Large-Scale Image and Video Processing

Lexing Xie
lexing.xie@anu.edu.au
Reality in this online world

one year of digital life
~200 GB?

news broadcast
ten channels, one year
1,300 GB, 1,830 hrs

Oct’09: 4 billion photos
6000+/minute
~ 500 TB

Apr’09: 15 billion photos
+220 million/week
~ 1.5PB

Mar’10: 24 hrs/minute
10% of internet traffic
~ 12 PB/yr ??

• Many other examples:
  – Android / IPhone GMap traces
  – Mobile phone call records
  – All environmental features sensed daily over Australia?
Outline

• Large-scale, why?
  – Less prone to overfit? Better generalization
  – Data vs intelligence

• Scaling up an SVM classifier

• Processing considerations

• Data-intensive processing framework
  – Hadoop + applications
Data ≠ Power

[Banko and Brill ACL 01]

Task: confusion set disambiguation
“... initial experiments with the GIST descriptor on ten thousand images were very discouraging ... however increasing the dataset to one million yielded a qualitative leap.”

[Hays and Efros SIGGRAPH07]
ML Systems
Visual Concept Detection

Task: score each image independently w.r.t. a set of pre-defined visual concepts.
Aggregated performance over 50 "core" visual categories [Xie et al’11].

<table>
<thead>
<tr>
<th>Classifier Tags</th>
<th>Tagging Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw classifier tags (baseline)</td>
<td></td>
</tr>
<tr>
<td>&quot;ImageNet-1000&quot;, UIUC-NEC</td>
<td>80% precision, @4 tags per image</td>
</tr>
<tr>
<td>&quot;Social 20&quot; KNN-voting [Li, Snoek’09]</td>
<td></td>
</tr>
<tr>
<td>&quot;ImageNet-1000&quot;, KNN</td>
<td></td>
</tr>
<tr>
<td>&quot;ImageNet-1000&quot;, libLin*</td>
<td></td>
</tr>
</tbody>
</table>

Normalized classifier tags

Precision-calibrated tags

Taxonomy-refined tags
Support Vector Machines

\[ x^T \beta + \beta_0 = 0 \]

\[ M = \frac{1}{\|\beta\|} \]

margin

\[ M = \frac{1}{\|\beta\|} \]

\[ M = \frac{1}{\|\beta\|} \]
SVM (in ML class)

• Training

\[
\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{N} \xi_i
\]

subject to \(\xi_i \geq 0, \ y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \ \forall i,\)

Dual form: A QP

\[
L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i' = 1}^{N} \alpha_i \alpha_i' y_i y_i' \langle h(x_i), h(x_{i'}) \rangle
\]

matlab code:

\[
\text{alpha_star} = \text{quadprog(diag(z)} * y'*y* \text{diag (z)}, -\text{ones(1, Nf)}, \text{zeros(1, Nf)} ... )
\]

• Testing

\[
\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0.
\]
What hyper-parameters?

\[ \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|_2^2 + C \sum_{i=1}^{N} \xi_i \]

how much mis-classification?

Poly nominal degree $d$?

what kernel?
SVM Hyper-parameter Selection

$C = 10000$

Training Error: 0.270
Test Error: 0.288
Bayes Error: 0.210

$C = 0.01$

Training Error: 0.26
Test Error: 0.30
Bayes Error: 0.21
How to selection C and gamma?

- Performance surface w.r.t hyper-parameter highly non-regular
- K-fold cross-validation seem to approximate test err well.

[Duan et al., Neurocomputing 2003]
Where, and how much to search?

- LibSVM Guide
- [http://blog.smola.org](http://blog.smola.org)

- Normalize your data
- Center gamma around quantile stats of inverse distance in the data.
- Search C in log space.

Q: what evaluation criteria to search on?
Q: what’s the error upper bound for leave-one-out cross-validation?
SVM: Where is the computation bottle neck?

