Sparsity and Learning Large Scale Models

Eric Xing

epxing@cs.cmu.edu

Machine Learning Dept./Language Technology Inst./Computer Science Dept.

Carnegie Mellon University

Acknowledgement: Bin Zhao, Xi Chen, Jun Zhu, Seyoung Kim @ CMU; and Jia Li, Hao Su, and Li Fei-Fei @ Stanford

Machine Learning problems are getting BIG

13 million Wikipedia pages

facebook

500 million users

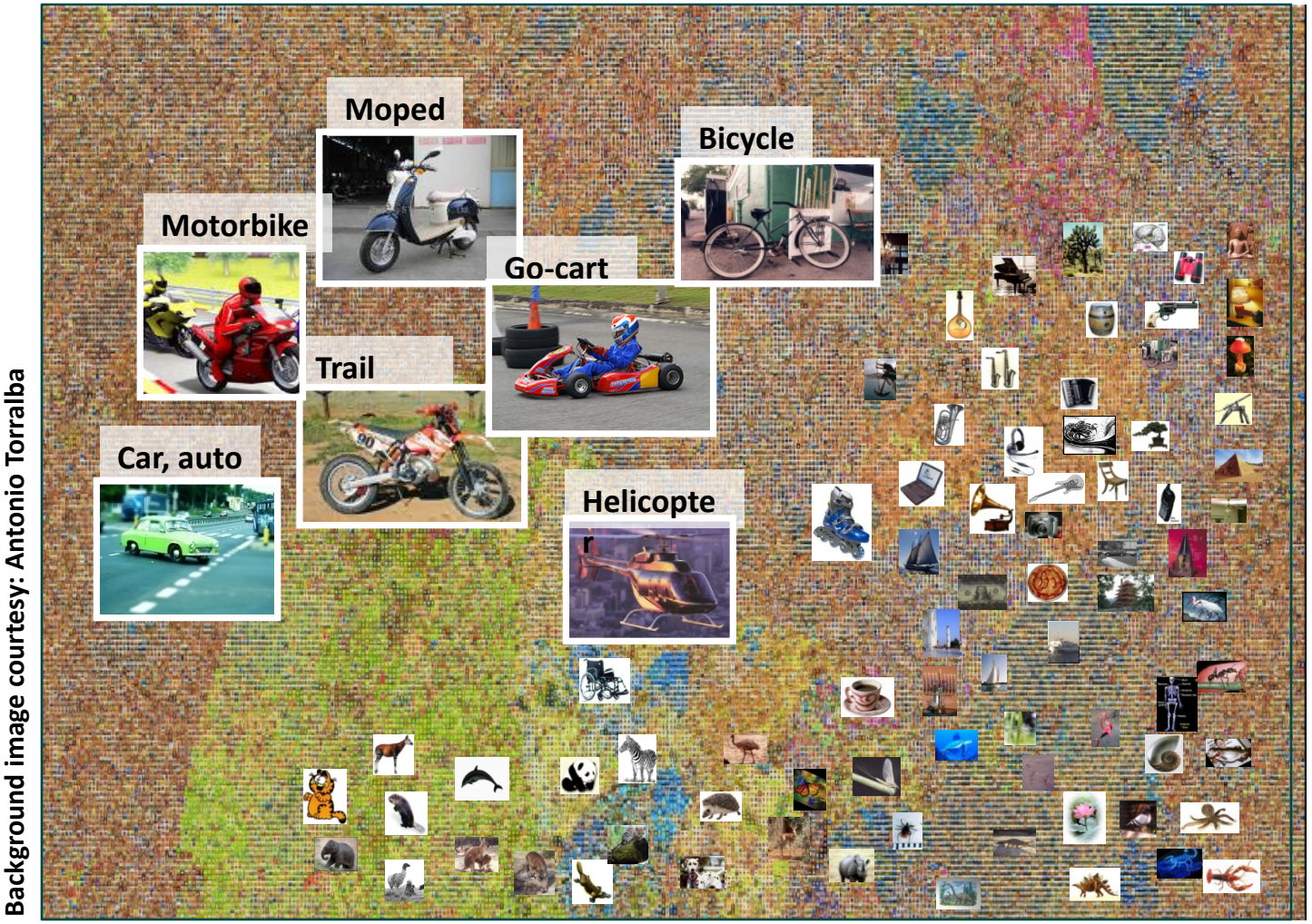
flickr

3.6 billion photos

YouTube

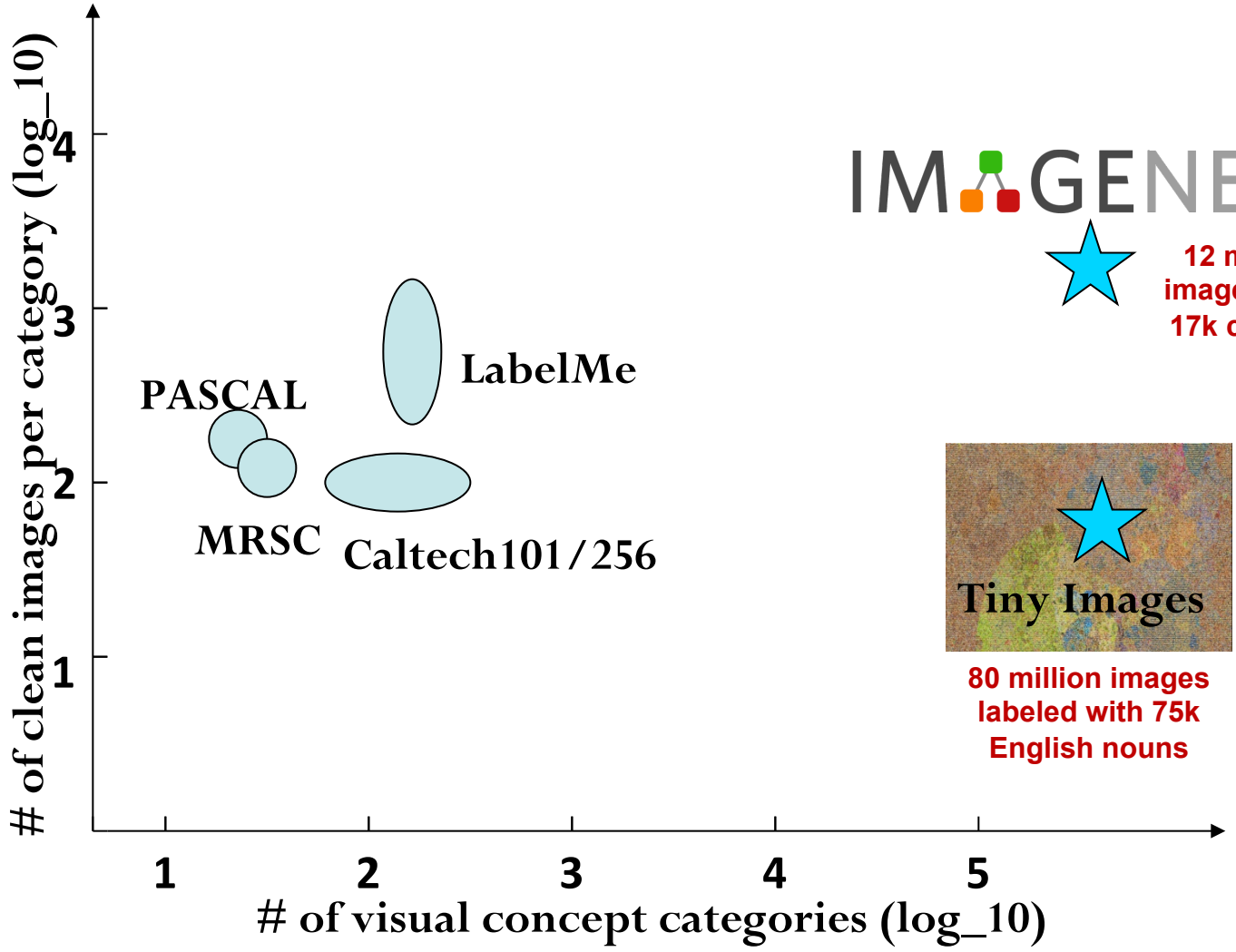
24 hours videos uploaded per minute

So do Computer Vision problems …



Courtesy L. Fei-Fei

Computer Vision problems are getting BIG



Courtesy L. Fei-Fei

When facing large scale data …

- Most popular ML approaches:
  - K Nearest Neighbor
  - Binary linear SVM
  - Least Square Regression

- Why are recent advancements of ML missing?
  - Time/Memory demanding
  - Never successfully tested on large scale data

Large Scale Problems

- How large are we talking about
  - Data can NOT fit into memory
- Example: ImageNet, large scale in **3 dimensions**
  - Data: 12 million images
  - Features: ~1 million (number comes from the top performing system in ILSVRC10, [Lin et al. 2011])
  - Classes: 17k classes

Toward Large Scale Problems:

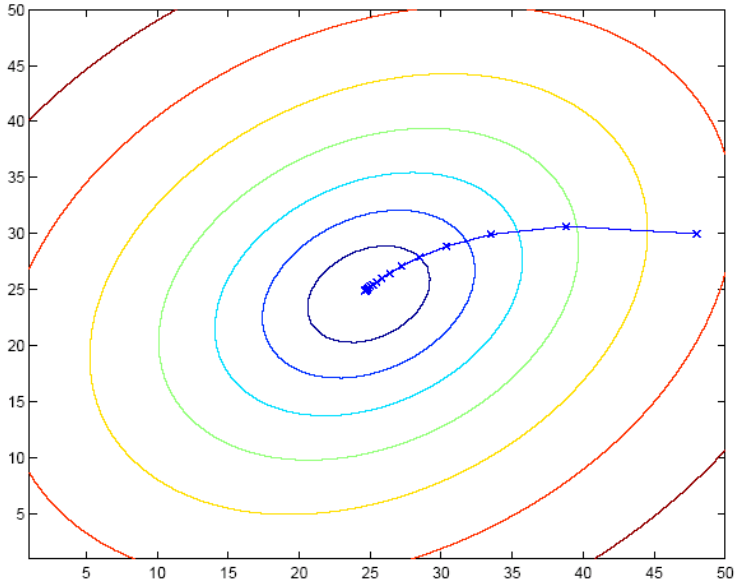
- Large data size:
  - Stochastic methods
  - Parallel computation, e.g., Map-Reduce
- Large feature dimension:
  - Sparsity-inducing regularization
  - Structured sparsity
  - Sparse coding
- Large concept space:
  - Multi-task and transfer learning
  - Structured sparsity

Toward Large Scale Problems:

- Large data size:
  - Stochastic methods
  - Parallel computation, e.g., Map-Reduce
- Large feature dimension:
  - Sparsity-inducing regularization
  - Sparse coding
  - Structured sparsity
- Large concept space:
  - Multi-task and transfer learning
  - Structured sparsity

How to optimize a (friendly) objective function?

