) removed - classes are expected to be linearly separable in the resulting subspace
- LSVM is trained in the resulting DA subspace

➢ LSVM solves a (almost) linearly separable problem in a low-dimensional space, thus, a small number of SVs is necessary

  - Improved training/testing computational complexity
  - Improved generalization performance
  - Less training observations are required to learn the separating hyperplane

Proposed solution: Nonlinear subclass Discriminant Analysis (DA) plus LSVM

- The main computational effort is “moved” to the DA method → we need to do this efficiently!
- Conventional nonlinear subclass DA methods identify the transformation matrix $\mathbf{\Gamma}$ that optimizes the following criterion

$$\text{argmax}_{\mathbf{\Gamma}} \text{tr}((\mathbf{\Gamma}^T\mathbf{K}\mathbf{A}\mathbf{K}\mathbf{\Gamma})^{-1}(\mathbf{\Gamma}^T\mathbf{K}\mathbf{K}\mathbf{\Gamma}))$$

- This optimization is equivalent to the following generalized eigenvalue problem

$$\mathbf{K}\mathbf{A}\mathbf{K}\mathbf{\Gamma} = \mathbf{K}\mathbf{K}\mathbf{\Gamma}\mathbf{\Lambda}$$

Proposed solution: Nonlinear subclass Discriminant Analysis (DA) plus LSVM

- Identifying $\mathbf{\Gamma} \in R^{N \times H-1}$ with conventional DA requires the eigenvalue decomposition of two N x N matrices (**KAK**, **KK**) ⟶ very expensive for large-scale datasets (in MED usually N > 5000)
- SRKSDA alleviates this problem:
  - eigenvalue decomposition of a H x H matrix (H << N, e.g. in MED, H = 2 or 3), and
  - solving a N x N linear system (done very efficiently using Cholesky factorization)
- In TRECVID datasets, SRKSDA+LSVM has the following advantages in comparison to LSVM
  - It is 1 to 2 orders of magnitude faster during training with fixed parameters
  - The overall training time is approximately 1 order of magnitude faster when a cross-validation procedure is necessary to learn the parameters
  - It provides an equivalent or better MAP performance

Experimental evaluation

- SRKSDA+LSVM is compared with LSVM and KSVM
- SRKSDA is implemented in Matlab
- For KSVM and LSVM the LIBSVM library is used
- Experiments run on an Intel i7 3.5-GHz PC
- Parameter identification (σ, C); σ = RBF scale, C = SVM penalty
  - SRKSDA+LSVM, KSVM: 13 x 1 search grid is applied (fixed C is used)
  - LSVM: 4 x 1 search grid is applied for identifying C
  - Cross-validation procedure with 2 random partitions of development set at each CV cycle
  - Partitioning : 70% training set, 30% test set
- Note that using a 2D search grid to find the best C (in addition to σ) has negligible computational cost for SRKSDA+LSVM (after SRKSDA, LSVM operates in a 2 or 3 dimensional space), while it is very expensive for KSVM

Experimental evaluation on older datasets: MED 2010

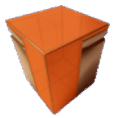
- 3 events, 1745 dev. videos, 1742 eval. videos
- Motion visual information is used: Dense trajectory (DT) features (HOG, HOF, MBHx, MBHy), Fisher Vector (FV) encoding with 256 GMM codewords; motion features are concatenated yielding a 101376-dimensional feature vectors per video
- Training complexity assuming traditional QP solver $O(FN^2)$ or $O(NF^2)$:
  - LSVM: N = 1745, F = 101376 : $FN^2 \approx 0.1\ 10^6\ 0.3\ 10^6 = 0.3\ 10^{12}$
  - LSVM (in SRKSDA+LSVM): N = 1745, F = 3 : $NF^2 \approx 1745\ 9 = 0.16\ 10^5$
  - SRKSDA training time is negligible
- Experimental results:

	LSVM			KSVM			SRKSDA+LSVM		
	AP	Train (min)	Test (min)	AP	Train (min)	Test (min)	AP	Train (min)	Test (min)
T01	**52.6%**	68.8	1.8	47.6%	398.1	1.4	51.9%	**10.7**	**0.3**
T02	75.9%	60	2.2	74.8%	341	4	**76.4%**	**10.9**	**0.2**
T03	39.8%	82.4	1.7	40.7%	376.7	3.7	**40.9%**	**11.1**	**0.1**
AVG	56.1%	70.4	1.9	54.3%	371.9	3	**56.4%**	**10.9**	**0.2**

Experimental evaluation on older datasets: MED 2012 (Habibian subset)

- 325 events, 8840 dev. videos, 4434 eval. videos
- Motion visual information is used: DT, FV encoding, 256 GMM codewords; concatenation yields a 101376-dimensional feature vectors per video
- Complexity assuming traditional QP solver $O(FN^2)$ or $O(NF^2)$:
  - SVM: N = 8840, F = 101376 : $FN^2 \approx 0.79\ 10^{13}$
  - LSVM (in SRKSDA+LSVM): N = 8840, F = 3 : $NF^2 \approx 8840\ 9 = 0.79\ 10^5$
- Computational cost for learning (using fixed parameters) and testing SRKSDA+LSVM is 1 to 2 orders of magnitude faster than LSVM (see example results on event E024)

	E024			
	Nsv	Niter	Train (min)	Test (min)
KSVM	3967	4767	547.6	38.7
LSVM	995	2066	91.8	9.5
SRKSDA+LSVM	**54**	**27**	**3.2**	**1.5**



Experimental evaluation on older datasets: MED 2012 (Habibian subset)

- Experimental results:

	LSVM			KSVM			SRKSDA+LSVM		
	AP	Train (min)	Test	AP	Train (min)	Test (min)	AP	Train (min)	Test (min)
E01	59.5%	356.2	8.1	**62.7%**	2137.1	6.9	62.5%	**57**	**1.6**
E02	14.9%	573.6	12.7	**15.3%**	3602.8	29.2	14.3%	**67.5**	**1.4**
E03	**46.5%**	306	5.9	44.3%	1665.4	8.5	42.3%	**64.7**	**1.9**
E04	66.3%	288.8	5.7	61.4%	1402.7	13.6	**66.6%**	**55.8**	**1.5**
E05	29.6%	397.2	8.4	**30%**	2414.4	17.3	29.4%	**55.5**	**1.5**
E06	27.2%	471.8	10.9	**28.2%**	2752.6	16.2	27.6%	**55.3**	**1.4**
E07	24.1%	510.8	10.2	20.8%	3169.6	10.1	**27%**	**56.5**	**1.5**
E08	58.9%	216.9	4.9	56.3%	971.2	3.8	**59%**	**54.2**	**1.6**
E09	**44.7%**	367.9	8	43.4%	2161.7	14.4	43.4%	**56.4**	**1.5**
E10	38.4%	499.3	10	**41%**	2927.2	15.1	39.3%	**56**	**1.4**

MED 2012 (Habibian subset) – contin.

	LSVM			KSVM			SRKSDA+LSVM		
	AP	Train (min)	Test (min)	AP	Train (min)	Test (min)	AP	Train (min)	Test (min)
E11	28.3%	527.5	10.5	**32.5%**	2609.1	19.6	31.4%	**56.9**	**1.6**
E12	51.1%	305.7	7.1	53.9%	1679.2	10.3	**54.7%**	**55.2**	**1.5**
E13	68.3%	188.3	4.2	67.3%	1010.4	3.5	**70.7%**	**55.7**	**1.5**
E14	**51.4%**	357.1	8	50.1%	1991.4	8	**51.4%**	**57.8**	**1.6**
E15	**61.1%**	439.5	8.7	60.1%	2440.9	14	57%	**55.5**	**1.6**
E21	53.1%	262.5	5.3	54.2%	1772.5	5.4	**56%**	**55.6**	**1.5**
E22	21.7%	342.3	7.6	**24.6%**	2170.3	8.6	24%	**56.6**	**1.6**
E23	75.2%	204.9	4.2	76.9%	1158.3	3.6	**80.5%**	**54.8**	**1.6**
E24	12.3%	439.7	9.5	11.5%	3041.3	38.7	**12.4%**	**55.7**	**1.5**
E25	16.9%	376	8.6	**18.9%**	2280.7	12	18.5%	**56.6**	**1.5**
E26	16.8%	308.3	6.2	**17.9%**	1897.1	8.1	**17.9%**	**30.1**	**1.5**
E27	63.4%	297.7	5.9	67.5%	1895.7	9.9	**69.4%**	**55.3**	**1.4**
E28	40.1%	294.8	5.9	41.2%	1846.4	7.3	**47.9%**	**56.9**	**1.6**
E29	37.6%	257.4	4.7	31.9%	1592.8	16.7	**39.8%**	**55.5**	**1.6**
E30	18.3%	354.6	7.8	**21.1%**	2304.5	11.7	20.2%	**55.8**	**1.5**
AVG	41%	357.7	7.56	41.3%	2115.8	12.5	**42.5%**	**55.7**	**1.5**

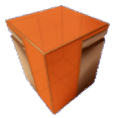
MED 2014 Runs

- 20 PS events, 10 AH events, approx. 7000 dev. videos, 32000 eval. videos
- Visual information is used:
  - Static: 1 keyframe every 6 secs, 4 local descriptors (SIFT, opponentSIFT, RGB-SIFT, RGB-SURF), VLAD encoding, random projection to 4000-dimensional vectors, concatenation yielding a 16000-dimensional feature vector per video
  - Model vectors: 346 SIN concept detectors for each of the above local descriptors; averaging model vectors of 4 local descriptors and over entire video, resulting to a 346-dimensional feature vector per video
  - Motion: DT, FV encoding, 256 GMM codewords; concatenation of DT features yields a 101376-dimensional feature vector per video
  - The feature vectors of different visual modalities are concatenated yielding a 117722-dimensional feature vector per video
- SRKSDA+LSVM method is used for event detection

	PS / 010Ex	AH / 010Ex	PS / 100Ex	AH / 100Ex
MAP	15.1%	16.2%	30.3%	30.2%

Conclusions – Future Work

- SRKSDA+LSVM is much faster than conventional LSVM (1 to 2 orders of magnitude)
- At the same time, it provides roughly equivalent (most often, better!) performance than LSVM, KSVM
- It can learn automatically useful dimensions of the feature space without the need to e.g. manually select the concepts that are most relevant with the target event (hence the good results in the AH subtask; we didn't use any knowledge about the PS events when building our system)
- Though SRKSDA+LSVM is applied here to the event detection problem, it is a generic, widely applicable machine learning method
- We used in MED'14 only visual features; we assume that also using audio, text would further increase our performance
- We are currently working on further reducing the computational complexity of SRKSDA, and on exploiting SRKSDA+LSVM in other analysis problems.



Questions?

More information and contact:
(incl. in relation to technology / software licensing and potential collaborations)
http://www.iti.gr/~bmezaris
bmezaris@iti.gr

To cite the work presented in these slides, please cite the corresponding paper:
N. Gkalelis, F. Markatopoulou, A. Moumtzidou, D. Galanopoulos, K. Avgerinakis, N. Pittaras, S. Vrochidis, V. Mezaris, I. Kompatsiaris, I. Patras, "ITI-CERTH participation to TRECVID 2014", Proc. TRECVID 2014 Workshop, Orlando, FL, USA, November 2014.