Mining visual actions from movies

Adrien GAIDON[1], Marcin MARSZALEK[2], Cordelia SCHMID[1]

[1] LEAR - INRIA Grenoble, LJK
[2] Visual Geometry Group - University of Oxford

Visual human actions

- Human actions: major visual events in movies, news, ...
- Real-world videos: complex imaging conditions, large intra-class variations, visually ill-defined concepts




- To help the analysis of large amounts of videos we:
  - discover what actions are performed
  - collect action samples for action recognition systems

→ Mining actions from movies using text and vision

Mining actions from movies (1)

- Why movies?
  - Large source of high-quality, realistic videos
  - Precise textual description: transcript (screenplay)
- Dataset: *Buffy the vampire slayer* TV-series
  - ➔ 39 DVDs, 144 episodes, 100 hours of video, 10^7 frames

Mining actions from movies (2)

- Summary of our approach
  1. Synchronize the transcripts with the videos
  2. Segment the movies into short clips
  3. Mine actions from transcripts
  4. Rank text-retrieved actions by visual relevance
- Main contributions
  - Ranking text-mined actions by visual consistency
  - Handling weak supervision obtained automatically
  - Iterative re-ranking using regression: *iter-SVR*

Related work

- Realistic videos
  - Temporal segmentation of TV-series and character naming [Cour ' 08]
  - Supervised action classification [Laptev ' 08, Marszalek ' 09]
- Similar approaches
  - Build collections of images for specific object classes [Berg ' 06, Fergus ' 05, Li ' 07, Schroff ' 07]
  - Naming characters in images [Berg ' 04, Ozkan ' 06] and videos [Everingham ' 06]

Outline

Text-based mining of actions from videos

- Temporal segmentation
- Mining actions

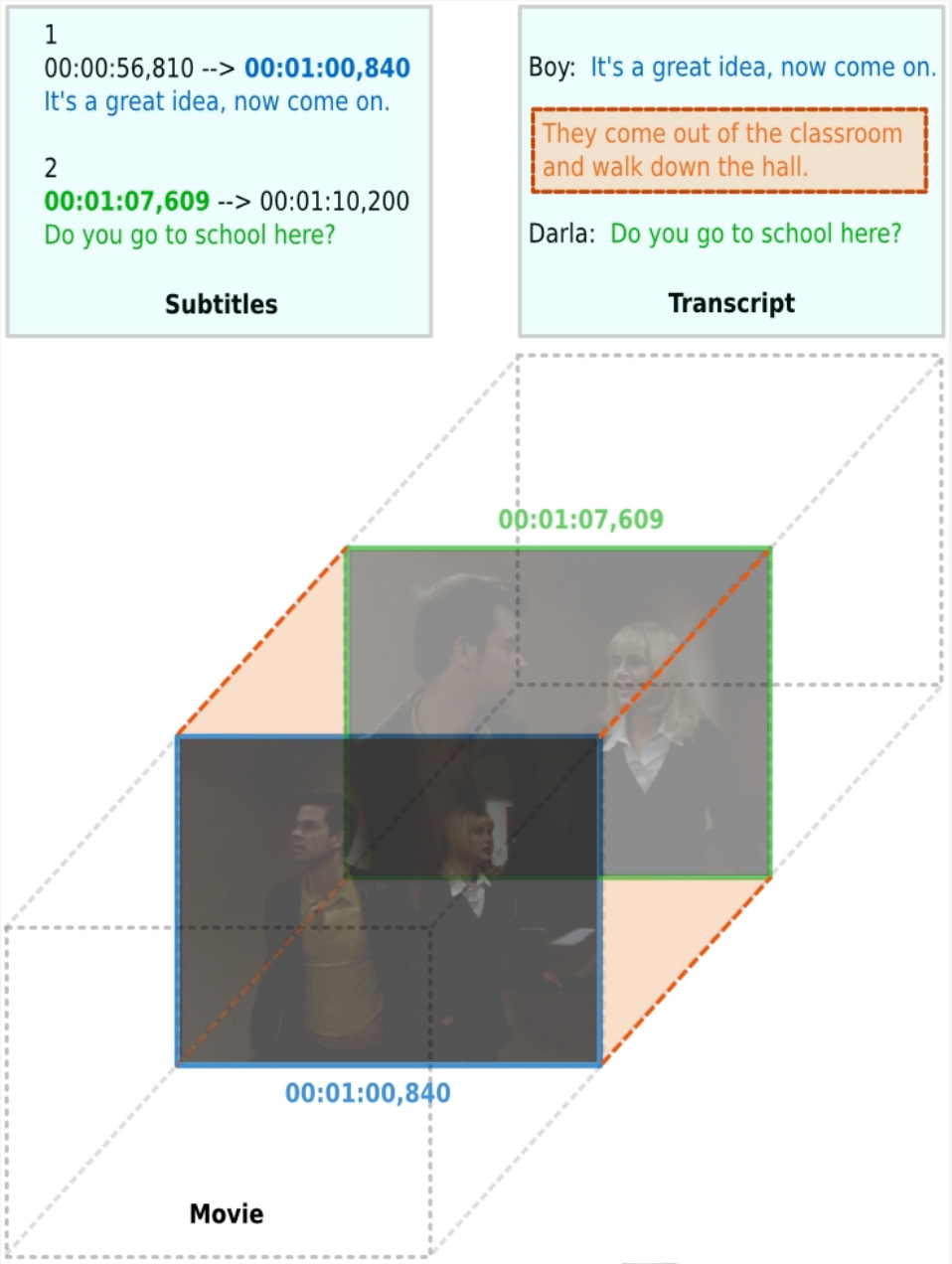
Ranking action samples by visual consistency