- Friendly objection functions:
  - Convex
  - Smooth
  - Unconstrained



- Newton–Raphson
  - Pro: extremely fast converging
  - Con: need to compute Hession
- Steepest (gradient) descent:
  - Pros: conceptually clean, guaranteed convergence
  - Cons: batch, often slow converging
- Stochastic gradient
  - Pros: on-line, and perhaps less prone to local optimum
  - Cons: convergence to optimum not always guaranteed

Stochastic/online methods

- True gradient approximated by a gradient at a single example

$$Q(w) = \sum_{i=1}^{n} Q_i(w), \quad w := w - \alpha \nabla Q(w) = w - \alpha \sum_{i=1}^{n} \nabla Q_i(w),$$

$$\Downarrow$$

$$w := w - \alpha \nabla Q_i(w).$$

- Often involves several epochs
- In practice, stochastic gradient can be orders of magnitude faster
- Be careful of learning rate …

Convergence rate

- **Theorem**: the steepest descent equation algorithm converge to the minimum of the cost characterized by normal equation:

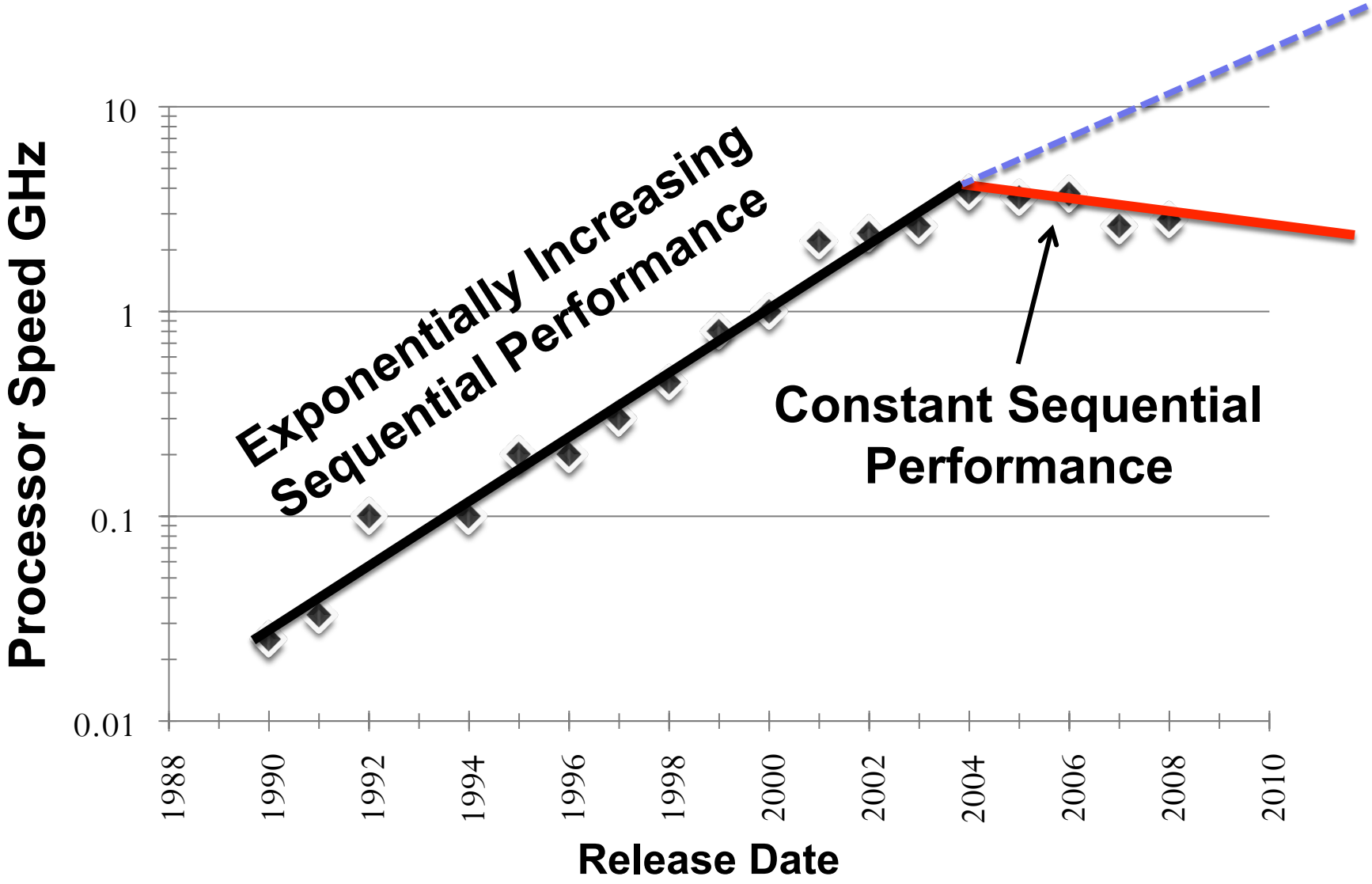
$$\theta^{(\infty)} = (X^T X)^{-1} X^T y$$

If

$$0 < \alpha < 2/\lambda_{\max}[X^T X]$$

- A formal analysis of stochastic gradient need more math-mussels; in practice, one can use a small α, or gradually decrease α.

Why Parallel Computation?



Parallel Computation

- Shared memory
  - GPU
  - Multi-core
- Cluster
  - GFS (Google)
  - HFS (Hadoop)
- Cloud computing
  - Amazon EC2, Windows Azure, etc.

Map-Reduce

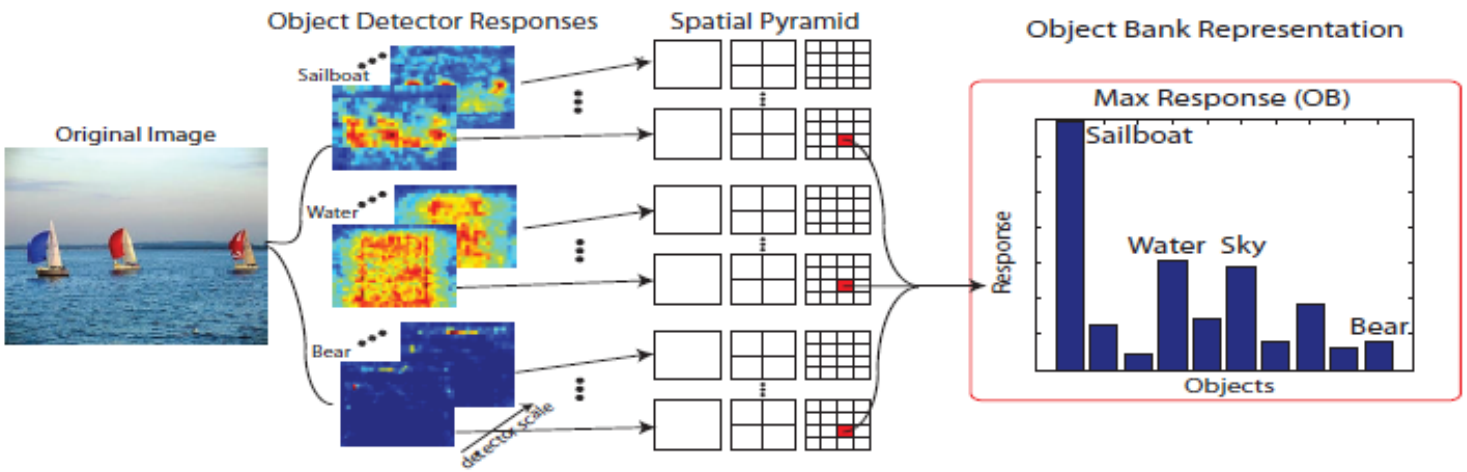
- Motivation
  - Huge data set
  - Want to use hundreds or thousands of CPUs
- Map-Reduce provides
  - Automatic parallelization and distribution
  - Fault-tolerance
  - I/O scheduling
  - Status and monitoring
- Some heavy users
  - Google, Yahoo!, Facebook, etc.

Toward Large Scale Problems:

- Large data size:
  - Stochastic methods
  - Parallel computation, e.g., Map-Reduce
- Large feature dimension:
  - Sparsity-inducing regularization
  - Sparse coding
  - Structured sparsity
- Large concept space:
  - Multi-task and transfer learning
  - Structured sparsity

Images are best represented in high-dimensional space

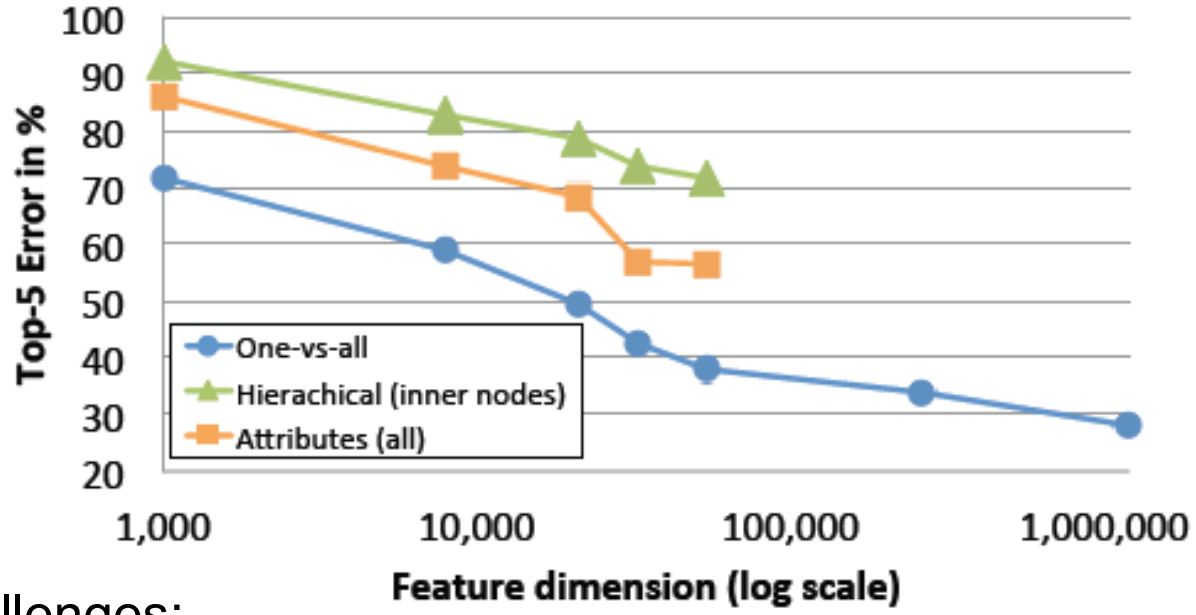
- Two examples:
  - Low-level pixels, e.g., medium resolution Caltech-101: ~300 x 300 = 90,00 dimension
  - High-level OB (Li et al., 2010): ~40,000 dimension