- **Training**
  \[
  \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{N} \xi_i \\
  \text{subject to } \xi_i \geq 0, \ y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \quad \forall i,
  \]
  \(\mathcal{O}(N^2) \sim \mathcal{O}(N^3)\)

- **Testing**
  \[
  \hat{f}(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + \hat{\beta}_0.
  \]
  \(\mathcal{O}(N^2)\) brute-force
  Can we even store \(K\) for large \(N\)?
  e.g. \(N=1\mathrm{e}6\)

- **Parameter selection**
  \(d\)th-Degree polynomial: \(K(x, x') = (1 + \langle x, x' \rangle)^d\),
  Radial basis: \(K(x, x') = \exp(-\gamma \|x - x'\|^2)\),
  Neural network: \(K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2)\)
  
  Adds up quickly: 5 fold x 4 values of \(C\) x 3 kernel types x 5 parameter values ...
GPU Speed up of Kernel Computation

\[ K(x, x') = \exp(-\gamma \|x - x'\|^2) \]

- Training speedup ~10x
  - 6000-dimension input data
  - Up to 3000+ training instances
- Testing speedup ??
  - Should be even more

http://mklab.iti.gr/project/GPU-LIBSVM
GPU-Tailored SVM [Cotter et al, KDD 2011]

• Key ideas
  – Load+solve working set on GPU (size 16)
  – Cluster on sparsity patterns to improve I/O
  – 1st order heuristics for choosing a working set

http://ttic.uchicago.edu/~cotter/projects/gtsvm
PSVM [Chang et. al. NIPS’07] code.google.com/p/psvm

• Key idea:
  – Use ICF (incomplete Cholesky Factorization) to approximate kernel matrix.
  – Perform column-based ICF and distributed matrix multiplication to achieve parallelization

• ~41x speed up using 50 machines
  – On 200K images
random subspace bagging

[SVM₁, SVM₂]

[Features] → [Training Examples] → [Classifiers]

[Yan, Tesic and Smith KDD07]
Many approaches for scaling up

- Large number of models vs. large models
- Some applicable to other models (e.g. graph construction)
- Other issues: normalize input, imbalanced training data, normalize output?

[Chang et. al. NIPS’07]

[Yan et. al. KDD’07]

Working set on GPU
Other parts of the ML pipeline
Example “Large” Processing Tasks

• Task 1: PACAL VOC Challenge 2010
  – 20 target classes
  – 10,103 training images, 23K+ objects

• Task 2: NIST TREC Multimedia Event Detection (MED) Benchmark
  – 15 events
  – 13,000+ training videos, 32,000+ testing
    sampled video frames: 660K+ training, ~2M testing
Botticelli, da Vinci, Monet, or Miro?
Visual Words for Image Retrieval

- Vocabulary size 1K ~ 1M
- Local detectors
  - DoG, Hessian-Affine, Harris-Affine, MSER, Harris-Corner, etc.
- Local descriptors
  - SIFT, PCA-SIFT, GIST, SURF, etc.
- Different coding methods
  - K-means, hashing, sparse coding, etc.
Back-of-the-Envelop Planning

- Store raw data: ~600GB videos
- Feature extraction (e.g. BoW)
  - BoW: 2 sec / image
    - VOC: ~3 hours; MED: 60 CPU days
    - Storing raw SIFT features: ~1TB for MED training
  - Object features: 20 secs / image for 200+ objects..
    - >1 year CPU time ... 3 months on a quad-core PC
- Learn model
  - Store 1M x 2K dims in matlab: 16.4GB!
  - SVM ~1 min per QP * #CV(5) * # param (24)
    → 2 hours!
• The good news
  – Most of these computation is data- and task-parralizable
  – Store intermediate results
  – many small files harmful (!)
    • “tar cfz”; hashing helps “machine learning”
  
• What if ...
  – PC dies?
  – Power outage?
  – Network timeout?

Dear NCI-NF User,

There will be an electrical outage at the ANU Huxley data centre on Saturday 17th September. As a result there will be a downtime from 17th Sept (12am) to 18th Sept (12am).
HBase  Pig  Hive  Chukwa
MapReduce  HDFS  Zoo Keeper
Core  Avro
Massive data storage
Distributed File Systems

Masters

Replicas

GFS Master

Chunkserver 1

C₀
C₁
C₅
C₂

Chunkserver 2

C₁
C₅
C₃

Chunkserver N

C₀
C₅
C₂

Client

Client

Client
MapReduce
Facebook example: check-in times

Summary

• Large data is the trend
• ML algorithms → practice
• Application-specific planning is important
• Many parallel computing paradigms to help
• other topics ...
  – Large amounts of sparse features
  – GMM, CRF and other models
  – Inverted index, hashing, indexing ..
Thanks! Questions?

• Slide credits
  – Winston Hsu (NTU Taiwan), Rong Yan (Facebook)
  – CIKM tutorial by Edward Chang (Google)
  – WWW’11 tutorial by Alex Smola “graphical models for the internet”
  – Flickr user @yuliyart

• Other pointers
  – John Langford large-scale tutorial@ KDD’11