CAP 4453 Robot Vision
Final Presentation

Boyang Wu

UCF ID: bo726798

boyang1724@gmail.com

Studying Computer Science at University of South Florida

Paper 43 - Less is more: zero-shot learning from online textual documents with noise suppression(*)

- Authors - School of Computer Science, The University of Adelaide, Australia

- Ruizhi Qiao,
  - PhD student on Computer Vision and Machine Learning
- Lingqiao Liu
  - Postdoctoral Research Fellow
- Chunhua Shen
  - Professor of Computer Science
- Anton van den Hengel
  - Professor of Computer Science

Notes(*)

- (*) The first two authors contributed to this work equally. Correspondence should be addressed to C. Shen.
- Looked this up and it seems like authors are ordered by degree of involvement in the work, with most active contributors listed first, otherwise they may be listed in alphabetical order instead.
- In Computer Science in general the principal contributor is the first in the author list. However, the practice of putting the principal investigator last in the author list has increasingly become an accepted standard across most areas in science and engineering.
  - https://en.wikipedia.org/wiki/Academic_authorship#Order_of_authors_in_a_list

Abstract

- Classifying visual concepts from associated online textual source (i.e.. Wikipedia) with zero-shot learning
- Add one more factor: Textual representation is usually too noisy for zero-shot learning.
- Researchers want to design a zero-shot learning method that is capable of suppressing noise in text
- Uses an l2,1-norm based objective function which can simultaneously suppress the noisy signal in the text and learn a function to match the text document and visual features.
- Also develop an optimization algorithm to efficiently solve the resulting problem.
- The proposed method significantly outperforms those competing methods which rely on online information sources but with no explicit noise suppression.

Zero-shot learning

- Solving a task without receiving training examples of the task (unlike what we did with Adaboost/TensorFlow in our class)
- It is an effective way for large scale visual classification
  - Do not need to collect training data
- Key component is to find intermediate representation to bridge between seen and unseen classes
- Can learn connection with image features, then transfer connection to unseen classes
- Even unseen classes can be classified through learned connection
- Zero-shot learning can require laborious human work

Introduction

- Develop an automatic zero-shot learning
- Use online text documents like Wikipedia
- Documents are noisier due to excessive wording
- Existing research has already been done on the subject
  - However earlier works show low performance due to noise
- This research takes it a step further by suppressing noise
- How to discard some words but still keep the relevant ones?
  - Use a noise suppression algorithm along with zero-shot learning

Method

- How?
  - Learn from a matrix V to optimize objective
  - Add noise suppression to an existing formula from prior works
  - Lots of equations that come from linear algebra
- Experiments
  - 1st part
    - Evaluate proposed method and compare against previous methods
  - 2nd part
    - Analyze noise suppression of proposed method to find what information in a document is useful for zero-shot learning