L.-J. Li, H. Su, E.P. Xing, & L. Fei-Fei. NIPS, 2010

Images are best represented in high-dimensional space

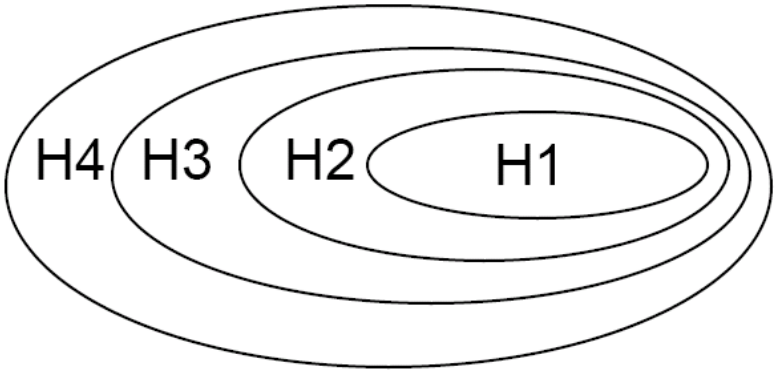
- Based on result of ILSVRC10 competition, more features often lead to better performance



- Challenges:
  - Huge parameter space
  - For multiclass problem, cannot even load parameter matrix into memory
  - Overfitting and structural risk?

Structural Risk Minimization

- Which hypothesis space should we choose?
- Bias / variance tradeoff



- SRM: choose H to minimize bound on true error!

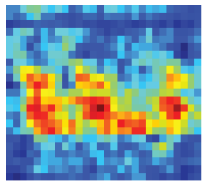
$$\epsilon(h) \leq \hat{\epsilon}(h) + O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} - \frac{1}{m}\log\delta}\right)$$

Inference in High-dimensions

- Manipulation in the feature space
  - Sparse Linear Models:
    - Feature selection, e.g., LASSO & variants --- supervised
    - Feature extraction, e.g., Sparse Coding --- unsupervised

Multivariate Regression for image classification

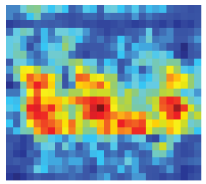
Class label | Input features | Feature strength

1 =  x ?

$$y = f(X \times \beta)$$

Multivariate Regression for image classification

Class label | Input features | Feature strength

1 =  x 

$$\beta^* = \arg\min_{\beta} (y - X\beta)^T (y - X\beta)$$

Many non-zero coefficients:
Which features are truly significant?

Sparsity

- One common assumption to make **sparsity.**
- **Makes semantic sense:** each concept is likely to be related to a small number of features, rather than all the features.
- **Makes statistical sense:** Learning is now feasible in high dimensions with “small” sample size

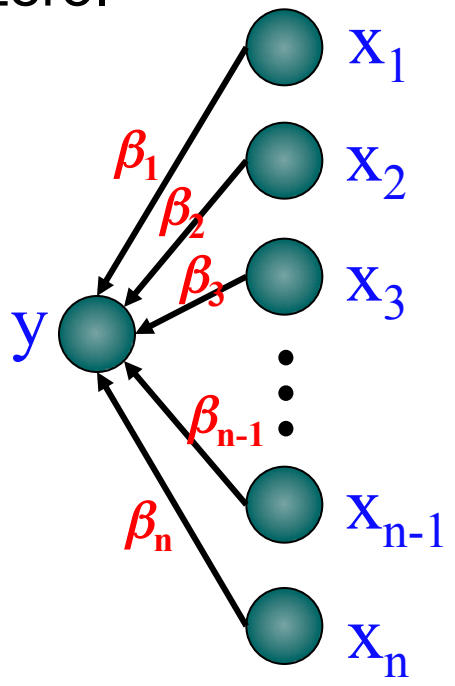
Sparsity: In a mathematical sense

- Consider least squares linear regression problem:
- Sparsity means most of the beta's are zero.

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

$$\text{subject to:}$$

$$\sum_{j=1}^{p} \mathbb{I}[|\beta_j| > 0] \leq C$$



- But this is not convex!!! Many local optima, computationally intractable.

L1 Regularization (LASSO)

(Tibshirani, 1996)

- A convex relaxation.

Constrained Form

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

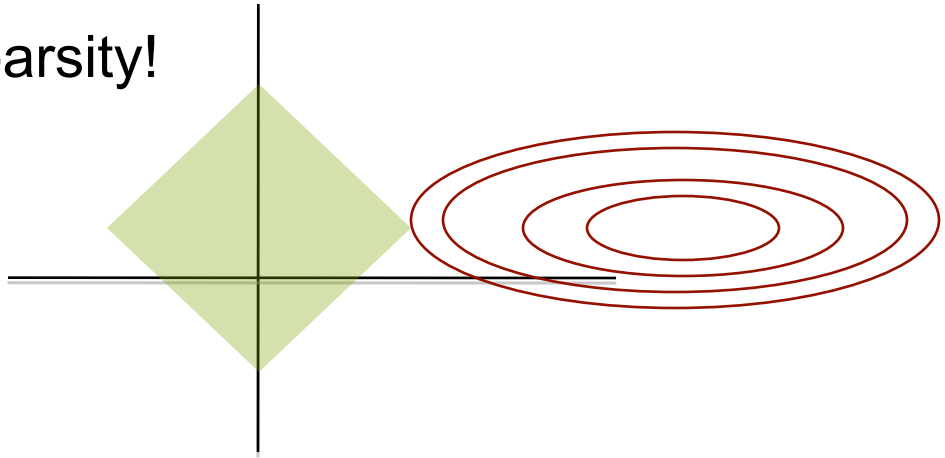
subject to:

$$\sum_{j=1}^{p} |\beta_j| \leq C$$

Lagrangian Form

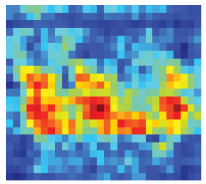
$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1$$

- Still enforces sparsity!



Lasso for Feature Selection

Class label | Input features | Feature strength

1 =  x ■ ■ ■

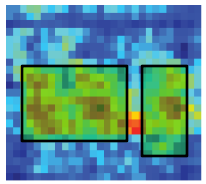
Lasso Penalty for sparsity

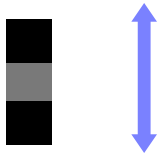
$$\beta^* = \arg\min_{\beta} (y - X\beta)^T (y - X\beta) + \lambda \sum_{j=1}^{J} |\beta_j|$$

Many zero strengths (**sparse** results), but what if the features are correlated?

"Structured" Lasso for Feature Selection

Class label | Input features | Feature strength

1 =  x



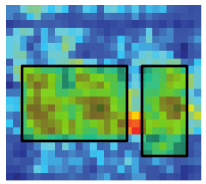
Lasso Penalty for sparsity

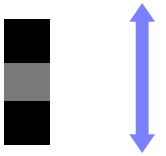
$$\beta^* = \arg\min_{\beta} (y - X\beta)^T (y - X\beta) + \lambda \sum_{j=1}^{J} |\beta_j|$$

Many zero strengths (**sparse** results), but what if the features are correlated?

"Structured" Lasso for Feature Selection

Class label | Input features | Feature strength

1 =  x



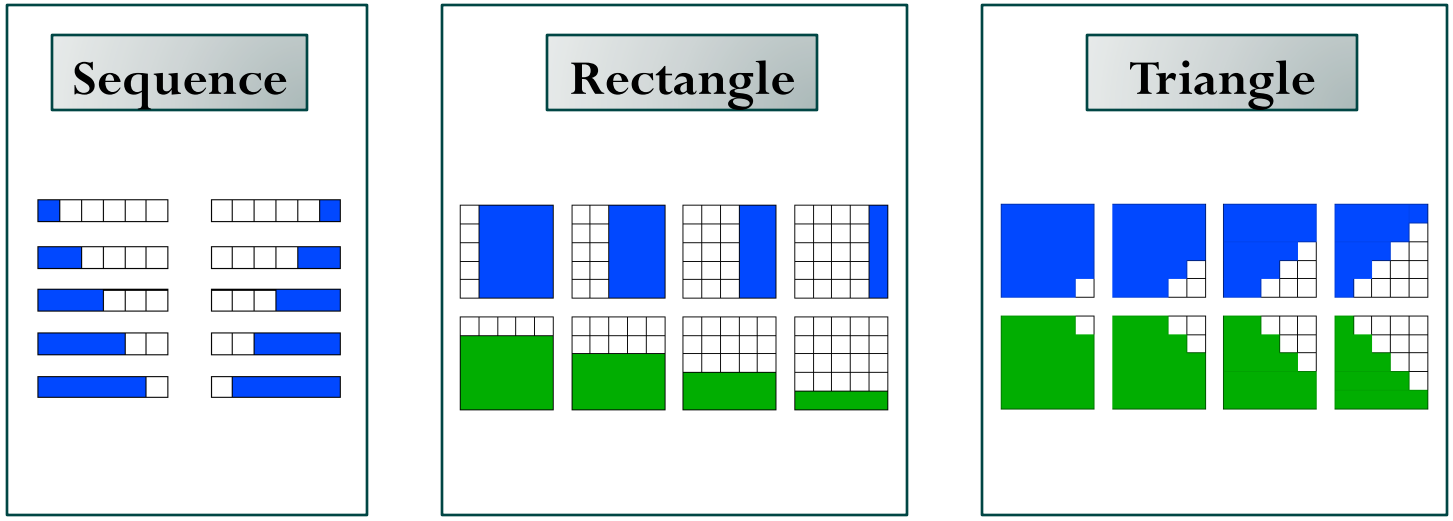
Lasso Penalty for sparsity

$$\beta^* = \arg\min_{\beta} (y - X\beta)^T (y - X\beta) + \lambda \sum_{j=1}^{J} |\beta_j|$$

