Online application of image attributes
Why are attributes relevant to image search?

- Human understandable
- Support familiar keyword-based queries
- Composable for different specificities
- Efficiently divide space of images
Attributes are composable

Caucasian  Teeth showing  Outside  Tilted head
Attributes efficiently divide the space of images

Female  Caucasian  Eyeglasses  Older

$k$ attributes can distinguish $2^k$ categories
Attributes in “Online” tasks

• Image search
  – Attributes as keyword queries
  – Multi-attribute queries
  – Attributes for relevance feedback

• Human-in-the-loop recognition
## Learning a Face Attribute Classifier

<table>
<thead>
<tr>
<th>Training images</th>
<th>Low-level features</th>
<th>Feature selection</th>
<th>Train classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Males</td>
<td>RGB, HoG, HSV</td>
<td>RGB, Nose</td>
<td>Gender classifier</td>
</tr>
<tr>
<td></td>
<td>HoG, Eyes</td>
<td>HoG, Eyes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HSV, Hair</td>
<td>HSV, Hair</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edges, Mouth</td>
<td>Edges, Mouth</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Females</td>
<td>RGB, HoG, HSV</td>
<td>RGB, Nose</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HoG, Eyes</td>
<td>HoG, Eyes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HSV, Hair</td>
<td>HSV, Hair</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edges, Mouth</td>
<td>Edges, Mouth</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Slide credit: Neeraj Kumar
Attributes in “Online” tasks

• Image search
  – Attributes as keyword queries
  – Multi-attribute queries
  – Attributes for relevance feedback

• Human-in-the-loop recognition
Issues in multi-attribute queries

• Calibration of attribute classifiers
  – Distances in raw multi-dimensional “attribute space” need not capture perceptual similarity

• Fusing the specified attributes in query
  – How to rank database images based on satisfying all attributes?

• Providing context for the query
  – Pairs of attributes are often correlated
**Problem**: Outputs from SVM do not follow a standard distribution.

SVM decision score distributions for 3 classifiers:
Fused Multi-Attribute queries

– How to rank database images based on how well they satisfy all attributes?

– Multiply attribute probabilities, but
  • Attributes often correlated
  • May be no database image satisfying all attributes
Interactive visual search

- Iteratively refine the set of retrieved images based on user feedback on results so far
- Potential to communicate more precisely the desired visual content

Slide credit: Adriana Kovashka
WhittleSearch: relative attribute feedback

Kovashka et al., CVPR 2012

**Query:** “white high-heeled shoes”

**Initial top search results**

**Feedback:** “more formal than these”

**Refined top search results**

**Feedback:** “shinier than these”

Whittle away irrelevant images via precise attribute feedback

Slide credit: Kristen Grauman

Attributes in “Online” tasks

• Image search
  – Attributes as keyword queries
  – Multi-attribute queries
  – Attributes for relevance feedback

• Human-in-the-loop recognition
Types of Recognition

**Basic-Level**
- Airplane?
- Chair?
- Bottle?...

**Parts & Attributes**
- Yellow Belly?
- Blue Belly?...

**Subordinate**
- American Goldfinch?
- Indigo Bunting?...

---

<table>
<thead>
<tr>
<th>Humans</th>
<th>Easy</th>
<th>Easy</th>
<th>Hard, limited memory &amp; experiences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computers</td>
<td>Some Success</td>
<td>Some Success</td>
<td>Hard, but can store large knowledge bases</td>
</tr>
</tbody>
</table>

Slide credit: Steve Branson
Recognition With Humans in the Loop

Wah et al., Multi-class Recognition and Part Localization with Humans in the Loop, ICCV 2011

- Computers: reduce number of required questions
- Humans: drive up accuracy of vision algorithms
Example Questions: Localize

Wah et al., Multi-class Recognition and Part Localization with Humans in the Loop, ICCV 2011
Example Questions: Name attributes

You will be asked to answer a series of questions based on identifying visual features from the bird image on the left. Closely follow the specific instructions for each question. Holding the mouse over each selectable option for 1 second will provide additional instructions or examples.

What is the color of the underparts of the bird?

Select at least one. If the underparts aren't visible, make your best guess, then select "Guessing". If the color is a mixture of two colors, select both (e.g., for blue-green select blue and green). If the underparts have multiple regions or patterns with multiple colors, select all relevant colors (e.g., for yellow with black stripes, select yellow and black).

