LEARNING TO COMPARE

GRAHAM TAYLOR
SCHOOL OF ENGINEERING
UNIVERSITY OF GUELPH

Deep Learning Summer School 2015
Montreal, Quebec
Overview: this talk
Overview: this talk

- Learning to compare examples
  - it’s a big field!
  - we will focus on methods inspired by deep learning and representation learning
Overview: this talk

• Learning to compare examples
  - it’s a big field!
  - we will focus on methods inspired by deep learning and representation learning

• Applications: finding similar documents, pose-sensitive retrieval, zero-shot learning
Learning similarity

- Pixel distance ≠ perceptual similarity
- Computing distances in pixel space is also computationally expensive
- Learning parametric embeddings that are *invariant* to certain input variability
The setup

- Perceptually similar observations are mapped to nearby points on a manifold
- Key question: where does similarity come from?
The setup

- Perceptually similar observations are mapped to nearby points on a manifold
- Key question: where does similarity come from?
The setup

- Perceptually similar observations are mapped to nearby points on a manifold
- Key question: where does similarity come from?
The setup

- Perceptually similar observations are mapped to nearby points on a manifold
- Key question: where does similarity come from?
One motivation: nearest neighbour methods

- Surprisingly effective (Boiman et al. 2008, McCann and Lowe, 2012)
- Fast, especially when combined with Approximate Nearest Neighbour or Hashing
- Generalize to new classes at near-zero cost (Mensink et al. 2013)

Image: Boiman et al. (2008)
Outline
Outline

Unsupervised
LSA, Semantic Hashing, Multi-index Hashing
Unsupervised
LSA, Semantic Hashing, Multi-index Hashing

Supervised
NCA, Nonlinear NCA, DrLIM, Triplet Embedding

\[ d_{ij} = ||z_i - z_j||_2 \]
Outline

Unsupervised
LSA, Semantic Hashing, Multi-index Hashing

Supervised
NCA, Nonlinear NCA, DrLIM, Triplet Embedding

Weakly supervised
Applications to pose-sensitive retrieval, zero-shot learning
Unsupervised approach
Unsupervised approach

- Learn (possibly deep) representations completely unsupervised
  - compute distances between top-level representations
  - representations are usually low-dimensional
Unsupervised approach

• Learn (possibly deep) representations \textit{completely unsupervised}
  - compute distances between top-level representations
  - representations are usually low-dimensional

• Classical methods: Latent Semantic Analysis (based on SVD), pLSA, LDA
  - But directed models don’t seem like a natural fit
  - fast inference is important for information retrieval
Unsupervised approach

- Learn (possibly deep) representations completely unsupervised
  - compute distances between top-level representations
  - representations are usually low-dimensional
- Classical methods: Latent Semantic Analysis (based on SVD), pLSA, LDA
  - But directed models don’t seem like a natural fit
  - fast inference is important for information retrieval
- Use undirected models in which exact inference is fast
  - Single layer approach by generalizing RBMs: Welling et al. 2005
  - Multi-layer approach: Salakhutdinov and Hinton 2007 “Semantic Hashing”
Constrained Poisson model

- Visible layer represents word-count vector of a document
  - special RBM: “Constrained Poisson Model”
- Learn Constrained Poisson ➔ Binary first layer
- This allows you to represent each document with a binary representation
- Forms the first layer of a deep model
Deep auto-encoders
Deep auto-encoders

input $x$
Deep auto-encoders

• Learn one or more binary RBMs in a “greedy” fashion
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
- Unroll to a deep autoencoder and “fine-tune” w/ backprop
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
- Unroll to a deep autoencoder and “fine-tune” w/ backprop
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
- Unroll to a deep autoencoder and “fine-tune” w/ backprop
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
- Unroll to a deep autoencoder and “fine-tune” w/ backprop
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
- Unroll to a deep autoencoder and “fine-tune” w/ backprop
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
- Unroll to a deep autoencoder and “fine-tune” w/ backprop
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
- Unroll to a deep autoencoder and “fine-tune” w/ backprop
Deep auto-encoders

- Learn one or more binary RBMs in a “greedy” fashion
- Unroll to a deep autoencoder and “fine-tune” w/ backprop
  - During fine-tuning add Gaussian noise to code layer
  - This forces the codes to be close to binary
Extremely fast retrieval

- Documents are mapped to 20-D binary codes
- Can retrieve similar documents stored at nearby addresses with no search
- Binary LSA significantly reduces performance
  - Not surprising: it has not been optimized to make binary codes perform well
- One weakness: documents with similar addresses have similar content but the converse is not necessarily true
  - Can we use external information (e.g. labels) to pull together codes of similar documents?