***L1/L2* norm**
Structured norm $\sum_{G \in \mathcal{G}} \|\beta_G\|_2 = \sum_{G \in \mathcal{G}} \left(\sum_{j \in G} \beta_j^2 \right)^{1/2}$

Structured Sparsity [Jenatton et al., 2009]

- When penalizing with the l_1-l_2 norm
  - The l_1 norm induces sparsity at the group level
  - Inside the groups, the l_2 norm does not promote sparsity
- Examples of set of groups

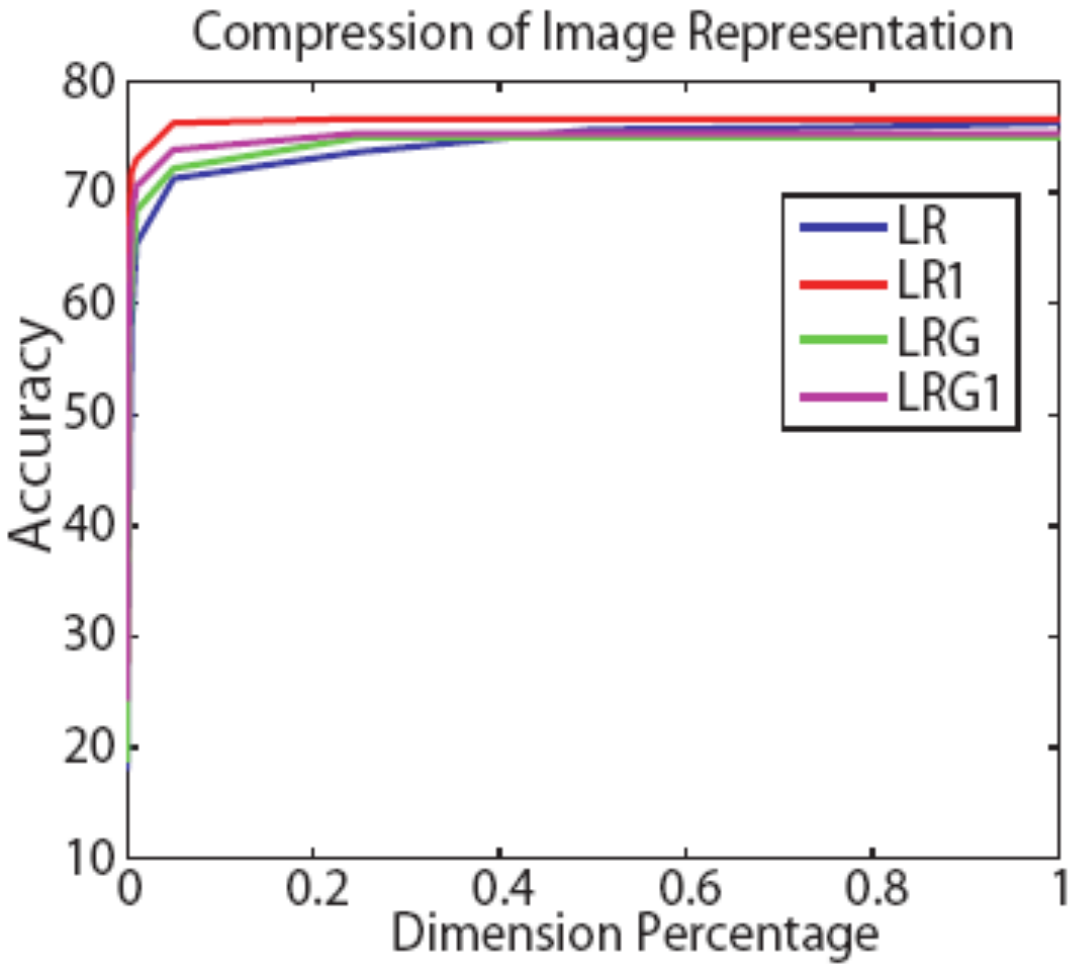


- Relationships between $\mathcal{G}$ and zero patterns

Structured Sparsity

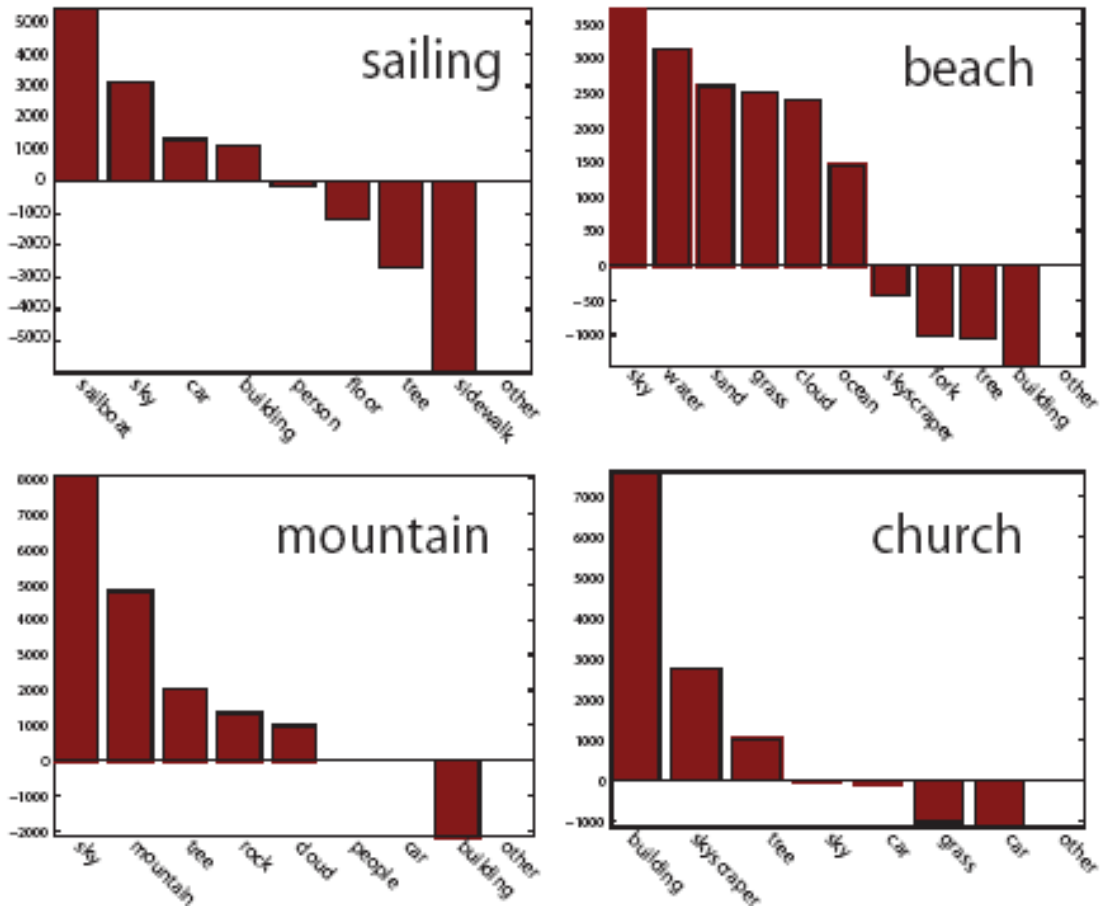
- Specific hierarchical structure (Zhao et al., 2009; Bach, 2008c)
- Union-closed (as opposed to intersection-closed) family of nonzero patterns (Jacob et al., 2009; Baraniuk et al., 2008)
- Non-convex penalties based on information-theoretic criteria with greedy optimization (Huang et al., 2009)

Some results on OB



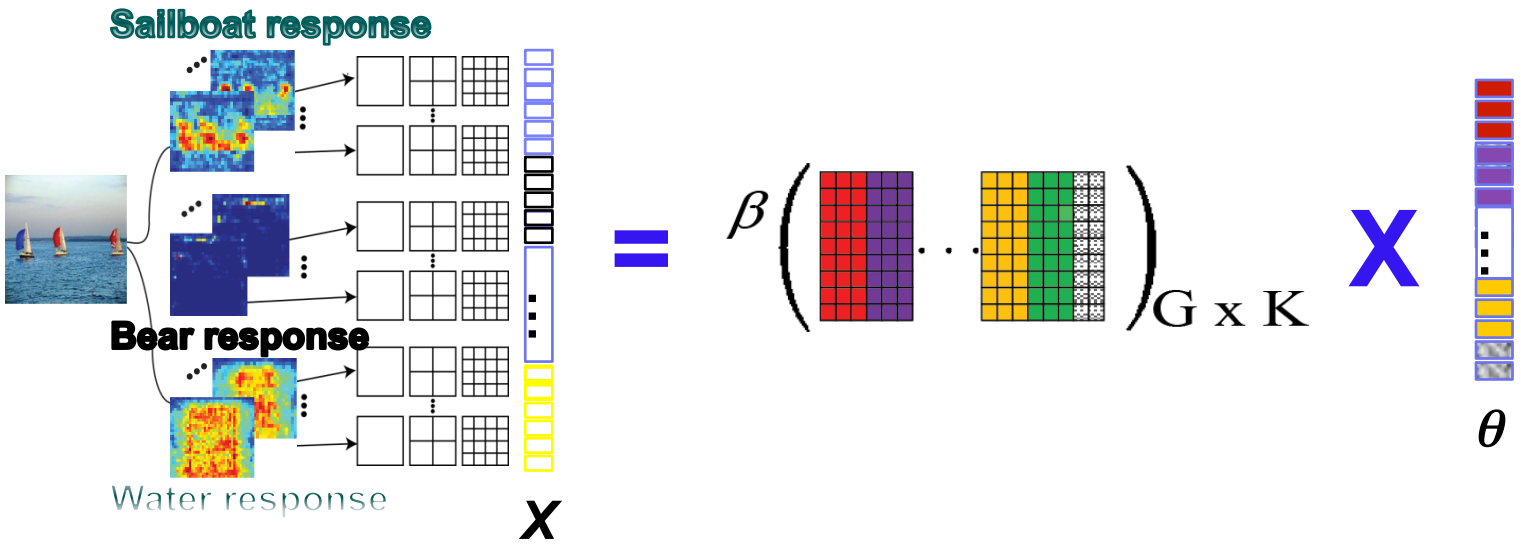
L.-J. Li, H. Su, E.P. Xing, & L. Fei-Fei. NIPS, 2010