- Visual representation and consistency
- Unsupervised estimation of inconsistency
- Ranking with weak supervision

Experiments on Buffy

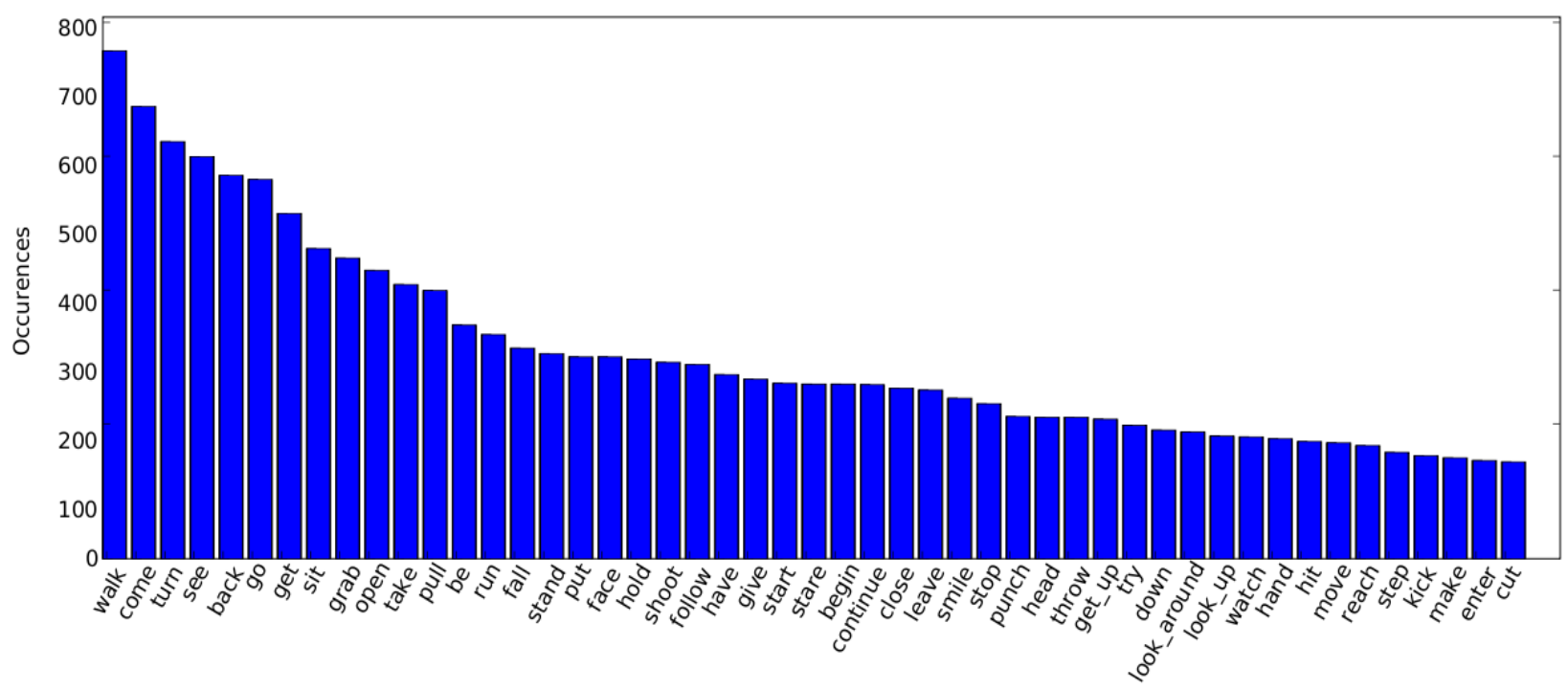
Text-driven temporal segmentation

- Align transcripts with videos by using subtitles [Everingham ' 06]
- Decompose the videos into short clips:
  - **clip**: video between two consecutive subtitles
  - **associated text**: transcript between the dialog lines



Text mining of actions

- Action in text: verb
- Action examples: clips whose textual description contain the corresponding verb [Link-grammar, Lafferty ' 92]
- Results on all of the “Buffy” episodes:



Outline

Text-based mining of actions from videos

- Temporal segmentation
- Mining actions

Ranking action samples by visual consistency