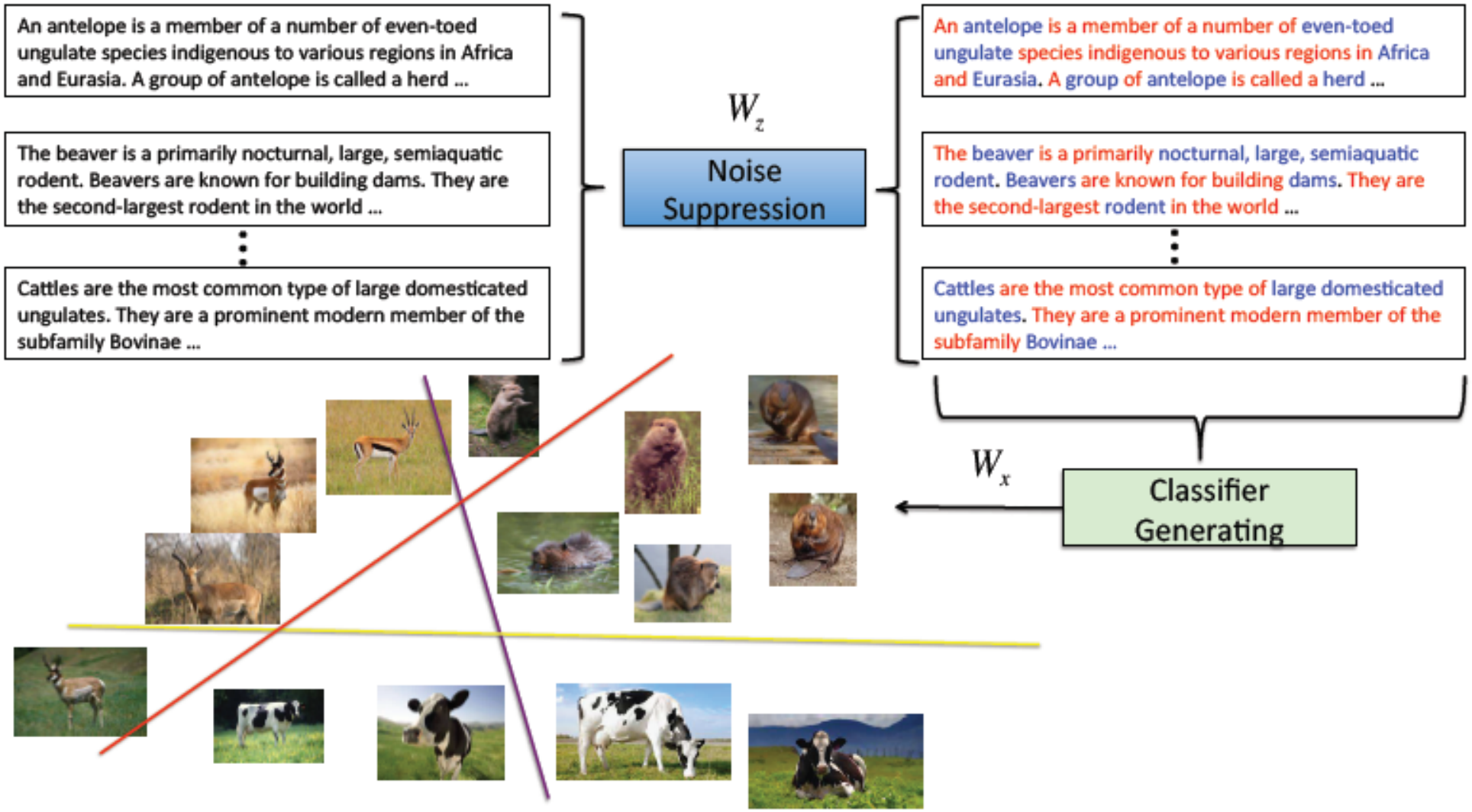

Figure 1. Overview of our zero-shot learning approach. The text representations are processed by the noise suppression mechanism to generate a classifier to detect relevant images and the noisy components of text representations are suppressed to gain better performance.

Conclusion

- Compare results against other prior algorithms
- This approach outperforms all other methods using online text sources
- Only gets beaten by methods that use human specified fine-grain attributes
- Noise has significant impact on zero-shot deep learning
- Some words removed by noise suppression could have more meaning when pieced together
- Three types of words after de-noising
  - Attributes describing the category
  - Words weakly related to the category
  - Non-informative words to humans (could show distribution patterns among categories)

- *Note that due to testing on similar bird species, some are so close in terms of description that Wikipedia articles may not have good enough descriptions.
- Could improve performance by using high quality bird watching articles

Paper 51 - Anticipating Visual Representations from Unlabeled Video

- Carl Vondrick
  - Research Scientist, Google
  - Assistant Professor, Columbia University (Fall 2018)
  - Ph. D from Massachusetts Institute of Technology
- Antonio Torralba
  - Professor of Computer Science and Artificial Intelligence
  - Massachusetts Institute of Technology
- Hamed Pirsiavash
  - Assistant Professor for Computer Vision and Machine Learning
  - University of Maryland, Baltimore County

Abstract

- Researchers want to anticipate actions and objects before they start or appear
  - This is a difficult problem in computer vision
  - Requires using extensive knowledge of the world that is hard to write down
- To do this, the researcher want to use readily available unlabeled video
- Present new framework that uses temporal structure in video to learn to anticipate actions and objects
  - Train deep networks to predict visuals in the future
- Experiment the new idea on two datasets
  - 1s in future and 5s in future

Introduction

- Computer vision that can anticipate actions and objects make robots and machines smart
  - This results in increased convenience for humans
  - Can be used for recommendations, predictions, etc.
- However, creating such an algorithm is extremely difficult
- Humans use accumulated knowledge, but what do machines have?
- Why videos?
  - Having consecutive frames mean image order is already established
- Prior research has been done on this matter, but generally require self-supervision which is expensive to scale