Some results on OB



L.-J. Li, H. Su, E.P. Xing, & L. Fei-Fei. NIPS, 2010

Sparse Coding (unsupervised)



- Let X be a signal, e.g., speech, image, etc.
- Let β be a set of normalized "basis vectors"
  - We call it dictionary
- β is "adapted" to x if it can represent it with a few basis vectors
  - There exists a sparse vector θ such that $x \approx \beta\,\theta$
  - We call θ the sparse code

Primer on Sparse Coding

- Sparse Coding with appropriate constraints:

reconstruction loss **sparsity-inducing regularizer**

$$\min_{\theta,\beta} \sum_d \ell(\theta_d, \beta | \mathbf{x}_d) + \lambda \Psi(\theta)$$
$$\text{s.t.}: \quad \beta \in \Omega_1; \theta \in \Omega_2.$$

- Reconstruction loss can be:
  - the general log-likelihood loss of an exponential family distribution (Lee et al., 2010)
- Sparisty-inducing regularizer can be:
  - the L_0 "pseudo-norm": $\|\theta\|_0 := \#\{i : \theta_i \neq 0\}$ **NP-hard**
  - the L_1 norm: $\|\theta\|_1 := \sum_i |\theta_i|$ **Convex**
  - Structured regularizers, e.g., group Lasso (Bengio et al., 2009) $\|\theta\|_1 := \sum_g \|\theta_g\|_2$
- Suggests an alternating optimization procedure

Opt. Algorithm for Sparse Coding

- Much research has been done for optimizing a convex, but non-smooth objective (may subject to some constraints, e.g., non-negativity)

- Greedy algorithm for the non-convex L_0 "pseudo-norm":
  - select the element with maximum correlation with the residual
  - known as "matching pursuit" (Mallat & Zhang, 1993)

- For the convex L_1 norm, many algorithms:
  - Soft-thresholding with coordinate descent (Friedman et al., 2007; Fu, 1998; Zhu & Xing, 2011)
  - Proximal methods (Nesterov, 2007; Jenatton et al., 2010)
  - Active-set methods (Roth & Fischer, 2008)
  - Iterative Re-weighted Least Squares (Daubechies et al., 2008)
  - LARS (Efron et al., 2004) solves for regularization path
  - Online/stochastic variants
  - …

Opt. Algorithm for Dictionary Learning

- Optimize a convex and usually smooth objective w/o (convex) constraints

- General optimization procedure can be applied, less research has been done for this step
  - Projected gradient descent
  - Block-wise coordinate descent
  - …
- A recent progress is made on online/stochastic optimization method (Mairal et al., 2010)

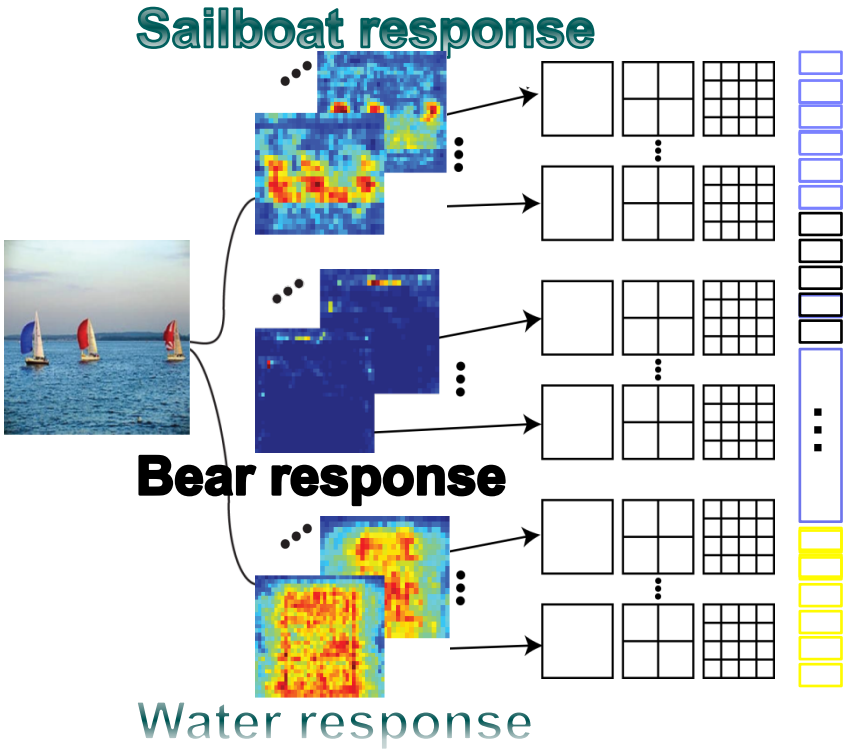
SAILING LAB
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

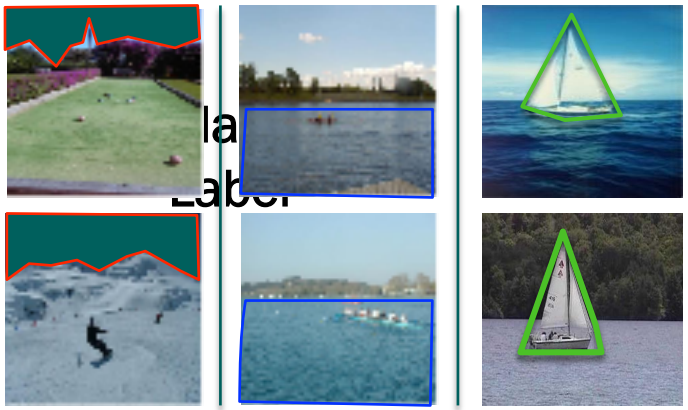
Dynamic Sparse coding for unusual event detection in videos

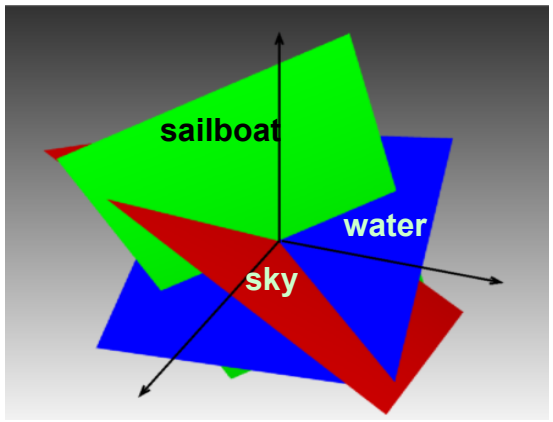
	WD	NP	LT	II	MISC	Total	FA
GT	26	13	14	4	9	66	0
ST-MRF [10]	24	8	13	4	8	57	6
Ours	25	9	14	4	8	60	5

B Zhao, L. Fei-Fei & E.P. Xing. CVPR 2011

Hierarchical Sparse Coding







Reside in Low-dimensional space

Courtesy J. Li

Hierarchical Sparse Coding

- Sparse Coding with appropriate constraints:

reconstruction loss **sparsity-inducing regularizer**

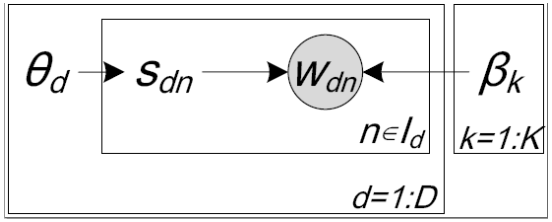
$$\min_{\theta,\beta} \sum_d \ell(\theta_d, \beta | \mathbf{x}_d) + \lambda \Psi(\theta)$$
$$\text{s.t.}: \quad \beta \in \Omega_1; \theta \in \Omega_2.$$

- Problems:
  - Encode different terms or image patches independently
  - *No high-order correlations*!
- To consider correlations:
  - Use a structured regularizer, e.g., group Lasso (Bengio et al., 2009)
  - Introduce another layer that capture the correlations (Zhu & Xing, 2011, Zhu et al., 2011, Yu et al., 2011)

Sparse Topical Coding

- Goal: design a non-probabilistic topic model that is amenable to
  - direct control on the posterior sparsity of inferred representations
  - avoid dealing with normalization constant when considering supervision or rich features
  - seamless integration with a convex loss function (e.g., svm hinge loss)

- We extend sparse coding to hierarchical sparse topical coding
  - word code θ
  - document code ***s***



reconstruction loss

sparse codes

$$\min_{\{\theta_d, \mathbf{s}_d\}, \beta} \sum_{d,n \in I_d} \ell(w_{dn}, \mathbf{s}_{dn}^\top \beta_{\cdot n}) + \lambda \sum_d \|\theta_d\|_1 + \sum_{d,n \in I_d} (\gamma \|\mathbf{s}_{dn} - \theta_d\|_2^2 + \rho \|\mathbf{s}_{dn}\|_1)$$

$$\text{s.t.}: \ \theta_d \geq 0, \ \mathbf{s}_{dn} \geq 0, \ \forall d, n \in I_d; \quad \beta_k \in \mathcal{P}, \ \forall k,$$

truncated aggregation

non-negative codes

topical bases

J. Zhu, & E.P. Xing. UAI, 2011

Opt. with Coordinate Descent

- Hierarchical sparse coding
  - for each document

$$\min_{\theta,\mathbf{s}} \quad \sum_{n\in I} \ell(w_n, \mathbf{s}_n^\top \beta_n) + \lambda\|\theta\|_1 + \sum_{n\in I}(\gamma\|\mathbf{s}_n - \theta\|_2^2 + \rho\|\mathbf{s}_n\|_1))$$
$$\text{s.t.:} \quad \theta \geq 0; \;\; \mathbf{s}_n \geq 0, \; \forall n \in I,$$