Figures from R. Salakhutdinov and G. Hinton
Hashing longer codes

- If code lengths are > 32 bits, use codes as direct indices (addresses) into a hash table
  - dramatic increase in search speed compared to exhaustive linear scan

- Code lengths are often much longer in order to achieve good performance
  - but number of hash buckets to examine grows near-exponentially with search radius

![Graph](https://example.com/graph.png)

---

06 Aug 2015 / 12

DLSS · Learning to Compare/ G Taylor

Figures: Norouzi et al. (2014)
Multi-index hashing

(Norouzi et al. 2012, 2014)

- When hash codes are > 32 bits, use Multi-index hashing
- Provably sub-linear search complexity for uniformly distributed codes
- Binary codes are indexed $m$ times into $m$ different hash tables, based on $m$ disjoint substrings
- Given a query code, entries that fall close to the query in at least one such substring are considered neighbour candidates
- Candidates then checked for validity using entire binary code
- Guaranteed that all true neighbours will be found

https://github.com/norouzi/mih
Learning embeddings with a Siamese network

\[ d(\cdot, \cdot) = \text{SMALL} \]
Learning embeddings with a Siamese network

\[ d(\cdot, \cdot) = \text{SMALL} \]

Identical pathways
Learning embeddings with a Siamese network

\[ d(\cdot, \cdot) = \text{SMALL} \]

Identical pathways

\[ f(\cdot | \theta) \]

\[ f(\cdot | \theta) \]

\[ d(\cdot, \cdot) = \text{BIG} \]

\[ f(\cdot | \theta) \]

\[ f(\cdot | \theta) \]
Not a new idea!

(Bromley, Guyon, LeCun, Sackinger, and Shah 1994)

- Architecture proposed for signature verification
  - didn’t really get the distance function right
  - learning unstable
  - small (by today’s standards) training set
- 1D convolution (TDNN)
- Developed independently elsewhere:
  - Baldi and Chauvin, 1992: fingerprint verification
  - Becker and Hinton, 1992 - discovering depth in random-dot stereograms
Convnets: single stage

Convnets: single stage

Convnets: single stage

Convnets: single stage

Convnets: single stage

Convnets: single stage

Convnets: single stage
Convnets: typical architecture

Single stage

Whole system
Embedding with a Siamese convnet

What’s the objective function?
- needs to pull together semantically similar pairs
- needs to push apart semantically dissimilar pairs
Training Siamese nets

(Bromley, Guyon, LeCun, Sackinger, and Shah 1994)

- Siamese nets can be trained by error backpropagation, just need to define an objective function:
  - Neighbourhood Component Analysis (Goldberger et al. 2004)
  - Dimensionality Reduction by Learning an Invariant Mapping (Hadsell et al. 2006)
  - Triplet-based Criterion (Chechik et al. 2010)
  - Quadruplet-based Criterion (Law et al. 2013)
Neighbourhood components analysis (NCA)

(21)

Neighbourhood components analysis (NCA)

(21)

Credit: Sam Roweis

06 Aug 2015 / 21

DLSS · Learning to Compare / G Taylor
Neighbourhood components analysis (NCA)

- Learn a metric which minimizes KNN classification error

(Q: What is the right distance metric for KNN classification?
A: The one that optimizes test error!

Let's try to approximate this by the one which optimizes training error, defined using leave-one-out cross validation.

So if I gave you a finite set of distance metrics to choose between (and I told you K), you could pick the best one.

Obvious next question: if I gave you a continuously parameterized family of metrics to search through, could you find the one which maximizes LOO classification performance?

And what about K...? (Goldberger et al. 2004)
Neighbourhood components analysis (NCA)

- Learn a metric which minimizes KNN classification error

- Two problems:

  Q: What is the right distance metric for KNN classification?

  A: The one that optimizes test error!

  Let's try to approximate this by the one which optimizes training error, defined using leave-one-out cross validation.

  So if I gave you a finite set of distance metrics to choose between (and I told you K), you could pick the best one.

  Obvious next question: if I gave you a continuously parameterized family of metrics to search through, could you find the one which maximizes LOO classification performance?

  And what about $K$...?

(Goldberger et al. 2004)
Neighbourhood components analysis (NCA)

• Learn a metric which minimizes KNN classification error

• Two problems:
  - Error is a highly discontinuous function of the distance metric

(Goldberger et al. 2004)
Neighbourhood components analysis (NCA)

- Learn a metric which minimizes KNN classification error
- Two problems:
  - Error is a highly discontinuous function of the distance metric
  - We still need to choose K

(Goldberger et al. 2004)
Neighbourhood components analysis (NCA)

• Learn a metric which minimizes KNN classification error

• Two problems:
  - Error is a highly discontinuous function of the distance metric
  - We still need to choose K

• Look for a smoother (or at least continuous) cost function

(Goldberger et al. 2004)
Stochastic nearest neighbour
Stochastic nearest neighbour

• Instead picking from a fixed set of $K$ nearest neighbours, select a single neighbour stochastically.
Stochastic nearest neighbour

- Instead picking from a fixed set of $K$ nearest neighbours, select a single neighbour stochastically.