Anticipating Visual Representations from Unlabeled Video

- Goal is to predict future frames within a video
- Use recognition algorithms on forecasted representations
- Used 600 hours of unlabeled video to train the network 1 to 5 seconds in the future
- Also used video from THUMOS (400 hours) to quantify performance
- Forecast actions and objects by applying recognition algorithms
- Finally evaluate the idea on two datasets of human action in television shows and egocentric videos of daily life

Previous work done before

- Prediction with unlabeled video
- Predicting actions, human path, and motions
- Big data (visual)
- Unsupervised learning in vision
- Representation learning

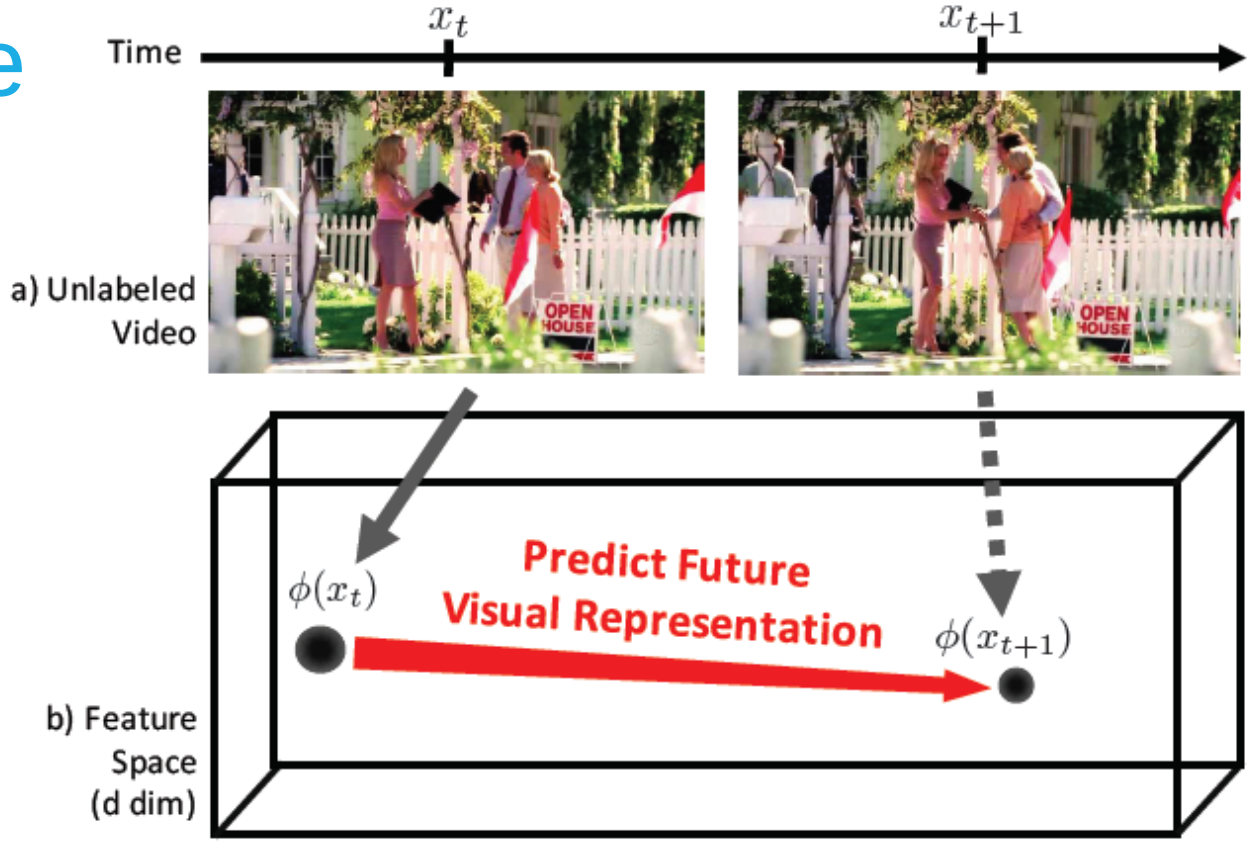

Figure 1: **Predicting Representations:** In this paper, we explore how to anticipate human actions and objects by learning from unlabeled video. We propose to anticipate the visual representation of frames in the future. We can apply recognition algorithms on the predicted representation to forecast actions and objects.

