Learning Representations for Computer Vision Tasks

Siddhartha Chandra & C. V. Jawahar
CVIT, IIIT Hyderabad

December 10, 2012

Outline

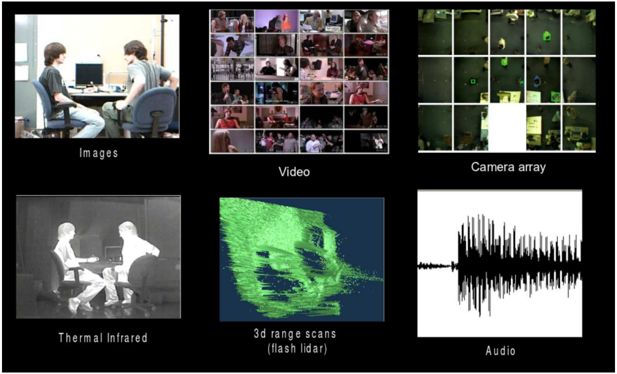
Computer Vision

- Computers still far inferior to Humans.
- Computers see pictures as arrays of numbers.
- Build machines that understand image data.
- Computer Vision is often posed as a Machine Learning problem.
- How to represent the world?

Features in Computer Vision

- Means of representing the world (image data).
- Good features
  - the representation size is manageable
  - most of the relevant information is captured
  - most of the redundancy in data is eliminated
  - invariance to external parameters
- Good Representations are task specific.

Image Data


Images
Video
Camera array
Thermal Infrared
3d range scans
(flash lidar)
Audio

Computer Vision Tasks

- Clustering.
- Action Recognition.
- Visual Classification.

Part-1

Partial Least Squares Kernel for Computing Similarities between Video Sequences

Similarity Kernels

- Computing Similarities is fundamental to many Computer Vision tasks.
- Better Similarity, More Accurate Prediction.
- Lot of Kernels in literature for text, images.
- Kernels for Videos are challenging.

Videos

- 3D
- Spatial and Temporal context.
- Applications to action classification, hand gesture recognition
- Kernels for Videos are challenging.

Hand Gesture Recognition

- Human Computer Interaction.
- Sign Language Interpretation.

Activity Classification

Interacting with a computer

Photographing

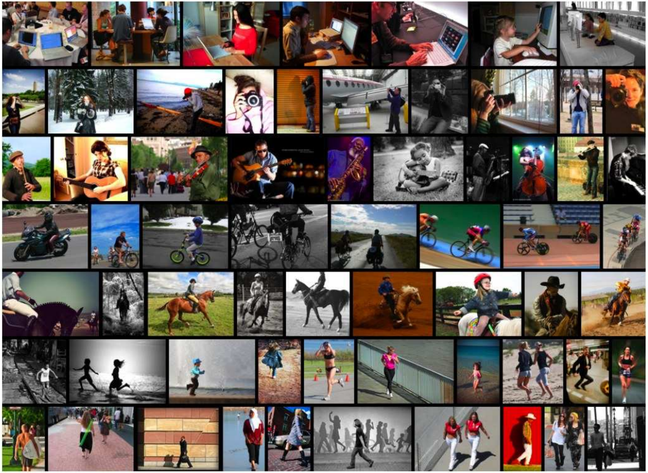
Playing music

Riding bike

Riding horse

Running

Walking



Partial Least Squares

- Modeling relations between sets of observed variables using latent variables.
- Maximizes covariance between two sets of variables.

PLS Kernel for 2D Matrices

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathbf{T}} + \mathbf{E} \tag{1}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathbf{T}} + \mathbf{F} \tag{2}$$

$$\mathbf{U} = \mathbf{T}\mathbf{B} + \mathbf{H} \tag{3}$$

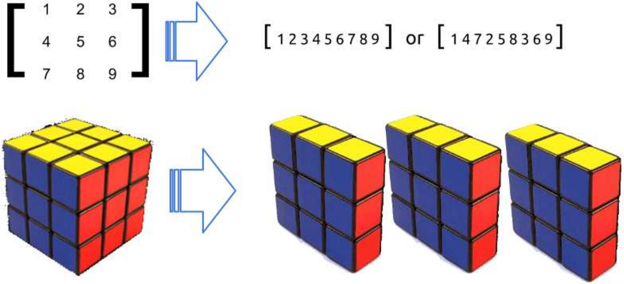
$$\mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{Q}^{\mathbf{T}} + (\mathbf{H}\mathbf{Q}^{\mathbf{T}} + \mathbf{F}) \tag{4}$$

PLS Kernel is defined as the sum of the regression coefficients in B.