- Let each point $i$ select other points $j$ as its neighbour with probability $p_{ij}$ based on the softmax of the distance $d_{ij}$:

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)}$$

where:

$$d_{ij} = \|z_i - z_j\|_2$$

$$z_i = f(x_i | \theta)$$

Figure: Sam Roweis
Maximize the expected number of points correctly classified under this scheme.

This is much smoother than the actual leave-one-out cross-validation error!

In fact, it is differentiable w.r.t. parameters of mapping.

- Can use SGD or other gradient-based optimizer.

And there is no explicit parameter $K$.

- See (Tarlow et al. 2013) for $K > 1$ objective.

$$L_{NCA} = - \sum_{i=1}^{N} \sum_{j:y_{i}=y_{j}} p_{ij}$$

Minimize loss w.r.t. $\theta$.
Linear NCA: embeddings

\[ f(x|\theta = A) = Ax \]
NCA: MNIST

MNIST
(D=784)
Nonlinear NCA

• The original NCA paper (Goldberger et al. 2004) points out that $f(x_i | \theta)$ need not be a linear mapping

• Salakhutdinov and Hinton (2007) pre-train with an RBM, then fine-tune with the NCA objective

• Can combine the NCA objective with an Autoencoder objective to regularize:

$$C = \lambda L_{NCA} + (1 - \lambda) L_{AE}$$

• Can take advantage of unlabeled data!
Learning nonlinear NCA

Figure: Salakhutdinov and Hinton
Limitations of NCA

- Despite very nice embeddings (see right) NCA has a quadratic normalization term (must consider all pairs)
  - mini-batch training (approximate)
  - objectives that don’t require normalization

- What about continuous labels?
  - (Goldberger et al. 2004) describe a “soft” form of NCA that can use continuous labels

(Figures from R. Salakhutdinov and G. Hinton)
Class-conditional metric learning

(Im and Taylor - In submission)

Daniel Im (here at DLSS!)
Class-conditional metric learning

(Im and Taylor - In submission)

- Optimize Image-to-Class distance (Boiman et al. 2008)

Daniel Im (here at DLSS!)
Class-conditional metric learning

(Im and Taylor - In submission)

- Optimize Image-to-Class distance (Boiman et al. 2008)

- Stochastic neighbour selection rule:

\[
p_{i}^{C} = \frac{\exp \left( -\frac{1}{k} \sum_{j=1}^{k} || z_{i} - \text{NN}_{j}^{C} (z_{i}) ||^2 \right)}{\sum_{C'} \exp \left( -\frac{1}{k} \sum_{j=1}^{k} || z_{i} - \text{NN}_{j}^{C'} (z_{i}) ||^2 \right)},
\]

Daniel Im (here at DLSS!)
Class-conditional metric learning

- Optimize Image-to-Class distance (Boiman et al. 2008)

- Stochastic neighbour selection rule:

\[
p_i^C = \frac{\exp \left( -\frac{1}{k} \sum_{j=1}^{k} \| z_i - NN_j^C(z_i) \|^2 \right)}{\sum_{C'} \exp \left( -\frac{1}{k} \sum_{j=1}^{k} \| z_i - NN_j^{C'}(z_i) \|^2 \right)},
\]

Daniel Im (here at DLSS!)
DrLIM (Dimensionality reduction by learning an invariant mapping)

\[ L = s_{ij}L_S(x_i, x_j) + (1 - s_{ij})L_D(x_i, x_j) \]

\[ L_S(x_i, x_j) = \frac{1}{2}(d_{ij})^2 \]

\[ L_D(x_i, x_j) = \frac{1}{2}[\max(0, \alpha - d_{ij})]^2 \]

- The similarity loss “pushes together” similar points
- The dissimilarity loss “pulls apart” dissimilar points
  - but only if their distance is within some margin, \( \alpha \)

Hadsell, Chopra and LeCun 2006
Spring analogy

- Solid dots are points that are similar to the point in the centre
- Hollow dots are points that are dissimilar to the point in the centre
- Forces acting on the points are shown in blue
  - The length of the arrow represents the strength of the force
- Radius represents the margin, $\alpha$

Figures from Hadsell et al.
Triplet-based embedding

(Chechik et al. 2010)