- Visual representation and consistency
- Unsupervised estimation of inconsistency
- Ranking with weak supervision

Experiments on Buffy

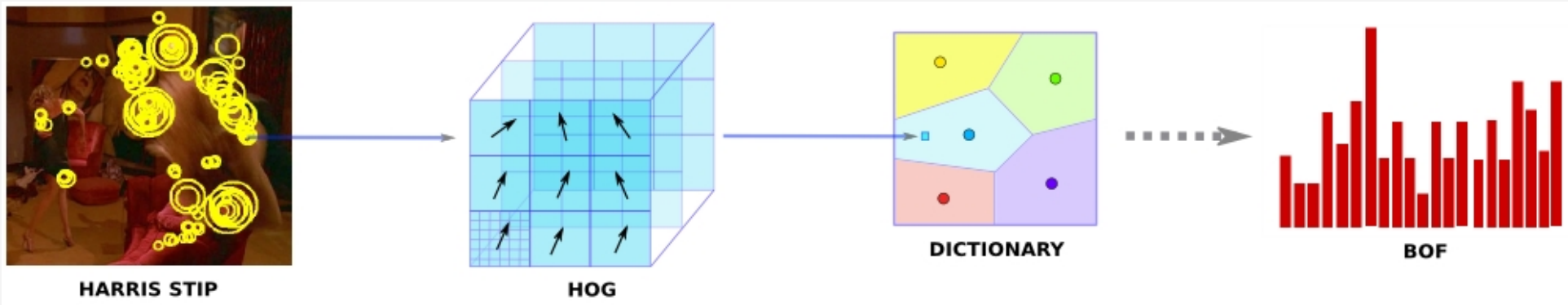
Visual consistency

- Problem:
  - Presence of visually irrelevant examples
- Goal:
  - Visually ranking action clips (retrieved from text)
- Assumption:
  - Relevant documents share common visual characteristics
- Consequence:
  - Retrieval errors are inconsistent samples
  - Relevance can be approximated by consistency

→ **Ranking by visual consistency**

Visual representation

- Bag of spatio-temporal visual words
  - Spatio-temporal Harris interest points [Laptev ' 05]
  - Description of cuboids using HOG [Dalal ' 05]
  - Bag-of-features using k-means with k=1000