Notation

- $\kappa(\mathbf{U}, \mathbf{V})$: PLS similarity between videos **U**,**V**.
- $\beta(\mathbf{P}, \mathbf{Q})$: PLS similarity between matrices **P**,**Q**.
- How to exploit $\beta(\mathbf{P}, \mathbf{Q})$ to compute $\kappa(\mathbf{U}, \mathbf{V})$?
  - Flatten 3D (Tensor) Videos to Matrices

Flattening: Joint Shared Modes



- A Video is a 3^{rd} order tensor $\mathbf{V} \in \mathbb{R}^{x \times y \times t}$.
- 3D Matrix with 3 modes: spatial axes (x, y) and time (t).
- 3 ways of flattening: reordering (x, y) or (x, t) or (y, t).
- Joint shared modes: $\mathbf{V}_{xy}$, $\mathbf{V}_{xt}$ and $\mathbf{V}_{yt}$.

PLS Similarity Kernel for Videos

$$\kappa(\mathbf{U}, \mathbf{V}) = \beta(\mathbf{U}_{xy}, \mathbf{V}_{xy}) + \beta(\mathbf{U}_{xt}, \mathbf{V}_{xt}) + \beta(\mathbf{U}_{yt}, \mathbf{V}_{yt})$$

The similarity between two videos is simply the sum of the similarities between their corresponding joint shared modes.

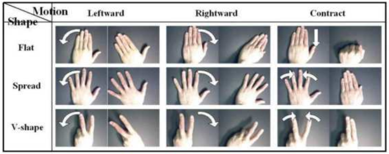
Why PLS?

- PLS extends the multiple linear regression model.
- Intuitive: maximizes covariance.
- More general than multivariate methods (discriminant analysis, principal components regression, and CCA).
- Multivariate methods impose two restrictions
  - latent variables recovered from X^TX and Y^TY matrices, not X^TY, Y^TX
  - $\#$ prediction functions $<$ $\#$ X, Y variables
- PLS
  - also uses X^TY, Y^TX
  - $\#$ prediction functions may exceed $\#$ X, Y variables

Datasets & Pipeline

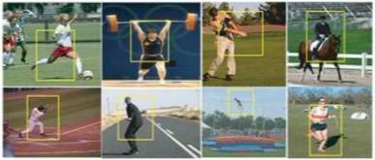
- Datasets
  - Cambridge Hand Gesture Dataset
    - 900 videos, 9 classes, 5 illumination settings
  - UCF Sports Action Dataset
    - 150 videos, 10 classes
- Pipeline
  - PLS Kernels
  - One-vs-Rest SVM per Class.

Cambridge Hand Gesture Dataset



Method	Set 1	Set 2	Set 3	Set 4	Total
Ours	**96%**	**92%**	**96%**	**93%**	**94 ± 2.1%**
TB (Liu et al. 2011)	93%	88%	90%	91%	91 ± 2.4%
PM (Liu et al. 2010)	89%	86%	89%	87%	88 ± 2.1%
DCCA (Kim et al. 2007)	-	-	-	-	85 ± 2.8%
TCCA (Kim et al. 2007)	81%	81%	78%	86%	82 ± 3.4%

UCF Sports Action Dataset



Method	Leave one out Cross Validation
Ours	**93.2%**
TB (Liu et al. 2011)	88%
HDN(Kovashka et al. 2010)	87.27%
OMD (Bregonzio et al. 2010)	86.9%

Conclusions & Future Work

- PLS regression is general and intuitive.
- PLS Kernel is straightforward. Requires no parameter tuning.
- Classification with discriminative PLS Kernels outperforms recent state of the art methods on two popular datasets.
- Insights into regression may reveal other interesting properties of pairs of tensors.