  - Word code

$$s_{nk} = \max(0, \nu_k)$$

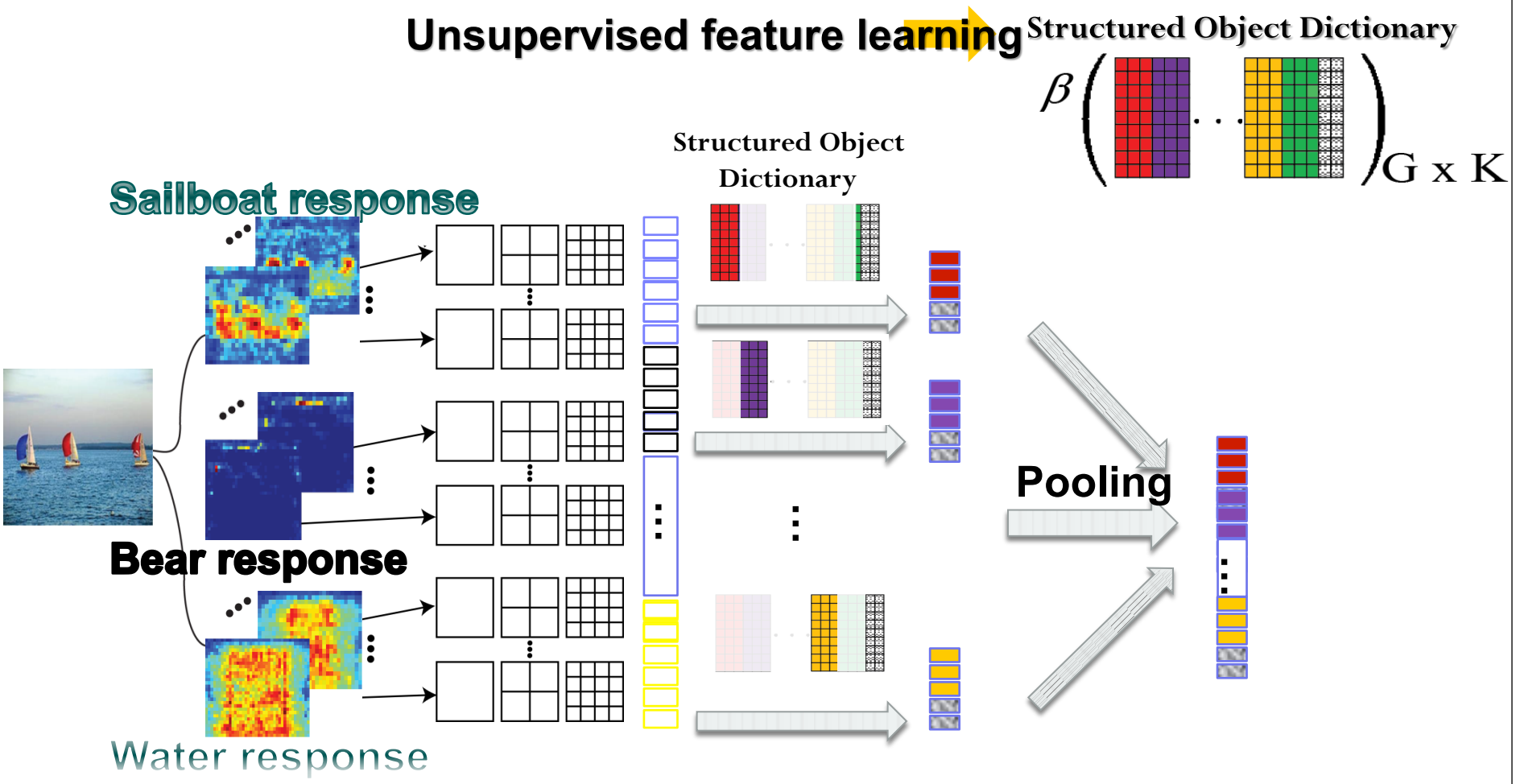
$$\text{where } 2\gamma\beta_{kn}\nu_k^2 + (2\gamma\mu + \beta_{kn}\eta)\nu_k + \mu\eta - w_n\beta_{kn} = 0$$

  - Document code (truncated averaging)

$$\theta_k = \max(0, \bar{s}_k - \frac{\lambda}{2\gamma|I|}) \text{ where } \bar{s}_k = \frac{1}{|I|}\sum_{n\in I} s_{nk}$$

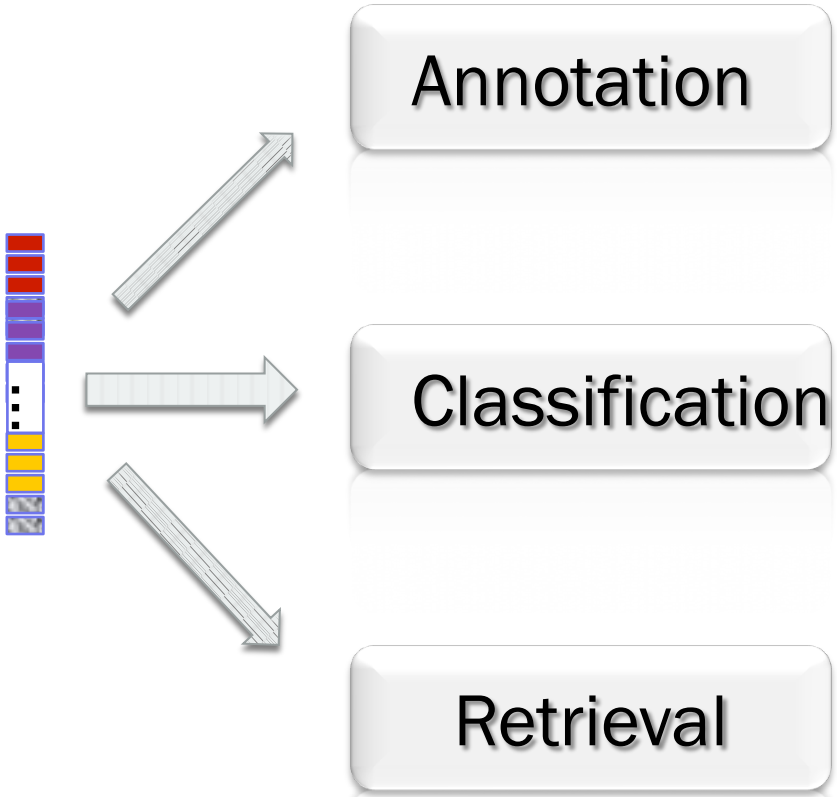
- Dictionary learning
  - projected gradient descent
  - any faster alternative method can be used

Hierarchical Image Coding



L.-J. Li, J. Zhu, H. Su, E.P. Xing, & L. Fei-Fei. Under preparation

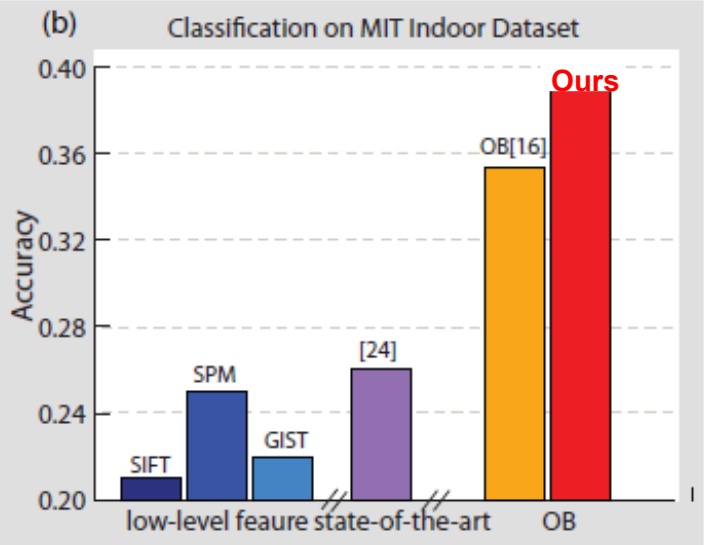
Explore the potential of compact OB in high level recognition tasks



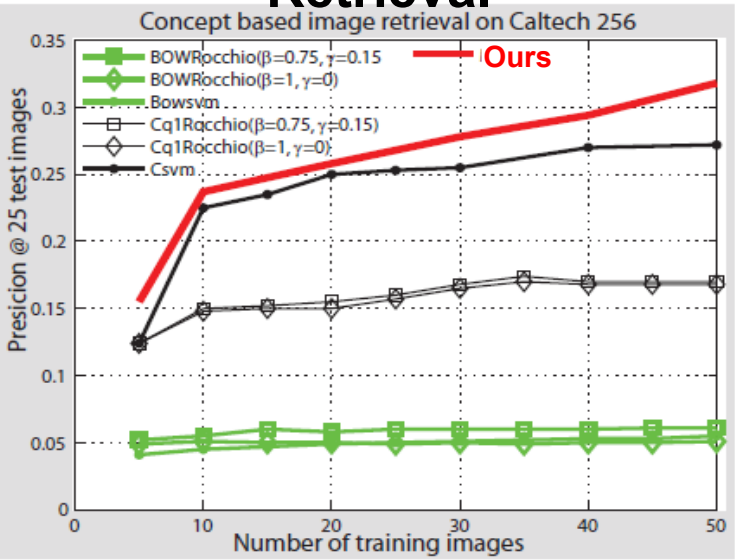
SAILING LAB
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

