Learning from the Web

Presenter: Meng Song
4-20-2015
Web images

• Large-scale
Knowledge

- Knowledge: concepts + relationships
- Concept: an abstraction or generalization from experience
  - Objects, scenes, actions
Knowledge

- Relationships between concepts
  - is-a
  - is-a-part-of
  - is / has
  - ...

Wheel is a part of car
Mapping between Concept and Images

• Images within the same semantic concept are not necessarily visually similar.

• Intra-class variation is very large, sometimes greater than inter-class variation.
Mapping between Concept and Image Cluster

- Images within the same semantic concept are not necessarily visually similar.
- Polysemy (multiple meanings of a concept)
Self-learning (Bootstrapping)

1. Labeled Seed Examples
   - Amphitheatre
2. Unlabeled Data
3. Select Candidates
   - Amphitheatre
4. Retrain Models
5. Add to Labeled Set
Challenges

- Trade-off between purity and diversity
  - Keep intra-class diversity
  - Avoid semantic drift
OPTIMOL: Automatic Online Picture Collection via Incremental Model Learning

- Only modeling isolated object categories
- Use thresholds to control intra-category diversity and avoid semantic drift
- Datasets are small and lack richness
NEIL: Never Ending Image Learning

- Scaling up
  - 2.5 month: 400K instances, 1700 relationships
- Automatically\textsuperscript{[1]} extract common sense relationships
- Use relationships as constraints to avoid semantic drift

\textsuperscript{[1]} A. Shrivastava and S. Singh and A. Gupta, Constrained Semi-Supervised Learning using Attributes and Comparative Attributes, ECCV 2012
LEVAN: Learning Everything about Anything

- **Intra-concept** modeling
  - Learning a model that exhaustively cover all appearance variances of a concept
  - Vocabulary is discovered from Google-book n-grams
    - Extensive and concept-specific
OPTIMOL: Automatic Online Picture Collection via Incremental Model Learning

Jia Li, Fei-fei Li
Motivation

- Automatically collect a larger and more diverse dataset
Framework

Initial batch learning

Keyword: accordion

Classification

Category Model

Background

Object

Related images

Accepted images

Images held in the cache

Incremental learning

Seed images

1st group (100 images)

2nd group (100 images)

More groups

Enlarged permanent dataset

Seeds

New images to the dataset
Modeling Categories

- Generative graphical model
  - Directly model data: \( p(x), p(x|y), p(x,y) \)

- For object image and background image, build a HDP (Hierarchical Dirichlet Process) model respectively
  - The number of clusters can be adjusted to accommodate new data
  - This model provides natural clustering of data
    - Sub-category clustering
    - Solving polysemy problem
HDP: Hierarchical Dirichlet Process

- Model each image as unordered bag of visual words

Image patches: x

Topics: clusters of image patches
View Clustering from Mixture Model Perspective

- **Data**
  \[ x = \{ x_1, x_2, \ldots, x_n \} \]

- Each cluster is modeled by a distribution

- For each data item \( x_i \)
  - Pick a cluster \( z_i \) from clusters
  - Generate \( x_i \) from corresponding distribution

\[
p(x|\theta) = \prod_{i=1}^{n} \sum_{z_i=1}^{K} \pi_{z_i} p(x_i|z_i, \theta_{z_i})
\]
HDP: Hierarchical Dirichlet Process

- Mixture model with unbounded number of clusters \( K \to \infty \)
- Encode prior belief that data arise from complex processes that cannot be described with a finite mixture model.
HDP: Hierarchical Dirichlet Process

• Hierarchical model:
  • Sharing global topics across images
  • Image-specific mixture proportion

![Image Diagram](image1, image2, image3)
How to generate an image patch from HDP

- Sample component distributions from Dirichlet process
- For image j, sample mixture proportion

\[ \pi_j = (\pi_{j1}, \pi_{j1}, \ldots, \pi_{jn}) \]

- For each patch \( x_{ji} \)
  - Sample topic index \( z_{ji} \)
  - Sample \( x_{ji} \) from multinomial distribution \( F(\theta^c_{z_{ji}}) \)
Learning Model Parameters

- Fit the data to the model
- The model parameters and hidden variables are updated iteratively
- Inference: Gibbs sampling
  - Sample local clusters
    - Choose an existing cluster
    - Sample a new cluster
  - Sample global topics
    - Choose an existing topic
    - Sample a new topic
Learning Model Parameters

- Incremental learning
  - Instead of using both new data and old data to re-train the model, only the images of current iteration are used.

\[ z_j \sim p(z|\Theta^{j-1}, I_j) \quad \Theta^j \sim p(\Theta|z_j, \Theta^{j-1}, I_j) \]

Batch vs. Incremental learning
Image Classification

• Choose the category model with higher probability of generating the image

\[ P(I|c) = \prod_i \sum_z P(x_i|z, c) P(z|c) \]

• Penalize false positives
  • Only accept an image when the risk of accepting it is lower than rejecting it.

\[
R_i(A|I) < R_i(R|I) \\
\frac{P(I|c_f)}{P(I|c_b)} > \frac{\lambda_{Ac_b} - \lambda_{Rc_b}}{\lambda_{Rc_f} - \lambda_{Ac_f}} \frac{P(c_b)}{P(c_f)}
\]

inline skate
Increasing Class Diversity

- Only use cache set for incremental learning
- Images with high entropy are more likely to have new topics

\[ H(I) = - \sum_z p(z|I) \ln p(z|I) \]
Experiments and Results

• Datasets:
  • Caltech 101
  • Web-23
  • Princeton-23
  • Fergus ICCV’05 dataset

• The number of categories: 7 to 101
• The number of images in each category: ~100 to ~1000
Experiments and Results

• Image collection and annotation results
  • Collected more or similar related images than other datasets
Experiments and Results

- Image collection and annotation results
  - Collected more related images than other datasets
Experiments and Results

• Classification results
  • Learned good model

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>airplane</td>
<td>76.0</td>
<td>14.0</td>
<td>0.3</td>
<td>5.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>car</td>
<td>1.0</td>
<td>94.5</td>
<td>0.3</td>
<td>4.5</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>face</td>
<td>0.5</td>
<td>1.4</td>
<td>82.9</td>
<td>3.7</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>guitar</td>
<td>2.2</td>
<td>4.9</td>
<td>5.6</td>
<td>60.4</td>
<td>13.3</td>
<td>0.2</td>
</tr>
<tr>
<td>leopard</td>
<td>1.0</td>
<td>2.0</td>
<td>1.0</td>
<td>5.0</td>
<td>89.0</td>
<td>1.0</td>
</tr>
<tr>
<td>motorbike</td>
<td>0.3</td>
<td>5.5</td>
<td>0.3</td>
<td>5.5</td>
<td>1.0</td>
<td>67.3</td>
</tr>
<tr>
<td>watch</td>
<td>1.7</td>
<td>5.5</td>
<td>17.7</td>
<td>11.0</td>
<td>5.5</td>
<td>5.0</td>
</tr>
</tbody>
</table>

[Bar chart showing performance comparison between OPTIMOL and Fergus et al. 2005.]
Q&A

Thank you!