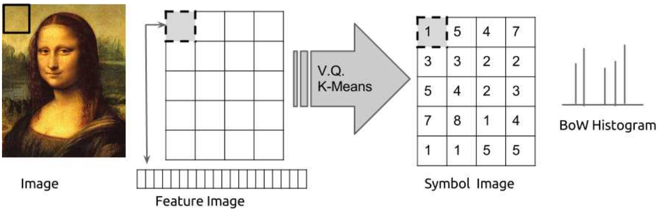
Part-2

Learning Hierarchical BoW using Naive Bayes Clustering

Image Representation

- Crucial component of solutions to popular computer vision tasks: classification, detection, clustering, retrieval.
- Two broad directions in the image representation community:
  - Bag of Words
  - Deep Learning

Bag of Words

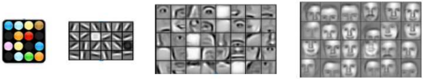


- Computing local features on interest points in the image.
- Vector Quantization.
- Image histogram computation.

Bag of Words

- Represent an image using the distribution of visual word occurrences.
- Ignore the spatial layout of visual words.
- Invariant to scale, translation and other deformations.
- Ignoring spatial information reduces discriminative power.

Deep Learning



- Learn artifacts hierarchically by assembling already learnt smaller artifacts.
- Exploit spatial information in Images.
- Summarize the features learnt in a neighbourhood by max/avearage pooling methods.
- Invariant to small translations and distortions.
- Training is expensive, requires many design decisions, huge training set. Most working methods are approximations of the actual objective.

Objective

- Raise the semantic depth of the low level BoW features.
- Use spatial context by bringing in neighbourhood information.
- Learn from both BoW, deep learning approaches.

Notation

- $\mathbf{X} = \{\mathbf{x}^n = (x_1^n \dots x_D^n)\}_{n=1}^N$: set of N data points.
- $X_d \in \mathbf{V}_d$. $\mathbf{V}_d = \left\{v_1^d \dots v_{M_d}^d\right\}$ is a *discrete* feature vocabulary.
- Each 2-D discrete image patch of size $P \times P$ is treated as a one-dimensional vector of size $D = P^2$.
- Each symbol comes from the same vocabulary.

Naive Bayes Clustering

- Cluster combinations of low level symbols.
- Image patch is a patch of (SIFT-BoW) visual words.

Equations

Mixture model: maximize a (log) likelihood objective.

$$J(\boldsymbol{\Theta}) = \log \prod_{n=1}^{N} P(\mathbf{x}^n) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} P(k) P\left(\mathbf{x}^n | k\right). \qquad (5)$$

Naive Bayes discrete density function:

$$P(\mathbf{x}^n | k) = \prod_{d=1}^{D} P\left(x_d^n | k\right) \qquad (6)$$

Equations

- Priors should add to one.
- Density functions over all values of a feature should add to one.
- E-step: Eq. 7. M-steps: Eq. 8, 9.

$$P_t(k|\mathbf{x}_n) = \frac{P_{t-1}(\mathbf{x}_n|k)P_{t-1}(k)}{\sum_{k'}^{K} P_{t-1}(\mathbf{x}_n|k')P_{t-1}(k')} \tag{7}$$

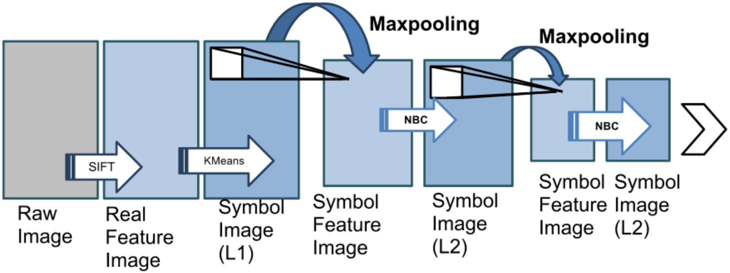
$$P_t(k) = \frac{\lambda + \sum_{n=1}^{N} P_t(k|\mathbf{x}_n)}{\lambda K + N} \tag{8}$$

$$P_t(v_m^d|k) = \frac{\lambda' + \sum_{n=1}^{N} \delta(x_{n,d} = v_m^d)P_t(k|\mathbf{x}_n)}{\lambda' M_d + NP_t(k)} \tag{9}$$

NBC vs K-Means

- Both EM approaches.
- K-Means clusters real-valued data. Visual words are symbols.
- NBC clusters symbolic vectors.