Wah et al., ICCV 2011
**Basic Algorithm**

Input Image \( (x) \)

1. **Computer Vision**
   - Max Expected Information Gain

2. **Question 1:** Click on the belly
   - Answer: \( (x,y) \)
   - \( p(c \mid x, u_1) \)

3. **Question 2:** Is the bill hooked?
   - Answer: YES
   - \( p(c \mid x, u_1, u_2) \)

\[ \ldots \]

Wah et al., ICCV 2011

Slide credit: Steve Branson
Summary: Attributes for online tasks

Attributes give human user way to
- Communicate search queries
- Give feedback in interactive search
- Assist in interactive recognition
Multi-Attribute Queries: To Merge or Not to Merge

Mohammad Rastegari, Ali Diba, Devi Parikh, Ali Farhadi
Introduction

• Multi-Attribute Queries
  – Find Criminal, or Loved ones
  – Specific Illustrations, photographers

• Common Way of Dealing
  – Train classifier separately for each attribute, combine the highest score

• Exhaustive Solution
  – Train all combinations
  – Test on the held-out dataset, choose the best
  – Expensive and slow (Exponential to the queried attributes)
Users often have very specific visual content in mind that they are searching for. The most natural way to communicate this content to an image search engine is to use keywords that specify various properties or attributes of the content. A naive way of dealing with such multi-attribute queries is the following: train a classifier for each attribute independently, and then combine their scores on images to judge their fit to the query. We argue that this may not be the most effective or efficient approach. Conjunctions of attribute often correspond to very characteristic appearances. It would thus be beneficial to train classifiers that detect these conjunctions as a whole. But not all conjunctions result in such tight appearance clusters. So given a multi-attribute query, which conjunctions should we model? An exhaustive evaluation of all possible conjunctions would be time consuming. Hence we propose an optimization approach that identifies beneficial conjunctions without explicitly training the corresponding classifier. It reasons about geometric quantities that capture notions similar to intra- and inter-class variances. We exploit a discriminative binary space to compute these geometric quantities efficiently. Experimental results on two challenging datasets of objects and birds show that our proposed approach can improve performance significantly over several strong baselines, while being an order of magnitude faster than exhaustively searching through all possible conjunctions.

1. Introduction

We often find ourselves searching for images with very specific visual content. For instance, if we witness a crime we might help law enforcement agents search through mugshots of criminals to find the specific individual we saw. Victims of disasters may search through hospital databases to find missing loved ones. Graphic designers may search for illustrations of specific styles. Bird watchers may search for photographs of birds with a particular appearance to identify its species. In such scenarios, the most natural way for users to communicate their target visual content is to describe it in terms of its attributes [3, 7] or visual properties. Given the specificity of the desired content, the user typically needs to specify multiple attributes in order to appropriately narrow the search results down.

A common way of dealing with such multi-attribute queries is to train classifiers for each of the attributes independently, and then combine their scores on images to judge their fit to the query. We argue that this may not be the most effective or efficient approach. Conjunctions of attributes often correspond to very characteristic appearances. It would thus be beneficial to train classifiers that detect these conjunctions as a whole. But not all conjunctions result in such tight appearance clusters. So given a multi-attribute query, which conjunctions should we model? An exhaustive evaluation of all possible conjunctions would be time consuming. Hence we propose an optimization approach that identifies beneficial conjunctions without explicitly training the corresponding classifier. It reasons about geometric quantities that capture notions similar to intra- and inter-class variances. We exploit a discriminative binary space to compute these geometric quantities efficiently. Experimental results on two challenging datasets of objects and birds show that our proposed approach can improve performance significantly over several strong baselines, while being an order of magnitude faster than exhaustively searching through all possible conjunctions.
To Merge or not to Merge

• Proposed an optimization approach:
• Given a multi-attribute query, efficiently identifies which components would be beneficial.
• **Intra-class variance:** Geometric notions that capture the compactness of the set of images that satisfy a combination
• **Inter-class variance:** Margin of these images from other distractor images provide good proxies for the likely effectiveness of a classifier trained to recognize the combination
• Test On aPascal and Bird200
Tightness and Margin

Training a merged red-blue classifier.

Purple (intersect of red and blue) have small Diameter (D) and enough margins (K) with rest dots.
Choice

• If tight (D) and margin (K) are both beneficial (Small D and large K), then train the combination is beneficial.

• Otherwise we train them separately.

