ImageNet classification: fast descriptor coding and large-scale SVM training

Empowered by Innovation

Yuanqing Lin, *Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu*

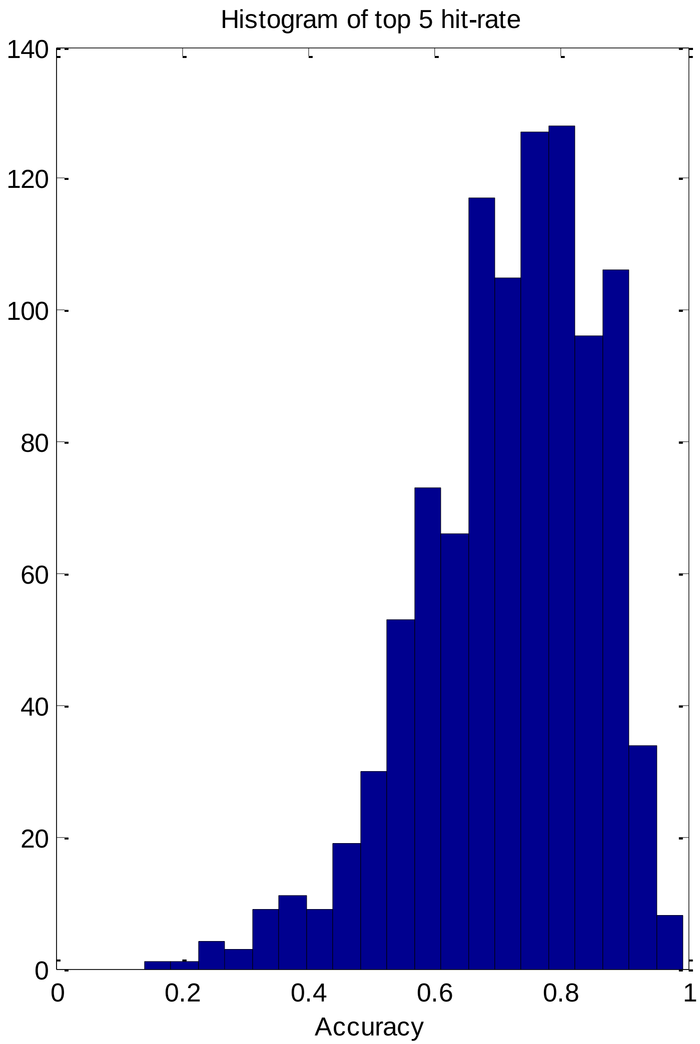
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

LiangLiang Cao, Zhen Li, Min-Hsuan Tsai, Xi Zhou, Thomas Huang

THE STATE UNIVERSITY OF NEW JERSEY

Tong Zhang

Where we are in imageNet challenge



Odometer, hodometer, mileometer, milometer 99.3%

Monarch, monarch butterfly, milkweed butterfly, Danaus plexippus, 98.0%

Cliff dwelling, 97.3%

lunar crater, 96.7%

Bonsai, 96.0%

Trolleybus, trolley coach, trackless trolley, 96.0%

Geyser, 95.3%

Snowplow, snowplough 95.3%

star anise, Chinese anise, Illicium verum, 94.0%

Our classification cost: 0.282 (top 5 hit rate, 71.8%, classification rate 52.9%)

Best performance of other teams: 0.336

System overview

Dense grid descriptor: HOG, LBP

Coding: local coordinate, super-vector

Pooling, SPM

Linear SVM

Fairly standard

Make good use of low level descriptors

How to train SVM efficiently

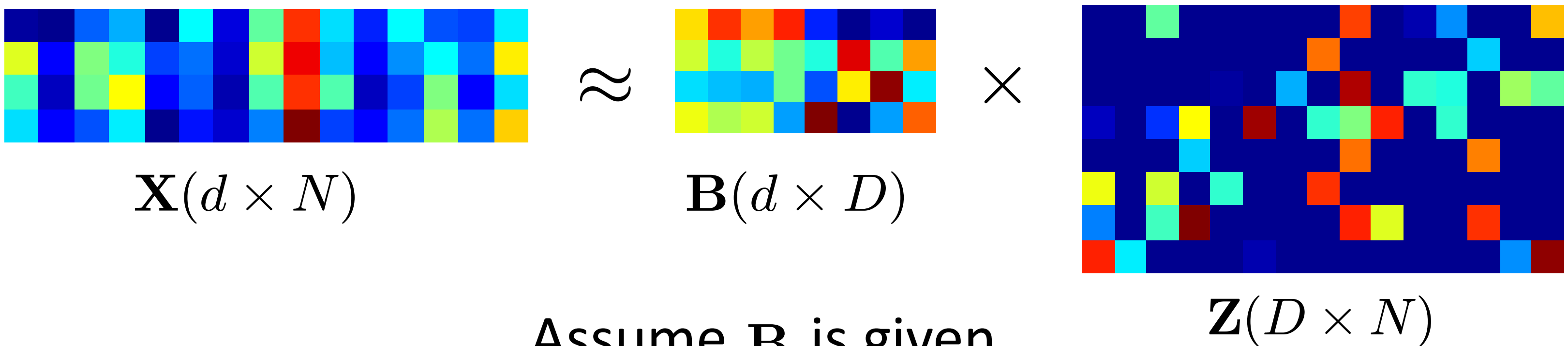
Outline

- **Fast descriptor coding**
  - Local coordinate coding (LCC)