Given a similarity score \( S(x_i, x_j) \) for inputs \( x_i, x_j \)

We want to learn an embedding \( f(x) \) such that

\[
D(f(x_i), f(x_i^+)) < D(f(x_i), f(x_i^-))
\]

\( \forall x_i, x_i^+, x_i^- \) such that \( S(x_i, x_i^+) > S(x_i, x_i^-) \)

```
“triplet”
```

\( D(f(x_i), f(x_j)) \) is a distance measure, commonly

\[
D(f(x_i), f(x_j)) = \| f(x_i) - f(x_j) \|^2
\]
Learning fine-grained image similarity with deep ranking

(Wang et al. 2014)

Objective:

$$\min \sum \xi_i + \lambda || \theta ||^2$$

s.t.: $\max (0, g + D(f(x_i), f(x_i^+)) - D(f(x_i), f(x_i^-))) \leq \xi_i$

$\forall x_i, x_i^+, x_i^- \text{ s.t. } S(x_i, x_i^+) > S(x_i, x_i^-)$

\[\xi_i\] penalty
\[g\] gap (hyperparameter)
\[\theta\] weights in network
\[\lambda\] regularization strength (hyperparameter)
How to: triplet sampling
How to: triplet sampling

- # of possible triplets increases cubically with # of images
- e.g. 12M images, 1.728 x 10^21 triplets!
- Optimization converges in ~24M triplet samples
- Uniformly sampling triplets is sub-optimal
How to: triplet sampling

- # of possible triplets increases cubically with # of images
- e.g. 12M images, 1.728 x 10^21 triplets!
- Optimization converges in ~24M triplet samples
- Uniformly sampling triplets is sub-optimal
- Propose an online triplet sampling algorithm (more details in paper):
  - Sample an image according to its “relevance” to a category
  - Sample a positive image with high relevance
  - Sample “out-of-class” negatives uniformly
  - Sample “in-class” relevant negatives but ensure a margin between positive and negative examples
Finding similarity data
Finding similarity data

- NCA, DrLIM: binary notion of similarity typically defined by class membership or explicitly constructed neighbourhood graph
Finding similarity data

- NCA, DrLIM: binary notion of similarity typically defined by class membership or explicitly constructed neighbourhood graph

- Defining pairwise similarity is difficult and inconsistent across observers; Google used “Golden Feature” - weighted linear combination of 27 features
Finding similarity data

- NCA, DrLIM: binary notion of similarity typically defined by class membership or explicitly constructed neighbourhood graph

- Defining pairwise similarity is difficult and inconsistent across observers; Google used “Golden Feature” - weighted linear combination of 27 features

- Despite crowd-sourcing platforms (e.g. Amazon Mechanical Turk) gathering semantically similar pairs of images is expensive
Hands by hand

- One solution is to turn to synthetic data (e.g. Shakhnarovich et al. 2003, Jain et al. 2008)
- Difficult to generalize to real (e.g. “YouTube” settings)
- Another solution: ask people to label heads and hands (Spiro et al. 2010) or superimpose articulated skeletons (Bourdev et al. 2009)

(Spiro et al. 2010)
One solution is to turn to *synthetic* data (e.g. Shakhnarovich et al. 2003, Jain et al. 2008)

Difficult to generalize to real (e.g. “YouTube” settings)

Another solution: *ask people* to label heads and hands (Spiro et al. 2010) or superimpose articulated skeletons (Bourdev et al. 2009)
Pose-sensitive embeddings

(Taylor et al. 2010)
Pose-sensitive embeddings

(Taylor et al. 2010)
Pose-sensitive embeddings

(Database) (Taylor et al. 2010)
Pose-sensitive embeddings

(Taylor et al. 2010)

• If we have a database of images labeled with 2D or 3D pose information - we can do non-parametric pose estimation
Pose-sensitive embeddings
(Taylor et al. 2010)

- If we have a database of images labeled with 2D or 3D pose information - we can do non-parametric pose estimation

Pose-sensitive embeddings (Taylor et al. 2010)
Pose-sensitive embeddings
(Taylor et al. 2010)

- If we have a database of images labeled with 2D or 3D pose information - we can do non-parametric pose estimation
- Nearest neighbor lookup must be quick (e.g. performed in a low-dimensional space)
Pose-sensitive embeddings

(Taylor et al. 2010)

- If we have a database of images labeled with 2D or 3D pose information - we can do non-parametric pose estimation.

- Nearest neighbor lookup must be quick (e.g. performed in a low-dimensional space).

- It also must be informative of pose and invariant to clothing, lighting, scale, and other appearance changes.
NCA regression

\[ L_{\text{NCA}} = \sum_{i=1}^{N} \sum_{j} p_{ij} ||\mathbf{y}_i - \mathbf{y}_j||^2_2 \]

Minimize loss w.r.t.