Method

- Use convolution (5 layers)
  - Last layer is output vector – this makes the prediction
- Formulas
  - Euclidean loss to minimize distance between predictions and representation of future frame
- Predicts the last hidden layer of Alexnet
- Can obtain large amounts of data since data does not need to be labeled
- Use deep regression to decide between multiple futures
- Ran on i7 3.4Ghz CPU

Conclusion

- Depending on algorithm settings, action predictions can achieve between 30 and 43% accuracy
- Method is still far from human performance (70-85% accuracy)
- Better than competing computer methods (25-35% accuracy)
- Object prediction is between 20 and 200% better than computer methods
- However, method can help machines anticipate some objects and actions
- Predicting visual representations is better than predicting pixels (difficult) or anticipating label categories (supervision)

Paper 73 - Fast Unsupervised Ego-Action Learning for First-Person Sports Videos

- Kris M. Kitani
  - Cooperative Research Fellow in the Institute of Industrial Science at the University of Tokyo
  - Assistant Research Professor in the Computer Vision Group, Robotics Institute, School of Computer Science
  - Courtesy appointment in the Electrical and Computer Engineering (ECE) department at Carnegie Mellon University
- Takahiro Okabe
  - Ph.D. in Information Science and Technology
  - Department of Artificial Intelligence
  - University of Tokyo

Authors continued

- Yoichi Sato
  - Professor at Institute of Industrial Science
  - University of Tokyo
- Akihiro Sugimoto
  - Professor for Digital Content and Media Sciences Research Division
  - National Institute of Informatics, Tokyo

Abstract

- 1st person POV action cameras are becoming common among sports enthusiasts
- Discover 1st person action categories (called ego-actions)
  - These can be useful for video indexing and retrieval
- In order to learn categories, researchers investigate use of motion-based histograms and unsupervised learning algorithms to cluster video content
- Approach assumes unsupervised scenario, without labeled training videos
- Videos are not segmented, and number of categories are unknown
- Use in-house and public Youtube videos to categorize across various sport genres
- Approach outperforms other topics models in terms of classification accuracy and computational speed
  - 10 minute video can be indexed in under 5 seconds

Introduction

- Action cams are becoming more popular, so there are more videos
- Since there are more videos, indexing and searching for videos may be difficult
- How do we classify and search everything?
  - Say you want to review a third jump of the second trial in a ski jump video
  - This experiment will try to solve that
- Searching can be done to quickly go to the desired location in the video
  - Can be done with color-coded times

Prior work?

- None when looking for experiments on ego-action categorization from 1st person sports videos.
- Only prior/related works are vision-based POV human action for indoor activities with focus on hand gesture recognition.
  - Sign language recognition
  - Context aware gesture recognition
  - Object recognition
  - Hand tracking
- Body sensors have been used in other experiments

Method Basics

- Unsupervised method is the best as to reduce human labor
- Close to real-time processing is needed so users won't have to wait long for results
- Extracted video features should be discriminative enough to differentiate between actions categories but also robust enough to deal with extreme ego-motion
  - Due to the nature of ego-motion (POV movement) there are large amounts of motion parallax, motion blur, rolling shutter, and more within videos
  - However, POV videos only have a single sport per video, and actions would look similar across videos due to the nature of the sport
  - Actions of the same sport share similar image distortion due to similar movements

Method Basics 2

- Use a simple global representation of motion that is robust and discriminative
- Use the Dirichlet process on real-world problems [1]
  - Dirichlet is a stochastic (random) process to find a distribution over distributions.
  - Similar to Gaussian process
  - Distributions are discrete, but cannot be described with finite number of parameters
- Provide a new labeled benchmark dataset for standardized analysis of outdoor POV sports videos
- Ran on Nvidia T40 Tesla GPU

- [1] https://www.stats.ox.ac.uk/~teh/research/npbayes/Teh2010a.pdf

Motion Feature Extraction

Figure 3. Color-coded video time bar indexed by category can be used to find ego-actions of the same category.

Figure 4. Distortion examples: projectiles, motion blur, water.

- POV sports footage is very noisy
- Need to find representation of motion that is robust in comparison to the distortion
- Use optical flow vectors
  - Quick movement can create false flags
  - Observe general direction and magnitude instead for consistency
- Use RANSAC to extract consistent flow vectors
  - This is an iterative method to estimate parameters
- Robustness further added by converting sets of flow vectors to histograms

Motion Feature Extraction 2

- Two types of motions needed to discern
  - Instantaneous motion (direction)
    - Turning you head is more directions
    - Encoded with a 36-bin histogram
      - 4 flow directions
      - 3 flow magnitudes
      - 3 flow variance (difference of flow magnitude compared to average flow magnitude)
  - Periodic motion (frequency)
    - Running and walking is more frequent
    - Done with a Fourier Transform over average flow magnitude
    - Encoded with 16-bin histogram
    - Frequency components are thresholded and normalized
  - Histogram from both are merged to show a final motion histogram