- Visual similarity: χ^2 kernel [Zhang ' 07]

Outline

Text-based mining of actions from videos

- Temporal segmentation
- Mining actions

Ranking action samples by visual consistency

- Visual representation and consistency
- **Unsupervised estimation of inconsistency**
- Ranking with weak supervision

Experiments on Buffy

Unsupervised estimation of inconsistency

- Outlier detection
  - Outliers: “samples that deviate markedly from others” or “inconsistent with the remainder of the data” [Barnett ' 94]
  - ➔ Inconsistency score: “outlierness”
- Two different approaches
  - Distance-based: **one-class SVM** [Schölkopf ' 99]
    - ➔ Inconsistency score: distance from a boundary of “normality” around most of the data
  - Density-based: **densest component search** [Ozkan ' 06]
    - ➔ Inconsistent samples: in sparse regions of the feature space

Outline

Text-based mining of actions from videos

- Temporal segmentation
- Mining actions

Ranking action samples by visual consistency

- Visual representation and consistency
- Unsupervised estimation of inconsistency
- Ranking with weak supervision

Experiments on Buffy

Ranking with weak supervision

- Supervised methods
  - Generally more efficient than unsupervised ones
  - Require training data: cost of supervision
- Weak supervision
  - Annotated training data with uncontrolled quality
    - **Positives**: all of the retrieved examples (many FP)
    - **Negatives**: random clips from the whole collection
  - Automatic: no manual intervention
  - Contains errors and uninformative samples

Baseline: binary ν-SVM

- Classification between consistent / inconsistent
  - Training on (noisy) positives + (random) negatives
  - Inconsistency score: distance from separating hyperplane
- Efficiency based on regularization and generalization capabilities of ν-SVM [Vapnik ' 00]

Handling weak supervision (1)

- Learning with weak supervision
  - Weak model
  - Still able to correctly decide in the easiest cases
- Iterative re-labelling and re-training
  1. Learn weak model
  2. Detect most obvious inconsistent samples
  3. Switch their label and re-train: improved model
  4. Iterate until convergence

Handling weak supervision (2)

- Problems
  - Evaluation of “most obvious” inconsistent samples
  - Wrong re-labelling
  - Convergence
- Solution: Regression instead of classification
  - Iteratively learning a regression function quantifying inconsistency
  - Soft re-labelling of all the retrieved samples

Iter-SVR

- Support Vector Regression Machines (SVR)
  [Vapnik ' 96, Drucker ' 97]
- Iterative re-training of SVR with weak supervision

1. Initialization:
   - Retrieved samples $P = \{ (\boldsymbol{x}_i^+ , +1) \}$
   - Random negatives $N = \{ (\boldsymbol{x}_j^- , -1) \}$
2. Learn SVR f (normalize outputs)
3. Replace P by $\{ (\boldsymbol{x}_i^+ , f(\boldsymbol{x}_i^+)) \}$
4. Go back to 2. until convergence

➔ Consistency score: final regressed values

Outline

Text-based mining of actions from videos

- Temporal segmentation
- Mining actions

Ranking action samples by visual consistency

- Visual representation and consistency
- Unsupervised estimation of inconsistency
- Ranking with weak supervision

Experiments on Buffy

Experimental results (1)

- Action examples automatically mined from *Buffy* for six action classes: 'walk', 'fall', 'punch', 'kick', 'kiss' and 'get up'
- Performance (Average Precision) comparison:

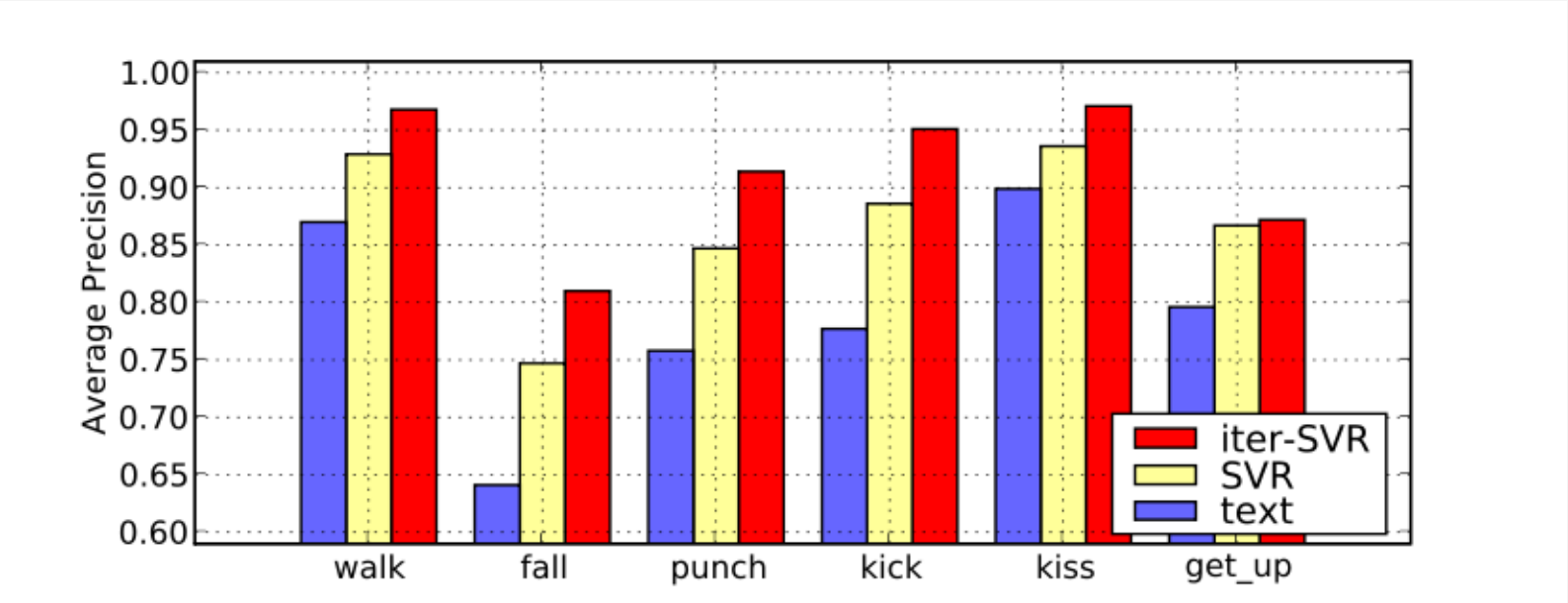
	walk	fall	punch	kick	kiss	get up	Mean
Iterative SVR	**96.8**	**81.0**	**91.4**	**95.1**	**97.1**	87.2	**91.4**
ν-SVM	96.4	77.6	89.7	94.5	96.6	86.9	90.3
One class SVM	93.6	76.5	91.0	93.2	95.1	85.8	89.2
Densest component	93.2	73.6	90.7	93.0	94.6	**89.2**	89.1
Text	87.0	64.1	75.8	77.7	89.9	79.6	79.0

- → Best method: ***iter-SVR***
- → Text-based retrieval always improved by using vision

+12.4% on average, up to +16.9% for 'fall'

Experimental results (2)