Pay a high cost for “neighbours” in feature space that are far away in pose space

\[ \mathbf{y}_i = [48.2, 46.3, \ldots, 63.3]^T \]

\[ \mathbf{x}_i \]

\[ \mathbf{y}_i = [54.4, 45.8, \ldots, 64.1]^T \]

\[ \mathbf{x}_j \]
Snowbird dataset

- We digitally recorded all contributing and invited speakers at the 2010 Snowbird workshop.

- After each session of talks, blocks of 150 frames were distributed as Human Intelligence Tasks (HITs) on Amazon Mechanical Turk.
## Comparison of Approaches

<table>
<thead>
<tr>
<th>Method</th>
<th>Not practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel distance</td>
<td><strong>GIST</strong></td>
</tr>
<tr>
<td></td>
<td>• Global representation of image</td>
</tr>
<tr>
<td></td>
<td>• Still not practical</td>
</tr>
<tr>
<td>Linear NCA regression (NCAR)</td>
<td>• Applied to pre-computed GIST</td>
</tr>
<tr>
<td></td>
<td>• Fit by conjugate gradient</td>
</tr>
<tr>
<td>Convolutional NCAR (C-NCAR)</td>
<td>• Convolutions applied to pixels</td>
</tr>
<tr>
<td></td>
<td>• Tanh(), Abs(), Average downsampling</td>
</tr>
<tr>
<td>DrLIM Regression (DrLIMR)</td>
<td>• Similar to NCAR but adds an explicit contrastive loss</td>
</tr>
<tr>
<td>Convolutional DrLIMR (C-DrLIMR)</td>
<td>• Similar to C-NCAR but adds an explicit contrastive loss</td>
</tr>
</tbody>
</table>
## Comparison of Approaches

<table>
<thead>
<tr>
<th>Method</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel distance</td>
<td></td>
</tr>
<tr>
<td>GIST</td>
<td>Global representation of image • Still not practical</td>
</tr>
<tr>
<td>Linear NCA regression (NCAR)</td>
<td>• Applied to pre-computed GIST • Fit by conjugate gradient</td>
</tr>
<tr>
<td>Convolutional NCAR (C-NCAR)</td>
<td>• Convolutions applied to pixels • Tanh(), Abs(), Average downsampling</td>
</tr>
<tr>
<td>DrLIM Regression (DrLIMR)</td>
<td>• Similar to NCAR but adds an explicit contrastive loss</td>
</tr>
<tr>
<td>Convolutional DrLIMR (C-DrLIMR)</td>
<td>• Similar to C-NCAR but adds an explicit contrastive loss</td>
</tr>
</tbody>
</table>
## Comparison of Approaches

<table>
<thead>
<tr>
<th>Method</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel distance</td>
<td>Not practical</td>
</tr>
<tr>
<td>GIST</td>
<td>• Global representation of image</td>
</tr>
<tr>
<td></td>
<td>• Still not practical</td>
</tr>
<tr>
<td>Linear NCA regression (NCAR)</td>
<td>• Applied to pre-computed GIST</td>
</tr>
<tr>
<td></td>
<td>• Fit by conjugate gradient</td>
</tr>
<tr>
<td>Convolutional NCAR (C-NCAR)</td>
<td>• Convolutions applied to pixels</td>
</tr>
<tr>
<td></td>
<td>• Tanh(), Abs(), Average downsampling</td>
</tr>
<tr>
<td>DrLIM Regression (DrLIMR)</td>
<td>• Similar to NCAR but adds an explicit contrastive loss</td>
</tr>
<tr>
<td>Convolutional DrLIMR (C-DrLIMR)</td>
<td>• Similar to C-NCAR but adds an explicit contrastive loss</td>
</tr>
</tbody>
</table>
Results
(qualitative)

- Both Pixel-based matching and GIST focus on scene content, lighting
- Our method learns invariance to background, focuses on pose
- Though trained on hands relative to head, seems to capture something more substantial about body pose
# Results (quantitative)