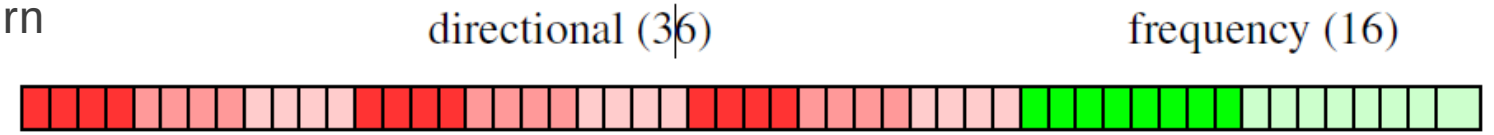

Figure 5. 52-dimensional motion histogram is a concatenation of 36 directional bins and 16 frequency bins.

Dirichlet Process (DP)

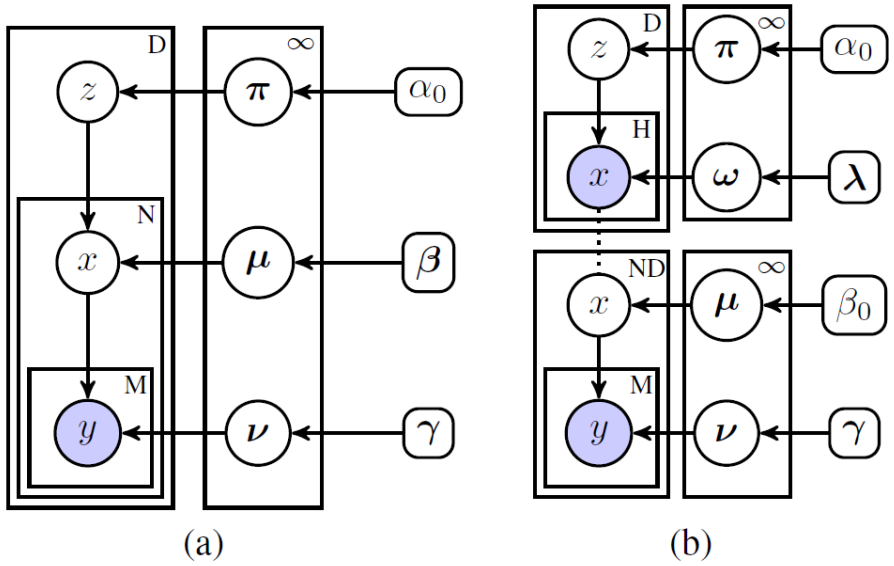

Figure 6. (a) Hierarchical Dirichlet Process Multinomial Mixture, (b) Stacked Dirichlet Process Multinomial Mixture. Graphs drawn using explicit cluster indicators and stick-breaking representation.

- Use a stacked Dirichlet Process Mixture (DPM)
  - Similar to a big urn with many biased K-faced dice
  - Single draw from urn yields one biased K-faced die
  - Expected value of probabilities is defined as $E[\boldsymbol{\pi}] = \{\frac{\alpha_1}{\alpha_0}, \ldots, \frac{\alpha_K}{\alpha_0}\}$

- The Dirichlet distribution is useful when working with K dimensional histograms (2 in this case)
- Histogram is interpreted to be created from multiple passes, so likelihood of histogram becomes a product of probabilities
- Result is another Dirichlet distribution

$$Dir(\boldsymbol{\pi}; \alpha_1, \ldots, \alpha_K) = B(\alpha_1, \ldots, \alpha_K)^{-1} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1} \quad (1)$$

Chinese Restaurant Process (CRP)

- Alternative way to explain Dirichlet Process
- It is a process of generating a probability distribution PI from DP
  - Probability distribution would be like a normal graph or histogram

- CRP and DP both generate a discrete probability distribution PI

$$p(z_{dk}|z_1,\ldots,z_{d-1};\alpha_0) = \begin{cases} \frac{c(k)}{\alpha_0+d}, & k \leq K_d \\ \frac{\alpha_0}{\alpha_0+d}, & k > K_d \end{cases} \quad (3)$$

Stacked Dirichlet Process Mixtures

- Experiment uses a single DPM and posses the results to second DPM to learn ego-action categories
- Stacking makes it so topics are organized by hierarchy, and all tops at each level of hierarchy have the same discrete set of observations (motion histograms)

Task of Inference

- Single video is cut into X equal sized video splices (by time)
- Each video splice is made up of N frames (60 in this case)
- Set of motion histograms Y are generated from each frame in the video then input into the model
- Output Z is another set of motion histograms of ego-action indices
  - These contain ego-action cluster assignment for each video splice
  - Total number of ego-action clusters is then estimated
- Everything is unsupervised learning
  - Motion codebook is learned
  - Ego-action categories are discovered
  - Codebook is learned and histogram accumulated efficiently over each video splice in a single pass of data.

