7/2
One-shot Recognition

Kevin Tang
Marshall Tappen
Rahul Sukthankar

Project Review

- Best results previously:
  - Logistic projection with 50 dimension reduction
  - Only rg-sift features
  - ~20% accuracy
  - Chance accuracy: 10%
- Goal
  - ~30% accuracy
  - Lampert: 40.5% accuracy

Extensions

- Tried a lot of different extensions
  - Using all features
    - lss, cq, phog, sift, rg-sift, surf
  - Linear projection to a higher dimension as the final layer
    - Spread points out in higher dimension
  - Multiple layers of projections at low dimension
    - More non-linearity
    - Prevent over-fitting

Results

- Using all features
  - PCA to reduce dimensionality of features
  - Concatenate reduced features together
  - ~22% accuracy
- Linear projection to higher dimension
  - Down to 10 dimensions, up to 20 dimensions
  - ~19% accuracy
  - ~18% accuracy without projection to 20 dimensions
- Multiple layers of low dimensional projections
  - Down to 4 dimensions, multiple 4 dimension layers
  - ~17% accuracy

More Extensions

- Different method of evaluating accuracy
- Current method
  - Randomly select 1 training image from each of the 10 test classes
  - Randomly select 50 additional images from the test classes for classification
  - Repeat for 10,000 iterations
- New method
  - Cluster all images from the 10 test classes into 10 clusters
  - Denote each cluster's class by majority of elements
  - Classify each image by cluster center it is closest to
  - Repeat for 10,000 iterations

More Results

- Using new method to evaluate our accuracy
  - Logistic projection to 10 dimensions
  - Only rgsift features
  - ~28.5% accuracy (preliminary)
  - Chance accuracy: 10%
- Interesting points
  - Protects against noise from randomly selected single training image
  - Some classes aren't assigned to any cluster center
- Extensions
  - Better clustering algorithm

New Ideas

- Try dividing classes into different sets for training
- Probabilistic Attributes
  - Randomly divide 40 training classes into positive set, negative set
  - Positive set “has attribute”, negative doesn’t
  - Train SVM’s to classify attribute
  - “Bag of SVM’s”, which are better?
- Discriminative Attributes
  - Take all possible pairs of 40 training classes
  - 40 choose 2 = 780 total
  - Train SVM’s to distinguish between every pair
  - “Is it more like a cat, or more like a dog?”

New Ideas

- Testing phase of both methods:
  - Given single training image from each of 10 test classes
  - Classify attributes of each training image
    - Results in an attribute vector for each test class
  - Given more images from test classes
    - Classify attributes of each testing image
    - Find closest attribute vector match from single training images

Next Steps

- Use previous extensions and evaluate with clustering method
  - Example: Use all features
  - Previous extensions could have worked, but hidden by noise from single training image
- Run probabilistic and discriminative attributes over long weekend
  - Code is all done
  - Probabilistic attributes
    - ~30 minutes to train each SVM
    - Hoping for ~200 SVM's

Next Steps

- SVM
  - Chi-squared kernels
  - Incorporate Platt scaling
- Lampert's data
  - Graciously made available all his code, original images for dataset
  - Leverage his code to provide benchmarks
    - How well does he do in a one-shot recognition case?