<table>
<thead>
<tr>
<th>Embedding</th>
<th>Input</th>
<th>Code size</th>
<th>Err-SY</th>
<th>Err-RE</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Pixels</td>
<td>16384</td>
<td>32.86</td>
<td>25.12</td>
</tr>
<tr>
<td>None</td>
<td>GIST</td>
<td>512</td>
<td>47.41</td>
<td>25.3</td>
</tr>
<tr>
<td>PCA</td>
<td>GIST</td>
<td>128</td>
<td>47.17</td>
<td>24.85</td>
</tr>
<tr>
<td>PCA</td>
<td>GIST</td>
<td>32</td>
<td>48.99</td>
<td>25.74</td>
</tr>
<tr>
<td>NCAR</td>
<td>GIST</td>
<td>32</td>
<td>34.21</td>
<td>24.93</td>
</tr>
<tr>
<td>NCAR</td>
<td>LCN+GIST</td>
<td>32</td>
<td>32.9</td>
<td>23.15</td>
</tr>
<tr>
<td>S-DrLIM</td>
<td>GIST</td>
<td>32</td>
<td>37.8</td>
<td>25.19</td>
</tr>
<tr>
<td>Boost-SSC</td>
<td>LCN+GIST</td>
<td>32</td>
<td>34.8</td>
<td>22.65</td>
</tr>
<tr>
<td>C-NCAR</td>
<td>LCN</td>
<td>32</td>
<td>28.95</td>
<td>16.41</td>
</tr>
<tr>
<td>C-DRLIM</td>
<td>LCN</td>
<td>32</td>
<td>25.4</td>
<td>19.61</td>
</tr>
</tbody>
</table>
MPII Human Pose

(Andriluka et al. 2014)

- Addresses appearance variability and complexity
- YouTube as a data source
- Many activities, indoor and outdoor scenes, variety of imaging conditions

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#training</th>
<th>#test</th>
<th>img. type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full body pose datasets</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parse [16]</td>
<td>100</td>
<td>205</td>
<td>diverse</td>
</tr>
<tr>
<td>LSP [12]</td>
<td>1,000</td>
<td>1,000</td>
<td>sports (8 types)</td>
</tr>
<tr>
<td>PASCAL Person Layout [6]</td>
<td>850</td>
<td>849</td>
<td>everyday</td>
</tr>
<tr>
<td>Sport [21]</td>
<td>649</td>
<td>650</td>
<td>sports</td>
</tr>
<tr>
<td>UIUC people [21]</td>
<td>346</td>
<td>247</td>
<td>sports (2 types)</td>
</tr>
<tr>
<td>LSP extended [13]</td>
<td>10,000</td>
<td>-</td>
<td>sports (3 types)</td>
</tr>
<tr>
<td>FashionPose [2]</td>
<td>6,530</td>
<td>775</td>
<td>fashion blogs</td>
</tr>
<tr>
<td>Upper body pose datasets</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffy Stickmen [8]</td>
<td>472</td>
<td>276</td>
<td>TV show (Buffy)</td>
</tr>
<tr>
<td>ETHZ PASCAL Stickmen [3]</td>
<td>-</td>
<td>549</td>
<td>PASCAL VOC</td>
</tr>
<tr>
<td>Human Obj. Int. (HOI) [23]</td>
<td>180</td>
<td>120</td>
<td>sports (6 types)</td>
</tr>
<tr>
<td>We Are Family [5]</td>
<td>350 imgs.</td>
<td>175 imgs.</td>
<td>group photos</td>
</tr>
<tr>
<td>Video Pose 2 [18]</td>
<td>766</td>
<td>519</td>
<td>TV show (Friends)</td>
</tr>
<tr>
<td>FLIC [17]</td>
<td>6,543</td>
<td>1,016</td>
<td>feature movies</td>
</tr>
<tr>
<td>Armlets [9]</td>
<td>9,593</td>
<td>2,996</td>
<td>PASCAL VOC/Flickr</td>
</tr>
<tr>
<td>MPII Human Pose (this paper)</td>
<td>28,821</td>
<td>11,701</td>
<td>diverse (491 act.)</td>
</tr>
</tbody>
</table>
Pose embeddings

- Similar to (Taylor et al. 2010), but uses:
  - MPII database: 2D locations of 16 body joints
  - Triplet-style learning
  - Modern, “Inception”-style convnet

(Mori et al. 2015)
Can we avoid explicit labeling of body parts?
Weakly-supervised embeddings

(Taylor et al. 2011)
Weakly-supervised embeddings

(Taylor et al. 2011)

• Have people imitate frames from a video:
  - imitated frames, though different in appearance, should be embedded nearby

\[
\text{seed} \{ \text{seed} \text{ seed} \ldots } \text{seed} \]
Weakly-supervised embeddings

(Taylor et al. 2011)

- Have people imitate frames from a video:
  - imitated frames, though different in appearance, should be embedded nearby

(seed)

imitations

X
Weakly-supervised embeddings

(Taylor et al. 2011)

- Have people imitate frames from a video:
  - imitated frames, though different in appearance, should be embedded nearby