    K. Yu et.al, NIPS2009; J. Wang et. al, CVPR 2010
  - super-vector coding

    X. Zhou et.al, ECCV2010
- **Large-scale SVM classification**
  - Averaged stochastic gradient descent

What is local coordinate coding (LCC)



Assume $\mathbf{B}$ is given.

Sparse coding:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{x} - \mathbf{B}\mathbf{z}\|^2 + \lambda \sum_{i=1}^{D} |z_i|$$

LCC: K. Yu et. al, NIPS 2009

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{x} - \mathbf{B}\mathbf{z}\|^2 + \lambda \sum_{i=1}^{D} \|\mathbf{x} - \mathbf{b}_i\|^2 |z_i|$$

Explicitly enforcing locality constraint

Why LCC
-- from functional approximation point of view

$$f(\mathbf{x}) \approx \sum_{i=1}^{D} z_i(\mathbf{x}) w_i$$

e.g. nonlinear separating hyperplane

$$|f(\mathbf{x}) - \sum_{i=1}^{D} z_i(\mathbf{x}) f(\mathbf{b}_i)| \leftarrow \text{Functional approximation error}$$

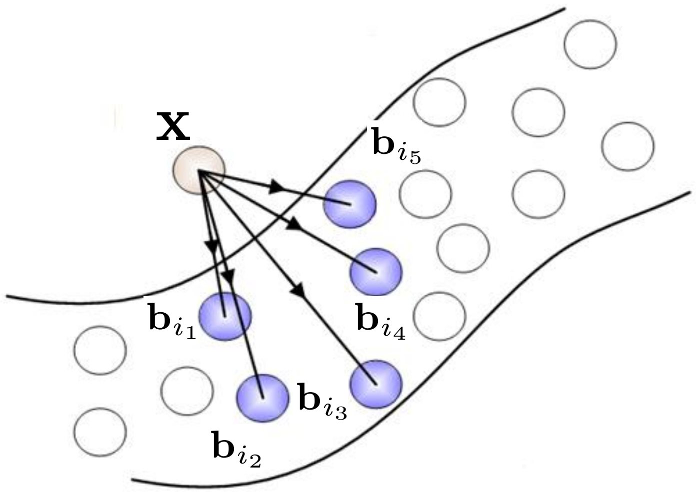
$$\leq \alpha \underbrace{\|\mathbf{x} - \mathbf{B}\mathbf{z}(\mathbf{x})\|}_{\text{Coding error}} + \beta \underbrace{\sum_{i=1}^{D} \|\mathbf{x} - \mathbf{b}_i\|^2 |z_i(\mathbf{x})|}_{\text{Locality term}}$$

- Good approximation: 1) local to the test point x
  2) small reconstruction error

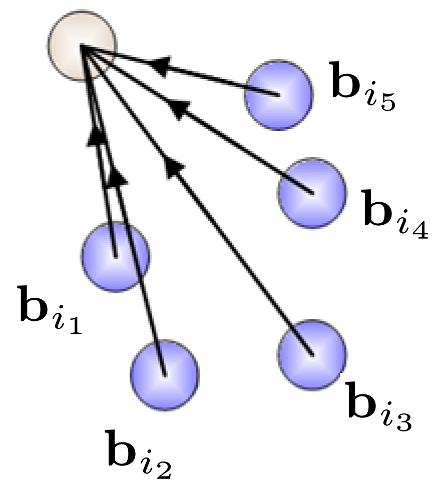
Local coordinate coding -- fast implementation

J. Wang et. al, CVPR 2009

Step 1: be local to the test point $\mathbf{x}$
-- given $\mathbf{x}$, find its KNNs.



Step 2: small reconstruction error -- solve LMS fitting using only the KNNs



- Approximated solutions, but significant speedup

For a regular image (7k patches), with D=8192:
sparse coding needs ~10mins, (approximate) LCC needs only ~2s

Parallel computing

☹ For LCC, $D = 8,192$, each image takes ~2 seconds

$$2s \times 1,200,000 \approx 28\ days$$

Not counting file I/O, networking delay, etc

☹☹ In our submission, $D = 16,384$

which would have taken **more than 56 days**

☺ With Hadoop map-reduce (about ~100 mappers), this was finished **within one day**.