• Proposed a mapping:
  – Images Feature Space $\rightarrow$ Binary space with discriminative properties preserved
Notations

- We have n attributes: \( A = \{a_1, \ldots, a_n\} \).
  - \{white, furry, dog\}
- Set of powerset of A: \( S = \{S_1, \ldots, S_m\} \), \( m = 2^n \).
  - \{white\}, \{furry, dog\}, \{white, furry\}
- Set of Combinations \( c \): subset of \( S \) covers \( A \).
  - \{\{white-furry\}, \{dog\}\}
- \( D(c) \) is the diameter of \( c \): \( \max_{x, y \in c} d(x, y) \), where \( d(x,y) \) is the distance of image \( x \) and \( y \).
- \( K(c, c') \) is the margin:
  \[
  \min_{x \in c, y \in c'} d(x, y)
  \]
Define Learnability

• Learnability of combination C:

\[ \mathcal{L}(\mathcal{C}) = \sum_{c \in \mathcal{C}} \left[ \sum_{c' \in \mathcal{C}, c' \neq c} \mathcal{K}(c, c') + \sum_{a \in c} \mathcal{K}(c, c \setminus a) - D(c) \right] \]

• a indexes attributes, \( \mathcal{K}(c, c \setminus a) \) is the margin of c that satisfy all attributes, and \( c \setminus a \) that satisfy all but one attribute.
Integer programming

• This is NP-hard optimization.
• Define a Gain Function:

\[ G(a_i, a_j) = K(a_i a_j, a_i) + K(a_i a_j, a_j) - D(a_i a_j) \]

• The higher the gain function the higher is the reward for merging two attributes. Our gain function exposes an interesting property that helps prune the search space drastically.
Greedy Method

1. Pick the pair with the biggest positive Gain
2. Combine the pair as a new attribute, remove the old pair, left with N-1 attributes
3. Continue this procedure until all attributes are covered.
4. If no positive gain in pairs, try triplets (Never happened in real experiment)
Bits Comparison

• To compute the $d(x,y)$ efficiently
• Using binary codes (e.g. LSH)
• For each dimension of the binary codes we can compute number of zero bits and number of one bits.
• $O(\text{constant})$
Experiment

• 1. On aPascal and the Caltech Bird200 dataset.

• 2. Compare methods with four different baselines.

• 3. Test our method with different binary code mapping methods

• 4. Evaluate the impact of different binary code sizes.
Baselines

- **Default (DEF):** Training classifiers for each of the attribute independently and then combining the result scores.

- **Random Selection (RND):** Randomly selects a combination from all possible combinations and learns a classifier for each component of that combination.

- **Upper Bound (UPD):** Exhaustively train all possible combinations, evaluate their performance on the test set, and select the best one.

- **Best Attribute First (BAF):** Intuitively, if an attribute predictor is accurate enough (in the limit, perfect), there is no benefit to merging it with another attribute.
Results

- aPascal 3-attributte query and 4-attributte query
Results

- Bird test set 3-attribute
Results

• Time to find the best combination and time to retrieval

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Bound</td>
<td>35.325</td>
</tr>
<tr>
<td>Ours</td>
<td>0.508</td>
</tr>
</tbody>
</table>

Time for finding best combination: Trying all possible combinations of attributes and picking the best one is very expensive.

Average Retrieval Time: Comparisons between the entire time needed to perform the default case, our method, and the upper bound. This table assumes that no classifiers for the default case are trained off line.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper Bound</td>
<td>167.68</td>
</tr>
<tr>
<td>Default</td>
<td>42.56</td>
</tr>
<tr>
<td>Ours</td>
<td>22.34</td>
</tr>
</tbody>
</table>

We now look at which attributes tend to merge with other attributes often, and which ones typically stay unmerged.
Results

• Calibration Effects

![Graph showing the comparison between calibrated and uncalibrated results]
Results

• Attributes analysis
Once merged with bird the classifier can find the right images. Retrieved images are mixed between planes and birds. This is due to the labeling in aPascal that both birds and planes wing and beaks are classifiers and red ones are for independent classifiers. It is interesting to see that when considered beak, wing and bird independently,

Figure 10. Qualitative comparisons between our method, the default case and the upper bound. Green boxes correspond to merged
Once merged with bird the classifier can find the right images. It is interesting to see that when considered beak, wing and bird independently,

Figure 10. Qualitative comparisons between our method, the default case and the upper bound. Green boxes correspond to merged

...
References

• 1. CVPR 2013 Short Course:
• https://filebox.ece.vt.edu/~parikh/attributes/