Motion Cookbook

- Output from the first DPM is a sequence of codeword assignments in histogram X
  - Each indicator variable in the histogram has been assigned to a codeword H that increases probability for later
- Outputs Y of all past motion histograms
  - This data can be decomposed into
    - Current prior over cluster assignments
    - Likelihood of observed motion histogram

Lost of difficult formulas

$$\begin{aligned}\mathcal{L} &= p(\boldsymbol{y}_{nd}|\mathbf{Y}^{-nd}, \mathbf{X}^{-nd}, x_{ndh}) \\ &= \int_{\nu_h} p(\boldsymbol{y}_{nd}|\boldsymbol{\nu}_h) q(\boldsymbol{\nu}_h|\mathbf{Y}^{-nd}, \mathbf{X}^{-nd}, x_{ndh}) d\boldsymbol{\nu}_h \\ &= E_q[p(\boldsymbol{y}_{nd}|\boldsymbol{\nu}_h)]\end{aligned}$$

$$E_q[p(\boldsymbol{y}_{nd}|\boldsymbol{\nu}_h)] \propto \prod_m^M \left[\frac{c(h,m) + \beta_0/M}{\sum_{m'} c(h,m') + \beta_0}\right]^{y_{mnd}} \quad (11)$$

$$\begin{aligned}\tilde{\mathcal{L}}_{\boldsymbol{r}} &= \prod_d p(x_d|\mathbf{X'}_{\boldsymbol{r}}^{-d}, \mathbf{Z}_{\boldsymbol{r}}^{-d}) \\ &= \prod_d \sum_k^K p(z_{dk}|\mathbf{X'}_{\boldsymbol{r}}^{-d}, \mathbf{Z}_{\boldsymbol{r}}^{-d}) \\ &\times \int p(\boldsymbol{x}'_d|\boldsymbol{\omega}_k) q(\boldsymbol{\omega}_k|\mathbf{X'}_{\boldsymbol{r}}^{-d}, \mathbf{Z}_{\boldsymbol{r}}^{-d}) d\boldsymbol{\omega}_k \\ &\approx \prod_d^D \sum_k^K p(z_{dk}|\mathbf{X'}_{\boldsymbol{r}}^{-d}, \mathbf{Z}_{\boldsymbol{r}}^{-d}) E_q[p(\boldsymbol{x}'_d|\boldsymbol{\omega}_k)]\end{aligned}$$

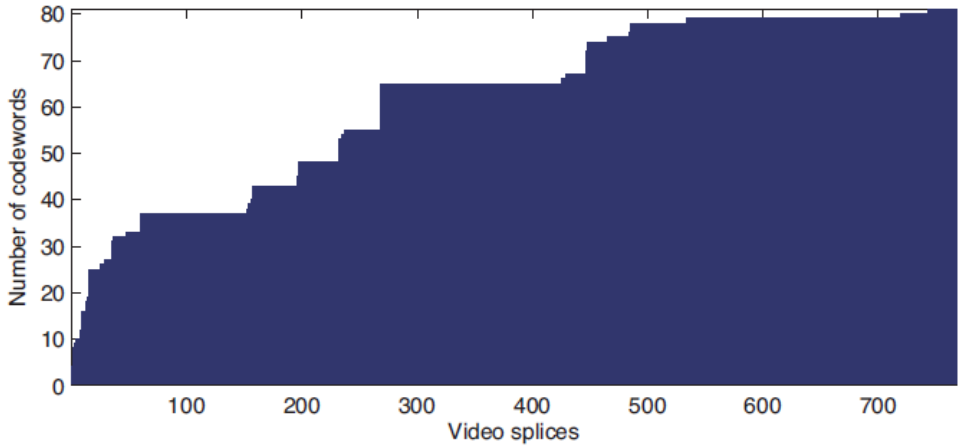

Figure 7. Logarithmic growth of the codebook for the PARK sequence.