- Use *temporal coherence* as a similarity signal:
  - i.e. frames which are close together in time should be embedded nearby

\[ \hat{Z} = f(X|\theta) \]
Zero-shot learning *(Nourouzi et al. 2014)*

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image 1" /></td>
<td>wig, fur coat, Saluki, gazelle hound, Afghan hound, Afghan stole</td>
<td>water spaniel, tea gown, bridal gown, wedding gown, spobel, tights, leotards</td>
<td>business suit, dress, frock, hairpiece, false hair, postiche, swimsuit, swimwear, bathing suit, outfit</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image 2" /></td>
<td>ostrich, Struthio camelus, black stork, Ciconia nigra, vulture, crane, peacock</td>
<td>heron, owl, bird of Minerva, bird of night, hawk, bird of prey, raptor, raptorial bird, finch</td>
<td>ratite, ratite bird, flightless bird, peafowl, bird of Juno, common spoonbill, New World vulture, cathartid, Greek partridge, cock partridge</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image 3" /></td>
<td>sea lion, plane, carpenter’s plane, cowboy boot, loggerhead, loggerhead turtle, goose</td>
<td>elephant, turtle, turtleneck, turtle, poloneck, flip-flop, thong, handcart, pushcart, cart, go-cart</td>
<td>California sea lion, Steller sea lion, Australian sea lion, South American sea lion, eared seal</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image 4" /></td>
<td>hamster, broccoli, Pomeranian, capuchin, ringtail, weasel</td>
<td>golden hamster, Syrian hamster, rhesus, rhesus monkey, pipe, shaker, American mink, Mustela vison</td>
<td>golden hamster, Syrian hamster, rodent, gnawer, Eurasian hamster, rhesus, rhesus monkey, rabbit, coney, cony</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image 5" /></td>
<td>thresher, threshing machine, tractor, harvester, reaper, half track, snowplow, snowplough</td>
<td>truck, motortruck, skidder, tank car, tank, automatic rifle, machine rifle, trailer, house trailer</td>
<td>flatcar, flatbed, flat truck, motortruck, tracked vehicle, bulldozer, dozer, wheeled vehicle</td>
</tr>
<tr>
<td><img src="image6.png" alt="Image 6" /></td>
<td>Tibetan mastiff, titi, titi monkey, koala, koala bear, kangaroo bear, llama, chow, chow chow</td>
<td>kernel, littoral, littoral, littoral zone, sands, carillon</td>
<td>dog, domestic dog, domestic cat, house cat, schnauzer, Belgian sheepdog, domestic llama, Lama peruana</td>
</tr>
</tbody>
</table>

Fig. 1 depicts some qualitative results. The first column shows the top 5 predictions of the convolutional net, referred to as the Softmax baseline [7]. The second and third columns show the zero-shot predictions by the DeViSE and ConSE(10) models. The ConSE(10) model uses the top T = 10 predictions of the Softmax baseline to generate convex combination of embeddings. Fig. 1 shows that the labels predicted by the ConSE(10) model are generally coherent and they include very few outliers. In contrast, the top 5 labels predicted by the DeViSE model include more outliers such as “flip-flop” predicted for a “Steller sea lion”, “pipe” and “shaker” predicted for a “hamster”, and “automatic rifle” predicted for a “farm machine”.

(Nourouzi et al. 2014)
Zero-shot learning (Nourouzi et al. 2014)

Can you exploit a trained word embedding model (Mikolov et al. 2013) and a trained object recognition model (Krizhevsky et al. 2012) to label images from unseen classes?

Figure 1: Zero-shot test images from ImageNet, and their corresponding top 5 labels predicted by the Softmax Baseline [7], DeViSE [6], and ConSE(10). The labels predicted by the Softmax baseline are the labels used for training, and the labels predicted by the other two models are not seen during training of the image classifiers. The correct labels are shown in blue. Examples are hand-picked to illustrate the cases that the ConSE(10) performs well, and a few failure cases.
Can you exploit a trained word embedding model (Mikolov et al. 2013) and a trained object recognition model (Krizhevsky et al. 2012) to label images from unseen classes?

Let softmax output of recognition model for top $T$ classes determine convex combination of semantic word embeddings.