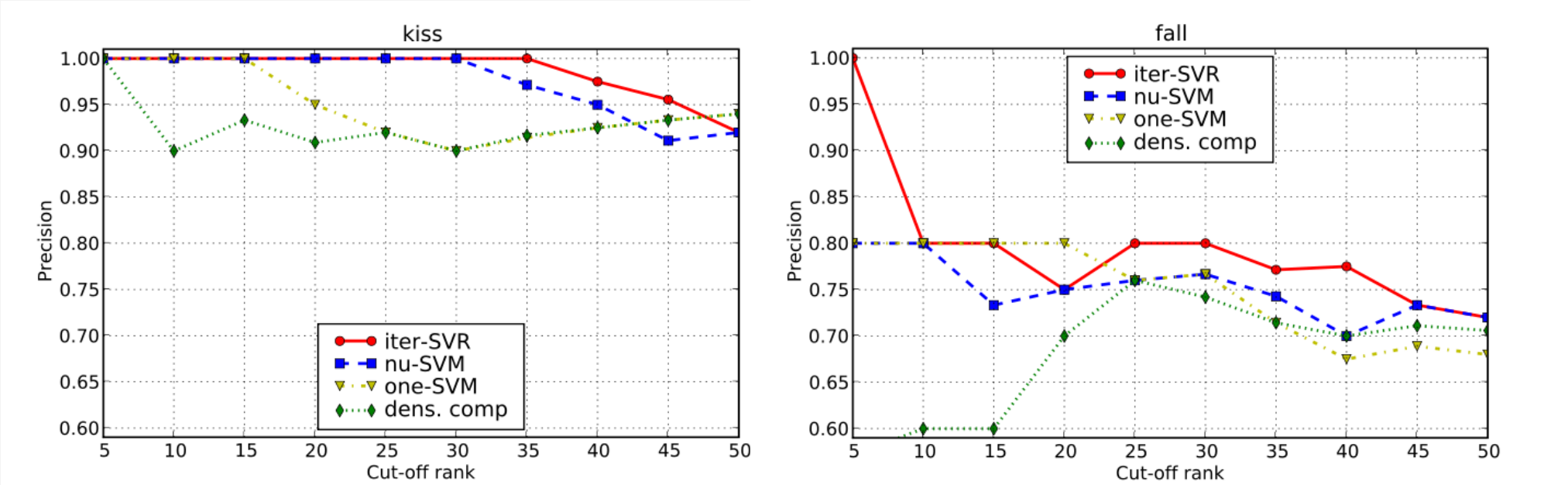
Comparison of text-based results, the SVR without re-training (1st iteration) and our *iter-SVR* approach:



➔ Strong improvement due to iterative re-ranking
(+4.4% on average between SVR and *iter-SVR*)

Experimental results (3)

Precision at the top N ranks:



- High precision at low recall
- Weakly supervised outperforms unsupervised

Experimental results (4)

- Clips automatically retrieved and ranked with *iter-SVR* for the action 'walk'

Conclusion

- Contributions
  - Automatic approach to mine actions from large real-world video collections using visual consistency and text
  - Weak supervision: *iter-SVR* algorithm, inexpensive, outperforms unsupervised methods
- Future work
  - Finer textual and visual representations
  - Joint model text/vision
  - Action localization (spatial and temporal)

Thank you for your attention

Any questions?

Appendix

Evaluation procedure

- Actions automatically retrieved from the *Buffy* episodes and their associated transcripts
- Six manually selected action classes: 'walk', 'fall', 'punch', 'kick', 'kiss' and 'get up'
- Manual annotation of the 100 shortest clips retrieved for each action:

	walk	fall	punch	kick	kiss	get up
true positives	80	59	72	73	71	74
false positives	12	33	23	21	8	19
unclear (unused)	8	8	5	6	8	7