Learning Hierarchical BoW



Hierarchical Bag of Words

Caltech 101

Caltech 101:

Method	Accuracy
BoW* (Lazebnik et al.)	$64.6 \pm 0.8\%$
CDBN (Lee et al.)	$65.4 \pm 0.5\%$
BoW (our implementation)	$68.3 \pm 1.3\%$
NBC	$72.4 \pm 1.8\%$

VOC Pascal 2007

- Study the effect of parameters: patch size (p), size of the symbol space at each level (K) and level of hierarchy (L).

VOC Pascal 2007:

Method	SIFT BoW	L2	L2	L2	L3	L3	L3
p	-	3	3	2	2	2	2
K	1000	100	250	100	250	100	200
mAP	52.84	54.90	55.86	55.64	56.20	56.48	57.04

Contributions

- A novel Naive Bayes Clustering algorithm for clustering symbolic vectors.
- A hierarchical feature learning framework to create higher level *symbols* from combinations of lower level *symbols*.

Part-3

Learning Multiple Non-Linear Subspaces using K-RBMs

Understanding Data

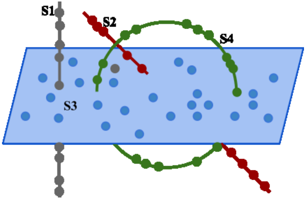


Figure: A set of points drawn from a union of four subspaces.

- Finding rich features that capture the complexity in data is challenging, yet necessary.
- In image domains, data might lie in multiple non-linear sub-spaces.

Visual BoW for Image Feature Learning

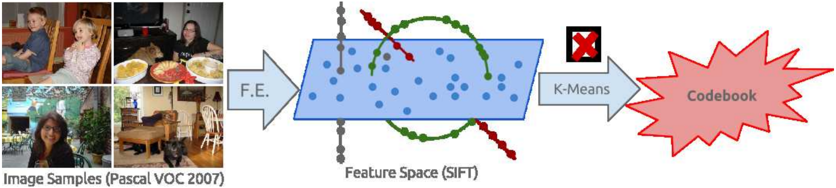


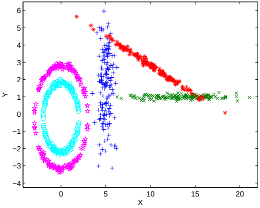
Figure: Part of the BoW representation pipeline. K-Means fails to capture the underlying structure in the feature space, for it assumes that *data points lie in the same space.*

Structure in Data

- We seek structure in data in two most pervasive forms: (a) *Clusters*, and (b) *Projections*.
- **Clustering hypothesis.** Data is not randomly distributed across the feature space but has inherent high density regions with few outliers and/or background noise points.
- **Projection hypothesis.** Features in the data are not completely independent of each other; they have some correlations among them. The real structure in the data could be in one or more linear or non-linear manifold(s) of the raw feature space.

Hypotheses

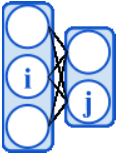
- *Clustering and projection are two “coupled” paradigms for understanding the nature of data*
- In general *the data is embedded in multiple non-linear subspaces and within each manifold there may be further clusters.*



Objectives

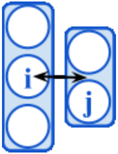
- A clustering framework that follows a "coupled" approach: *learns cluster associations and projections simultaneously.*
- Application of this framework to enhancing the visual bag of words pipeline by following a *two level (non-linear + linear) clustering strategy.*
- Application of the clustering framework to *feature learning from raw image patches.*

Restricted Boltzmann Machines



- Two layered, fully connected networks.
- Model a distribution over visible variables by introducing a set of stochastic features.