High level recognition tasks

Classification



Retrieval



Annotation

Human Sky Water Sailboat

Human Sky Ski Snowfield

Human Sky Stuff Battledor

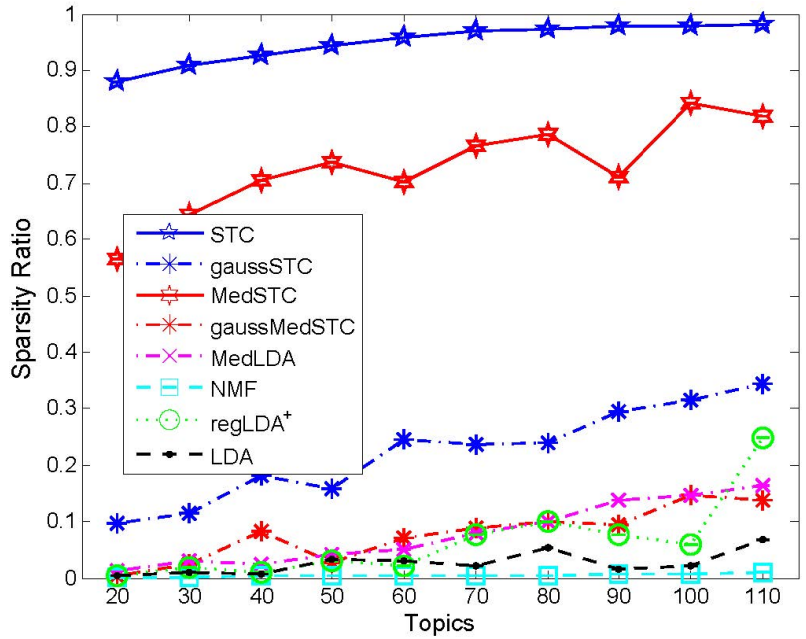
Human Floor Net Battledor

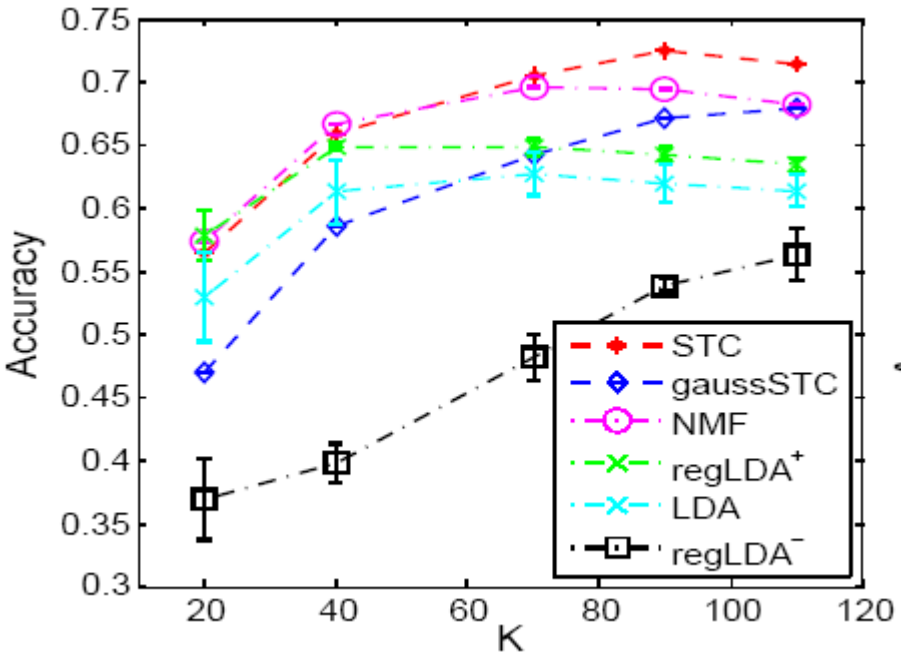
Method	Wang et al 09	OB	Ours
F-measure	38.2%	45.5%	48.3%

~40 times compression

Sparsity

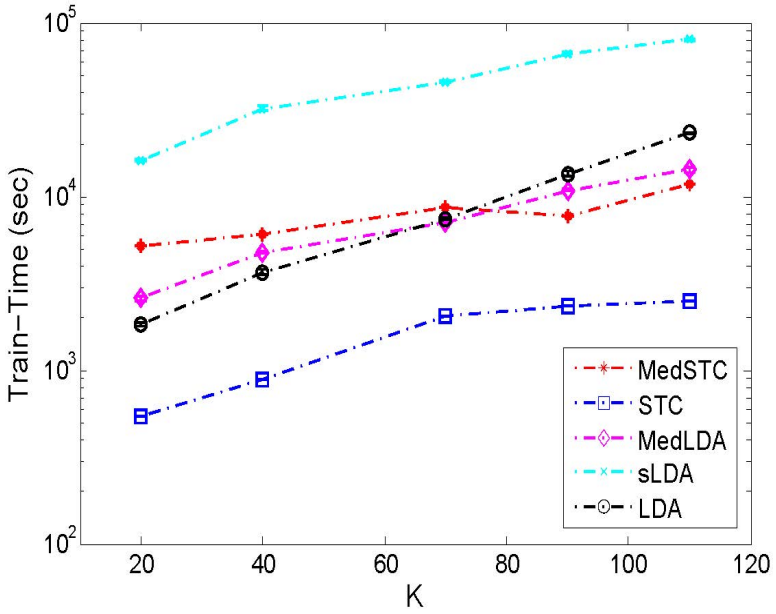
- Probabilistic LDA is ineffective in controlling sparsity by adjusting the Dirichlet parameter
- Sparse topical coding is much more effective than many other methods

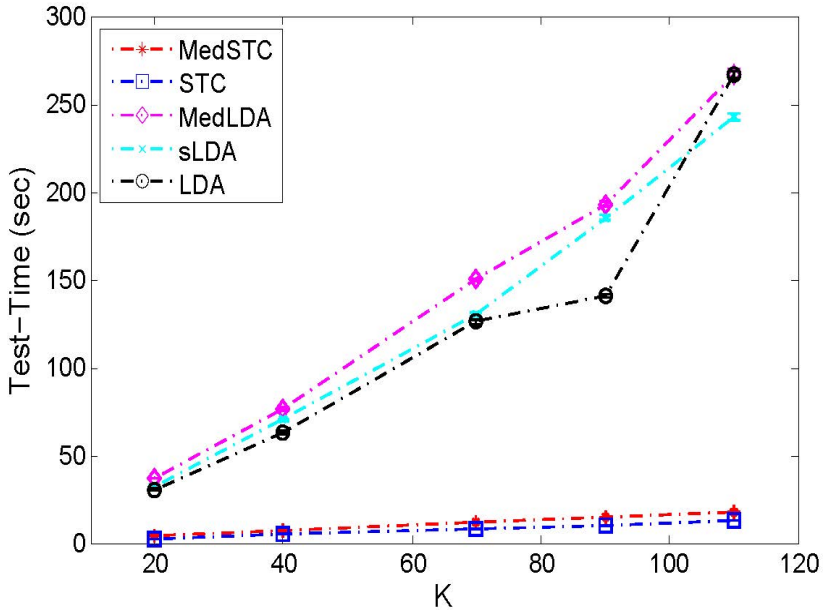




Time Efficiency

- Efficient coordinate algorithm with closed-form update rules for codes
- 1 order of magnitude improvement on training; 2 orders of magnitude improvement on test





Code available at: http://cs.cmu.edu/~junzhu/ctc.htm

Toward Large Scale Problems:

- Large data size:
  - Stochastic methods
  - Parallel computation, e.g., Map-Reduce
- Large feature dimension:
  - Sparsity-inducing regularization
  - Sparse coding
  - Structured sparsity
- Large concept space:
  - Multi-task and transfer learning
  - Structured sparsity

SAILING LAB
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

IMAGENET is a knowledge ontology

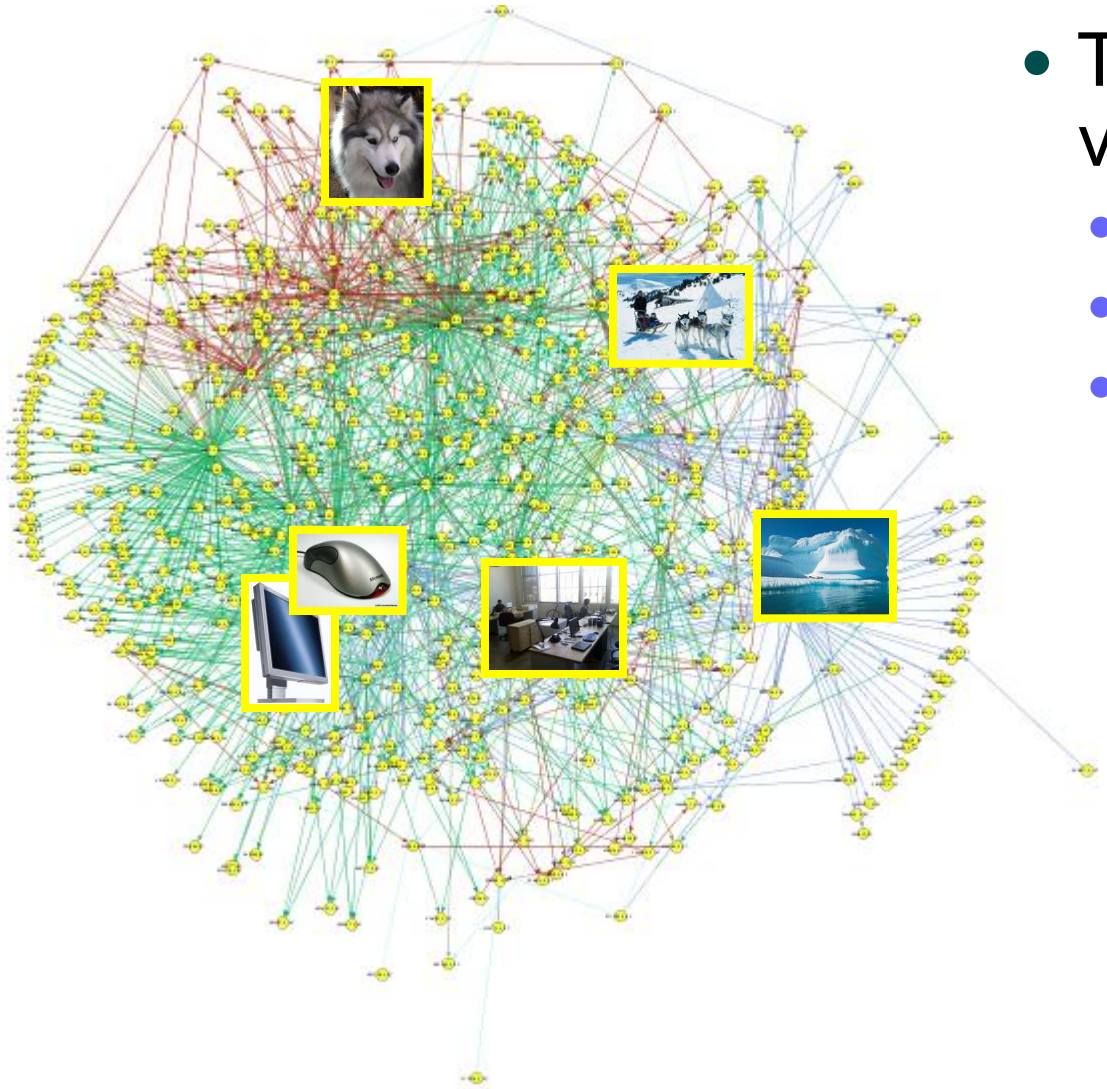
- Taxonomy

- S: (n) Eskimo dog, **husky** (breed of heavy-coated Arctic sled dog)
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) working dog (any of several breeds of usually large powerful dogs bred to work as draft animals and guard and guide dogs)
      - S: (n) dog, domestic dog, Canis familiaris (a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) *"the dog barked all night"*
        - S: (n) canine, canid (any of various fissiped mammals with nonretractile claws and typically long muzzles)
          - S: (n) carnivore (a terrestrial or aquatic flesh-eating mammal) *"terrestrial carnivores have four or five clawed digits on each limb"*
            - S: (n) placental, placental mammal, eutherian, eutherian mammal (mammals having a placenta; all mammals except monotremes and marsupials)
              - S: (n) mammal, mammalian (any warm-blooded vertebrate having the skin more or less covered with hair; young are born alive except for the small subclass of monotremes and nourished with milk)
                - S: (n) vertebrate, craniate (animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium)
                  - S: (n) chordate (any animal of the phylum Chordata having a notochord or spinal column)
                    - S: (n) animal, animate being, beast, brute, creature, fauna (a living organism characterized by voluntary movement)
                      - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
                        - S: (n) living thing, animate thing (a living (or once living) entity)
                          - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                            - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                              - S: (n) physical entity (an entity that has physical existence)
                                - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Courtesy L. Fei-Fei