Zero-shot learning (Nourouzi et al. 2014)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>wig</td>
<td>water spaniel</td>
<td>business suit</td>
<td></td>
</tr>
<tr>
<td>fur coat</td>
<td>tea gown</td>
<td>dress, frock</td>
<td></td>
</tr>
<tr>
<td>Saluki, gazelle hound</td>
<td>bridal gown, wedding gown</td>
<td>hairpiece, false hair, postiche</td>
<td></td>
</tr>
<tr>
<td>Afghan hound, Afghan stole</td>
<td>tights, leotards</td>
<td>swimsuit, swimwear, bathing suit kit, outfit</td>
<td></td>
</tr>
<tr>
<td>ostrich, Struthio camelus</td>
<td>heron</td>
<td>ratite, ratite bird, flightless bird</td>
<td></td>
</tr>
<tr>
<td>black stork, Ciconia nigra</td>
<td>owl</td>
<td>peafowl, bird of Juno</td>
<td></td>
</tr>
<tr>
<td>vulture</td>
<td>bird of Minerva, bird of night</td>
<td>common spoonbill</td>
<td></td>
</tr>
<tr>
<td>crane</td>
<td>hawk</td>
<td>New World vulture, cathartid</td>
<td></td>
</tr>
<tr>
<td>peacock</td>
<td>bird of prey, raptor, raptorial bird</td>
<td>Greek partridge, rock partridge</td>
<td></td>
</tr>
<tr>
<td>sea lion</td>
<td>elephant</td>
<td>California sea lion</td>
<td></td>
</tr>
<tr>
<td>plane, carpenter’s plane</td>
<td>turtle</td>
<td>Steller sea lion</td>
<td></td>
</tr>
<tr>
<td>cowboy boot</td>
<td>turtleneck, turtle, polo-neck</td>
<td>Australian sea lion</td>
<td></td>
</tr>
<tr>
<td>loggerhead, loggerhead turtle</td>
<td>flip-flop, thong</td>
<td>South American sea lion</td>
<td></td>
</tr>
<tr>
<td>goose</td>
<td>handcart, pushcart, cart, go-cart</td>
<td>eared seal</td>
<td></td>
</tr>
<tr>
<td>hamster</td>
<td>golden hamster, Syrian hamster</td>
<td>golden hamster, Syrian hamster</td>
<td></td>
</tr>
<tr>
<td>broccoli</td>
<td>rhesus, rhesus monkey</td>
<td>rodent, gnawer</td>
<td></td>
</tr>
<tr>
<td>Pomeranian</td>
<td>pipe</td>
<td>Eurasian hamster</td>
<td></td>
</tr>
<tr>
<td>capuchin, ringtail</td>
<td>shaker</td>
<td>rhesus, rhesus monkey</td>
<td></td>
</tr>
<tr>
<td>weasel</td>
<td>American mink, Mustela vison</td>
<td>rabbit, coney, cony</td>
<td></td>
</tr>
<tr>
<td>threshing machine</td>
<td>truck, motortruck</td>
<td>flatcar, flatbed, flat</td>
<td></td>
</tr>
<tr>
<td>tractor</td>
<td>skidder</td>
<td>truck, motortruck</td>
<td></td>
</tr>
<tr>
<td>harvester, reaper</td>
<td>tank car, tank</td>
<td>tracked vehicle</td>
<td></td>
</tr>
<tr>
<td>half track</td>
<td>automatic rifle, machine rifle</td>
<td>bulldozer, dozer</td>
<td></td>
</tr>
<tr>
<td>snowplow</td>
<td>trailer, house trailer</td>
<td>wheeled vehicle</td>
<td></td>
</tr>
<tr>
<td>(farm machine)</td>
<td>(alpaca, Lama pacos)</td>
<td>(alpaca, Lama pacos)</td>
<td></td>
</tr>
<tr>
<td>Tibetan mastiff</td>
<td>kernel</td>
<td>dog, domestic dog</td>
<td></td>
</tr>
<tr>
<td>titi, titi monkey</td>
<td>litoral, litoral, litoral zone, sands</td>
<td>domestic cat, house cat</td>
<td></td>
</tr>
<tr>
<td>koala, koala bear, kangaroo bear</td>
<td>carill</td>
<td>schnauzer</td>
<td></td>
</tr>
<tr>
<td>llama</td>
<td>chow, chow chow</td>
<td>Belgian sheepdog</td>
<td></td>
</tr>
<tr>
<td>chow</td>
<td>Cabernet, Cabernet Sauvignon</td>
<td>domestic llama, Lama peruana</td>
<td></td>
</tr>
<tr>
<td>(farm machine)</td>
<td>poodle, poodle dog</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Summary

Unsupervised
Learn similarity structure completely from unlabeled data.
Difficult to ensure that similar examples map to similar codes.

Supervised
Use labels or neighbourhood graph to inform map.
Often, this information is not available!

Weakly supervised
Use of temporal coherence to guide learning.
Application to zero-shot learning.
Where to go from here?

- Architectural improvements, (e.g. going deeper, more efficient use of parameters, multi-scale pathways, etc.), will continue to make impact.

- Databases will only continue to grow, so efficiency of search (e.g. Hashing) will be important.

- Approaches will roll out to domains beyond images, audio and text.

Multi-modal learning (next talk)
Thank You!