Restricted Boltzmann Machines, Formulation



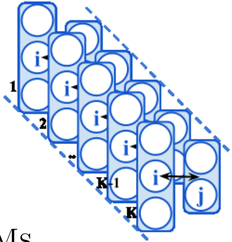
$$Pr(h_j|\mathbf{v}) = \sigma\left(\sum_{i=0}^{I} \mathbf{w}_{ij} v_i\right) \quad (10)$$

$$Pr(v_i|\mathbf{h}) = \sigma\left(\sum_{j=0}^{J} \mathbf{w}_{ij} h_j\right) \quad (11)$$

$$\Delta w_{ij} = \eta(< v_i^+ h_j^+ > - < v_i^- h_j^- >) \quad (12)$$

$$\epsilon = \sum_{i=1}^{I} (v_i^+ - v_i^-)^2 \quad (13)$$

K-RBMs



- K component RBMs
- Each component RBM learns a non-linear subspace.
- The associations between data samples and component RBMs are dictated by the reconstruction errors.

Clustering using K-RBMs

- **Hard** Clustering
  - Each input sample is fed to all component RBMs, and is assigned to the **one** which reconstructs it best.
  - Each RBM is then trained using the **samples assigned to it.**
- **Soft** Clustering
  - Each point is assigned **softly to all the component** RBMs.
  - Each RBM is trained using **all the points**, the contributions are **weighted** by the soft associations of points.

Clustering using K-RBMs

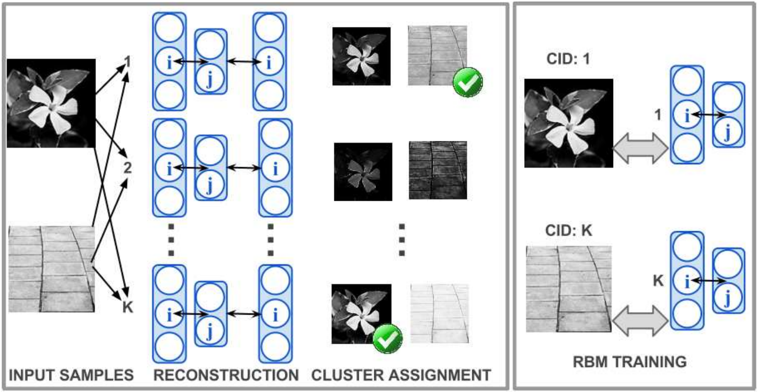


Figure: Schematic Diagram of our hard clustering approach.

RBMs vs K-RBMs

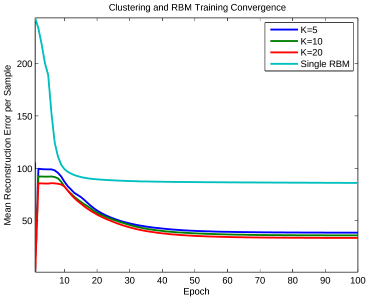
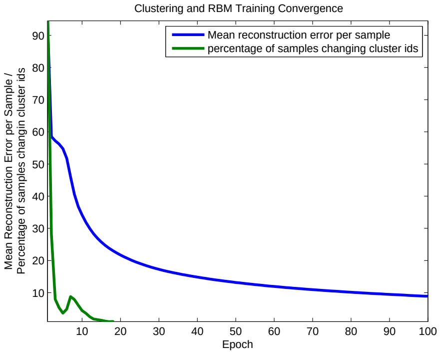


Figure: A plot of reconstruction errors vs epochs of training process for our experiments on the Pascal dataset. **For the Single RBM, we divide the mean error by 10 to bring it to scale with the others.**

Convergence

- Two kinds of convergence.
  - **Clustering** convergence.
  - **RBM** convergence.



Clustering Synthetic Datasets

Table: Running Time, Misclassification Errors and Mutual Information between cluster and class labels of various methods on two synthetic datasets.

Method	Dataset D1			Dataset D2		
	Runtime	Error	M.I.	Runtime	Error	M.I.
K-means	0.68s	27.4%	1.9219	2.76s	29.6%	1.9219
PCA + K-means	0.37s	27.4%	1.9219	0.42s	29.8%	1.9219
t-SNE + K-means	11.68s	11.3%	1.9619	11.93s	23.6%	1.9329
RBM + K-means	3.29s	26.6%	1.9219	3.89s	28.2%	1.9219
RANSAC	134.80s	66.6%	0.1529	474.72s	69.6%	0.1499
SSC	365.29s	0%	2.3219	690.93s	15.6%	2.0732
K-RBM	**0.46**s	**0%**	**2.3219**	**3.62**s	**0%**	**2.3219**

Clustering MNIST Dataset.