System overview

Dense grid descriptor: SIFT, LBP

Coding: local coordinate, SV

Pooling, SPM

Each image is represented by a long vector

Linear SVM

Our training sets

Sets	Coding scheme	Descriptor	Coding dimension	SPM	Feature dimension	Data set Size(GB)
1	Local coordinate coding	HOG+LBP	8,192	10	81,920	167*
2		HOG	16,384	10	163,840	187*
3		HOG+LBP	20,480	10	204,800	260*
4	Super-vector coding	HOG	32,768	8	262,144	1374
5		HOG+LBP	51,200	4	204,800	1073
6		HOG	65,536	4	262,144	1374

**In sparse format*

- Very high dimensional features, huge data sets
- LCC features have smaller size -- they are sparse

How monster is the resulting feature sets

Compare to PASCAL classification task:

	# of training data	# of class	(assumed) training time
PASCAL	10,103	20	1 hour
ImageNet	1,200,000	1000	6000 hours = 250 days*
Ratio	120	50	6000

* Not including file I/O, networking delay, etc

☹ Life is short -- we need efficient SVM training algorithms

SVM using averaged stochastic gradient descent (ASGD)

One-against-all SVM classifier:

$$L = \sum_{t=1}^{T} L(\mathbf{w}, \mathbf{x}_t, y_t) = \sum_{t=1}^{T} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \max\left[0, 1 - y_t(\mathbf{w}^T\mathbf{x}_t + b)\right]$$

Stochastic update:

$$\mathbf{w}^t = \mathbf{w}^{t-1} - \eta \nabla L(\mathbf{w}, \mathbf{x}_t, y_t)$$

$$\bar{\mathbf{w}}^t = (1 - 1/t)\bar{\mathbf{w}}^{t-1} + \mathbf{w}^t/t$$

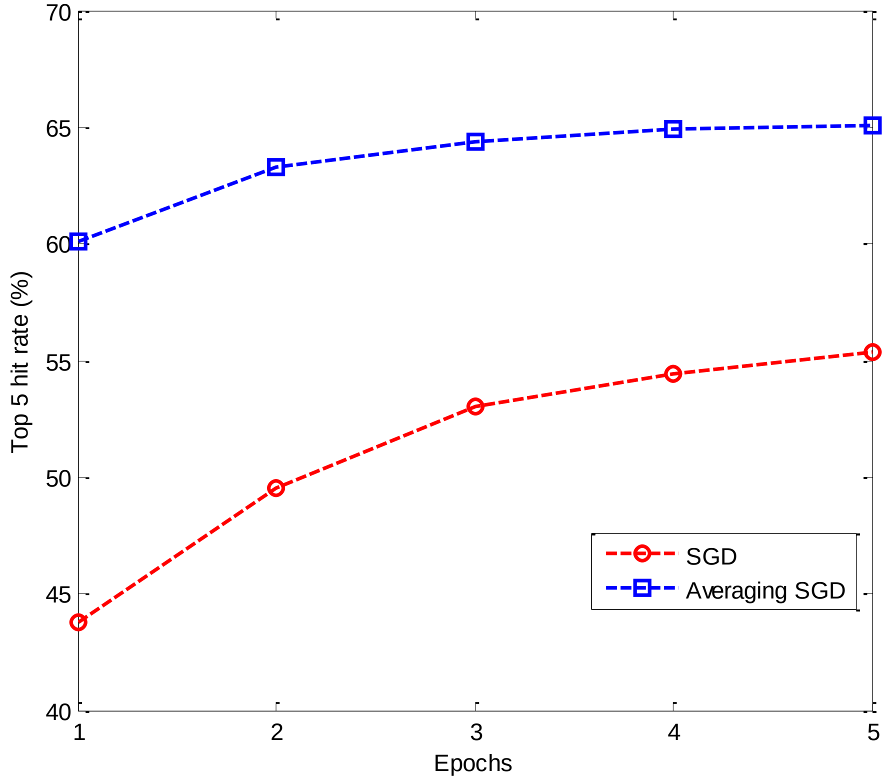
B. Polyak and A. Juditsky, 1992

☺ Memory efficient: only need to load data one-by-one

☺ Easy to parallelize: distribute the training of 1000 binary classifiers to different machines

☺ Fast convergence: need only a small number of epochs…

Fast convergence of ASGD



☺ Significant speed-up by averaging:

5 epochs already give fairly good results.

☺ ASGD: has similar convergence properties as Stochastic Newton methods when appropriate stepsize is chosen

☺ Training time: LCC features, ~ 2days (using two 8-core machines)

Super-vector features, ~ 7 days (using three 8-core machines)

Conclusion

What's the key:

1) learning: local coordinate coding and supervector coding + linear SVM
2) being able to handle large-scale data

Best single method: ~65%

Combined the 6 sets of features: 71.8%

Long way to go:

Our method performs poorly on some categories…

Long way to go …

China tree, false dogwood, 14.0%

logwood, logwood tree, 20.0%

shingle oak, laurel oak, 23.3%

red beech, brown oak, 25.3%

Kentucky coffee tree, 26.7%

cap opener, 26.7%

alder, alder tree, 29.3%

teak, Tectona grandis, 29.3%

iron tree, 30.0%

grass pink, Calopogon pulchellum, 31.3%

- Better features: Hierarchical coding, discriminative coding
- More data


Thank you