Some explanation

- So those formulas are all above me at my current level
- But in general, they are used to calculate various things, such as codewords from the videos, checking over past distributions, and count of flow vectors accumulated from all histograms
- The inference algorithm here adds some new codewords and order is preserved when added to previous codewords
- Codebook length is log bounded in terms of how many codewords it can have

More explanations

- Histograms are again decomposed, then inference is performed for every frame
- Run inference several times to search for more clustering

Figure 8. YouTube sports dataset.

Discovering Ego-action Categories

- Run controlled experiments on many videos to observer performance across various sports
- First choreographed video (QUAD) uses 124 video splices and 11 different ego-actions categories
  - Used to show how design of motion features affect performance
- Second video (PARK) is a 25-minute workout video with 766 video splices and 29 different ego-action categories
- Used 6 Youtube videos (previous page) to check performance
- All sequences were recorded with a GoPro HD

Matching Categories

- Use F-measure for performance
- Found one-to-one correspondence between ground truth and discovered actions
- Did a search to identify best match between single ground truth and discovered action
- Keep repeating until done
- If there are extra discovered categories, these are later appended to the original list but have an F-value of 0 (not used)
- Average F-measure is computed from weighted average of each category

More

- Recall flow direction has 4 directions
- Magnitude has 3 bins and are divided into small, medium and large motion magnitudes.
- Variance between actions were also divided into 3 bins like magnitude

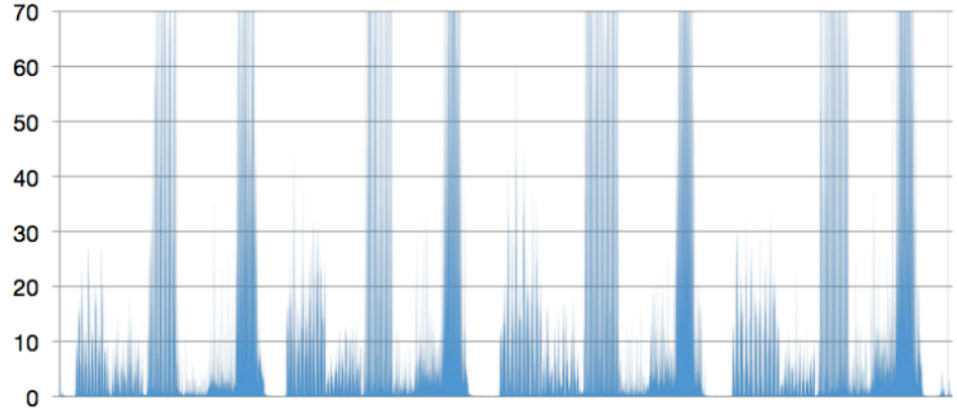

Figure 9. Peaks in flow **variance** produced by actions with extreme changes in acceleration caused by floor impact (*e.g.* jump). Horizontal axis is time and the vertical axis is the variance. The eight peaks are generated by the four repetitions of *jump* and *run*.

Performance

- F-value max of 1.0
- Skiing, surfing, snowboarding average 0.6

- Horseback riding, mountain biking,

  Slope style below 0.6

- Some sports having more discernable actions,
- Thus the result
- Proximity also had effect