The MNIST data has $70,000$ data points of binary handwritten digits from 0 to 9.

Table: Comparision of coupled vs. de-coupled projection + clustering learning algorithms on MNIST data.

METHOD	PURITY	ERROR	M.I.
K-MEANS	59.43%	45.23%	1.6651
PCA + K-MEANS	59.36%	45.24%	1.6627
RBM + K-MEANS	60.20%	44.83%	1.6951
K-RBM-B	63.83%	42.79%	1.9127
K-RBM-R	**65.16%**	**38.90%**	**2.0878**

K-RBMs for Visual Bag of Words.

- **2 level** clustering
  - **Non-linear clustering** using K-RBMs
  - **Linear clustering** in each manifold using K-Means

Table: Classification performance of traditional and K-RBM BoW representations.

Dataset	Baseline BoW		K-RBM BoW	
	Performance	mean Q.E.	Performance	mean Q.E.
VOC PASCAL 2007	52.84%	0.7678	**56.40**% ($K_1 = 8$, $K_2 = 125$)	0.1620
15 Scene	80.50 ± 0.5%	0.5635	**85.75** ± **0.6**% ($K_1 = 20$, $K_2 = 50$)	0.0840
Caltech 101	68.34 ± 1.3%	0.6420	**72.80** ± **1.1**% ($K_1 = 8$, $K_2 = 125$)	0.1365

Feature Learning using K-RBMs

- **Learnt** vs **SIFT** features.
- **Dense, local** K-RBM features, computed over raw image patches.

Table: Classification Performance of K-RBM Features on Caltech 101 and VOC Pascal 2007 Datasets.

Table: Caltech 101

Method	Accuracy
SIFT Features	68.34 $\pm$ 1.3%
CDBN (layers 1+2)	65.4 $\pm$ 0.5%
K-RBM Features ($K_1 = 20$)	**74.2 $\pm$ 1.7%**

Table: VOC Pascal 2007

Method	Mean AP
SIFT Features	52.84%
K-RBM Features ($K_1 = 20$)	**58.40%**

Highlights

- Our *EM like framework* learns cluster associations and projections *simultaneously*.
- Our clustering method is significantly faster than other state of the art approaches.
- Use of K-RBMs as a non-linear clustering component along with K-Means for learning BoW representations improves image classification accuracy significantly.
- Dense local K-RBM features outperform SIFT based representations for image classification.

Conclusions & Future Work

- Our experiments support our hypotheses: (a) clustering and projection are coupled paradigms, and (b) in general, the data lies in multiple non-linear subspaces, and within each manifold, there may be further linear clusters.
- Compared to other state of the art approaches, K-RBMs are significantly faster, and hence more practical.
- K-RBMs can be extended to incorporate class-supervision where a separate K-RBM can be learnt for each class.

Final Words

- We learnt representations for popular CV tasks such as Action Recognition, Clustering, Visual Classification.
- PLS Kernel for Video Similarity.
- Naive Bayes Clustering, Hierarchical Bag of Words.
- K-RBMs for learning multiple subspaces in data.
- Despite all advances, Computer Vision is still a hard problem.

Related Publications

1. **Partial Least Squares Kernel for Computing Similarities between Video Sequences** *(Oral)*
   Siddhartha Chandra & C.V. Jawahar.
   International Conference on Pattern Recognition, Japan, November 2012

2. **Learning Hierarchical Bag of Words using Naive Bayes Clustering**
   Siddhartha Chandra, Shailesh Kumar & C.V. Jawahar.
   Asian Conference on Computer Vision, Korea, November 2012

3. **Learning Multiple Non-Linear Supspaces using K-RBMs**
   Siddhartha Chandra, Shailesh Kumar, C.V. Jawahar.
   IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013, USA

Thank You!