- 100 random negatives sampled from the whole collection

Experimental results

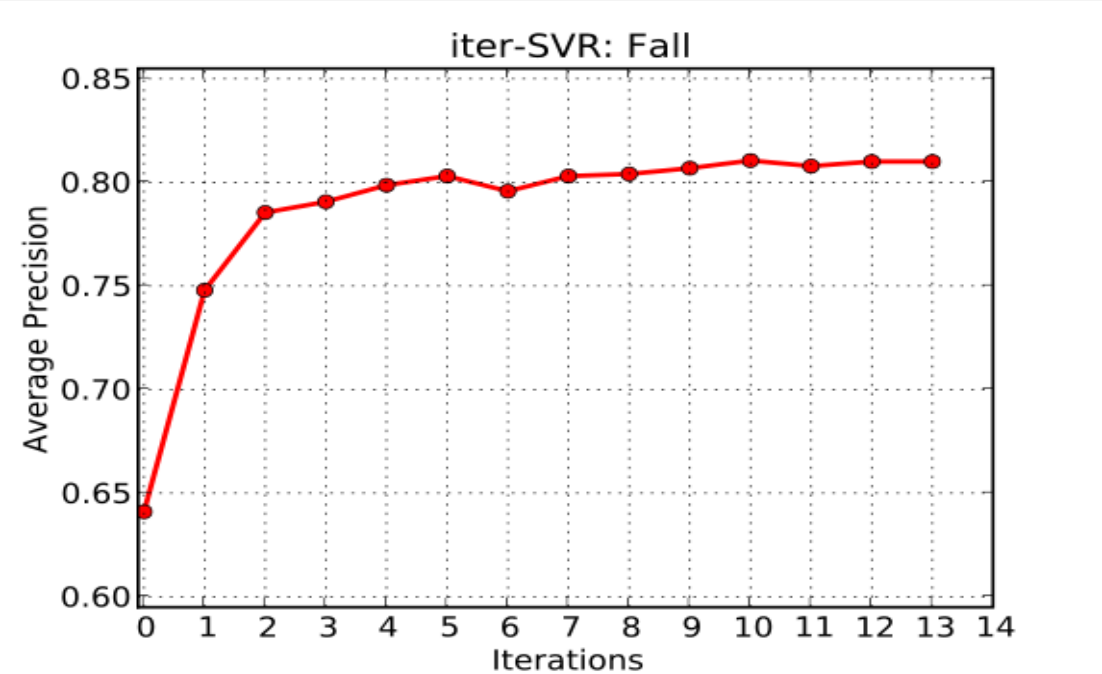
- Key frames of the 1st retrieval result and the 1st false positive (and associated rank) with *iter-SVR*:

- False positives:
  - Confusion at the text level
  - Visually ambiguous: multiple events, complex motions, partial observations, misleading context,...

Convergence

In our experiments:

- Fast convergence (less than 20 iterations)
- Last iteration: best one

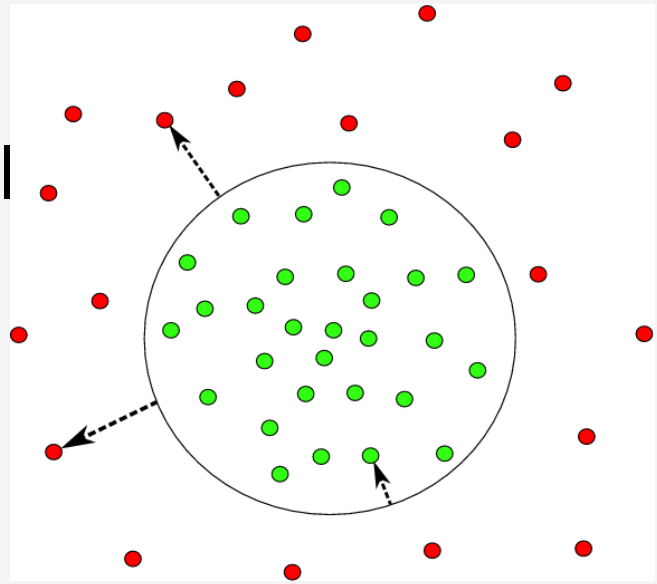


Experimental results

- Clips automatically retrieved and ranked with *iter-SVR* for the action fall

One-class SVM

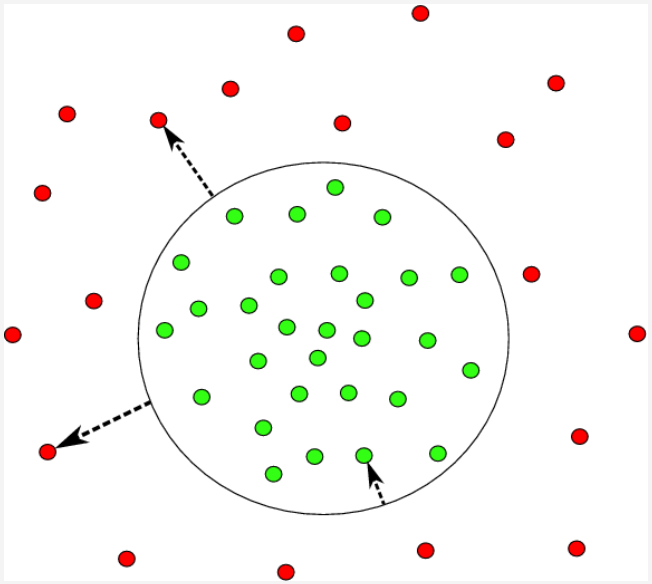
- Unsupervised estimation of the support of the underlying distribution of the data:
  - Finding a boundary around a small region capturing most of the data
  - Maximal margin separation of the samples from the origin