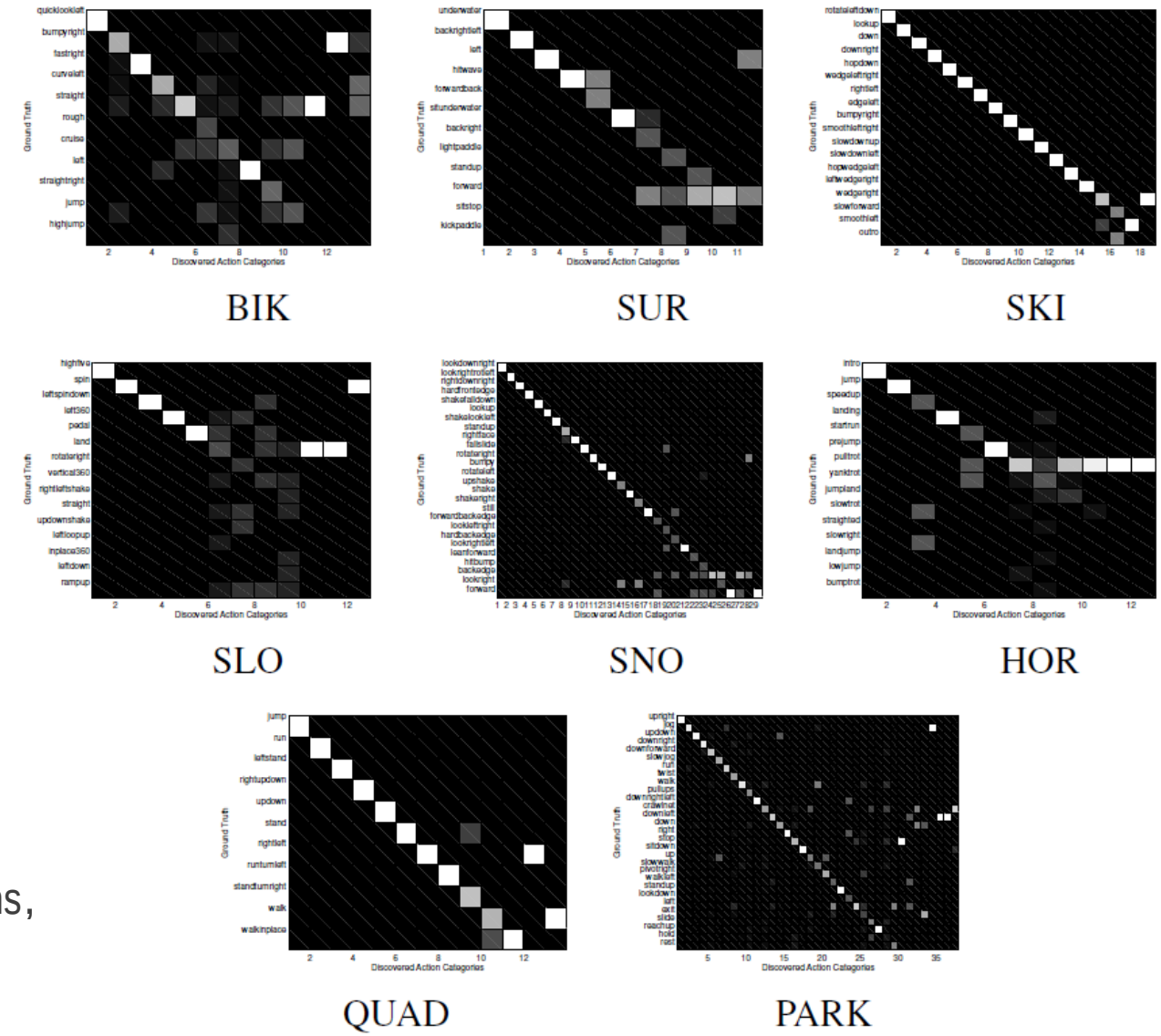

Figure 11. Matching matrix visualization of performance across sports genres. Vertical axis is the ground truth label and the horizontal axis is the discovered ego-action categories. Perfect performance yields an identity matrix.

Comparison

- Online inference test works the best
- Non-DP models underestimate true K value
- Method is second in speed next to Bayes
- Average compute was less than 1 second for
- 2 minutes of video (60 segments)
- DPM variances took significantly more time due to complexity
- All tests performed on 2.66Ghz CPU

Table 3. Detailed comparison for choreographed datasets.

QUAD	F-measure	P	R	K	sec.
DPM-OL	**0.93**	**0.95**	**0.92**	13	0.47
DPM-VI	0.92	0.94	0.92	12	10.12
LDA-VI	0.87	0.89	0.87	11	3.38
PLSA-EM	0.89	0.91	0.89	11	2.88
NBM-EM	0.66	0.59	0.91	5	**0.25**
K-means	0.89	0.89	0.91	9	1.44

PARK	F-measure	P	R	K	sec.
DPM-OL	**0.71**	**0.76**	0.71	37	8.69
DPM-VI	0.61	0.66	0.62	40	73.64
LDA-VI	0.56	0.56	0.66	27	30.99
PLSA-EM	0.53	0.58	0.59	29	63.51
NBM-EM	0.44	0.38	**0.73**	10	**3.75**
K-means	0.53	0.62	0.52	29	25.04

Conclusion

- Introduced a new way to discover ego-action categories from POV sports videos
- Dirichlet process was used to infer motion codebooks and ego-action categories with no training data
- Shown that DPMs can be applied to real-world problems without costing too much compute complexity.
- Online inference can perform on part with other approximate inferences over models, while saving in computational cost
- Vision-based ego-action analysis can be successfully applied to dynamic POV videos

Results

- https://www.youtube.com/watch?v=42pVGWAQOVU
- Video of discovered vs ground truth results from the experiment

Thank you!