➔ Inconsistency score: distance from margin

One-class SVM (2)

- Inconsistency score = distance from margin $\frac{f(\mathbf{x})}{||\mathbf{w}||}$
  - $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x_i}, \mathbf{x}) - b$: SVM confidence value of ***x***
  - $\frac{2}{||\mathbf{w}||}$: margin width
  - x_i: training samples
  - K: kernel (χ^2 kernel in our case)
  - α_i: SVM dual variables
  - b: obtained from KKT conditions

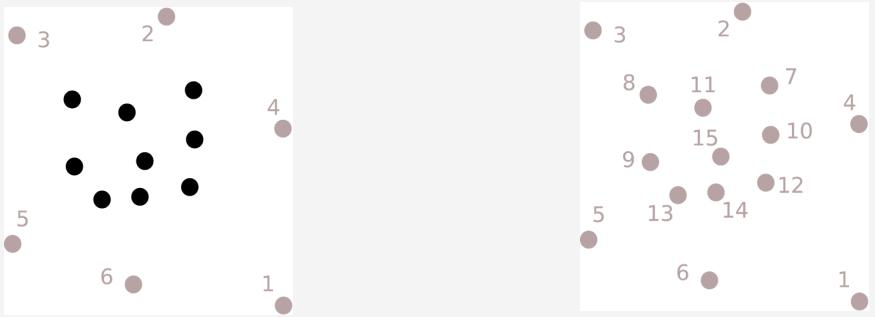


Densest component

- Set of the most consistent samples: densest sub-graph of the similarity graph:
  - Nodes: clips, edge weights: visual similarity between clips
  - Density of a set of samples: average degree of the induced sub-graph

Densest component (2)

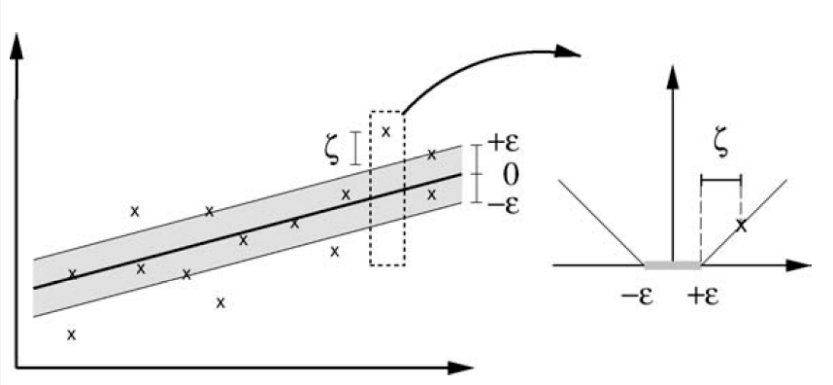
- Algorithm: greedy 2-approximation [Charikar'00]
  - Iteratively deleting the node with minimal degree
  - Densest component = sub-graph with maximal density



- Ranking by inverse pruning order
  - Most inconsistent samples = nodes with minimal degree, *i.e.* lying in the sparsest regions

SVR

- Support Vector Regression Machines (SVR) [Vapnik'96, Drucker'97]
  - Fit data in a “tube” around regression function f
  - “Tube”: ε-sensitive loss function



- Regression function: $f(\mathbf{x}) = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) K(\mathbf{x_i}, \mathbf{x}) - b$
  - *K*: kernel (χ^2 kernel in our case)
  - α_i, α_i^*: dual variables
  - *b*: obtained from KKT conditions