SAILING LAB
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

IMAGENET is a knowledge ontology

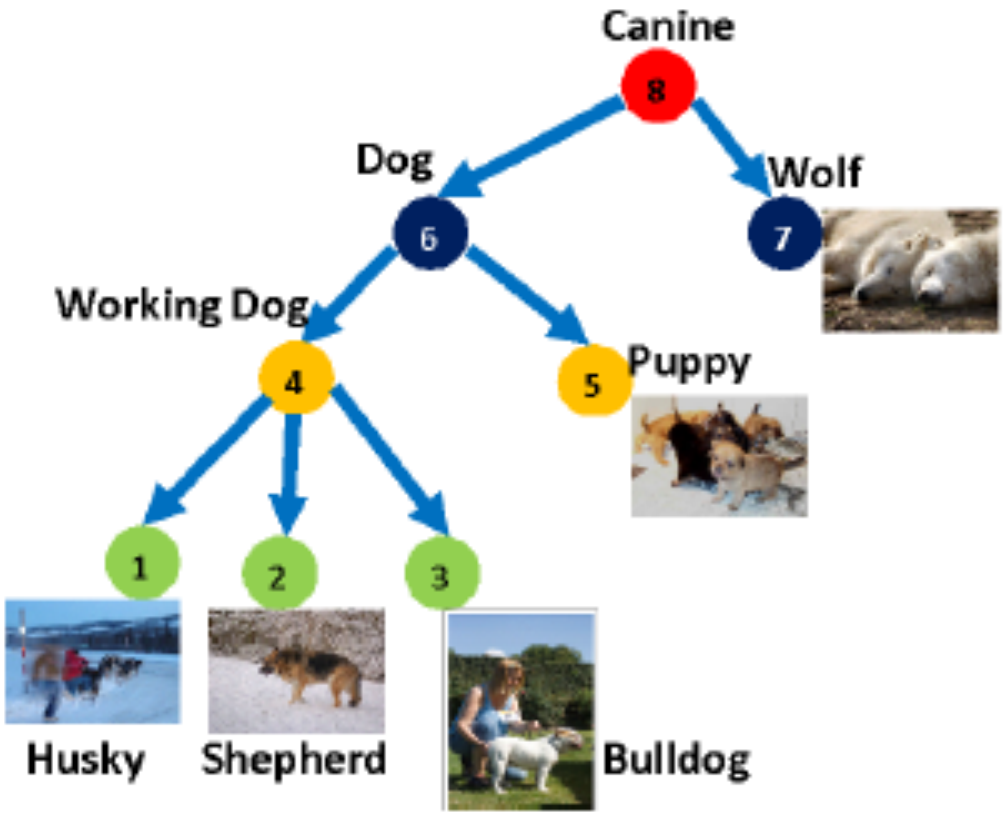


- The "network" of visual concepts
  - Prior knowledge
  - Context
  - Hidden knowledge and structure among visual concepts

Courtesy L. Fei-Fei

Hierarchical Semantic Structure

- Tree hierarchy in ImageNet



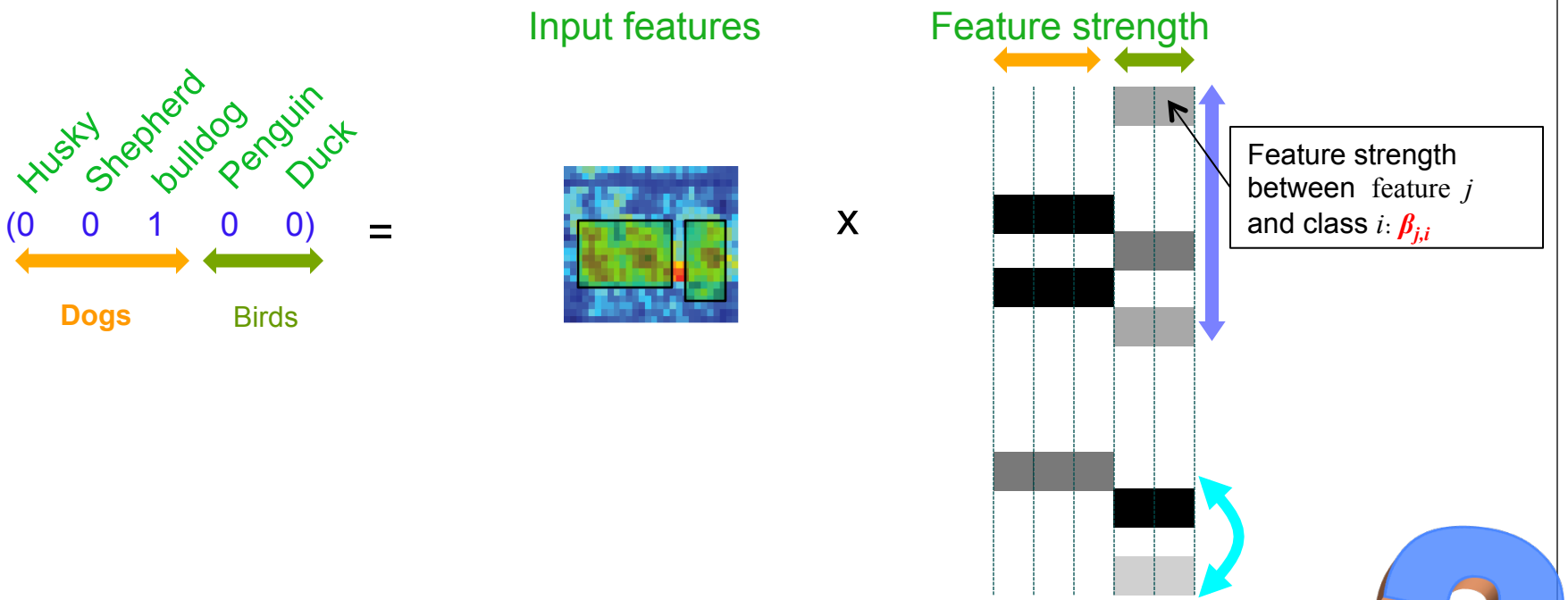
Structured Prediction

- Binary classification: black-and-white decisions

- Multi-class classification: the world of technicolor

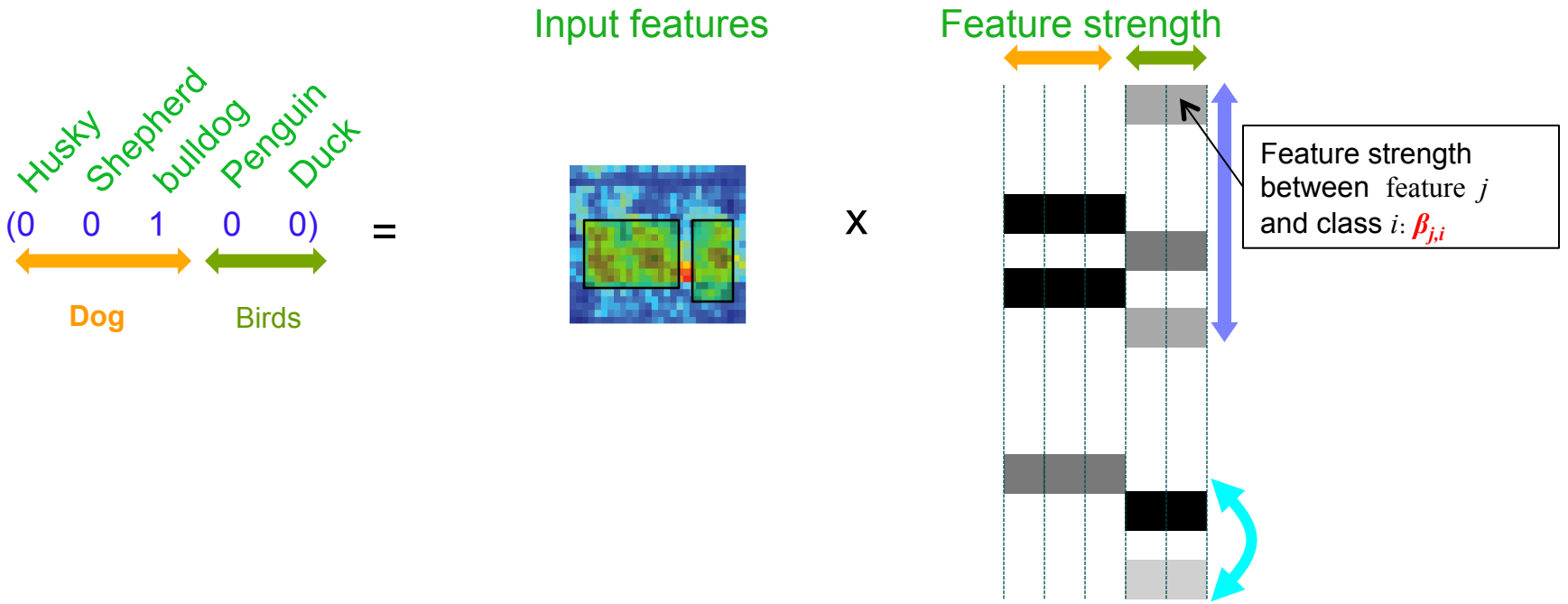
  - can be reduced to several binary decisions, but...
  - often better to handle multiple classes directly
  - how many classes? 2? 5? exponentially many?
- **Structured** prediction: many classes, strongly interdependent
  - Example: sequence labeling (number of classes exponential in the sequence length

Multivariate Regression for Multi-task classification



How to combine information across multiple classes to increase the power?

Multivariate Regression for Multi-task classification



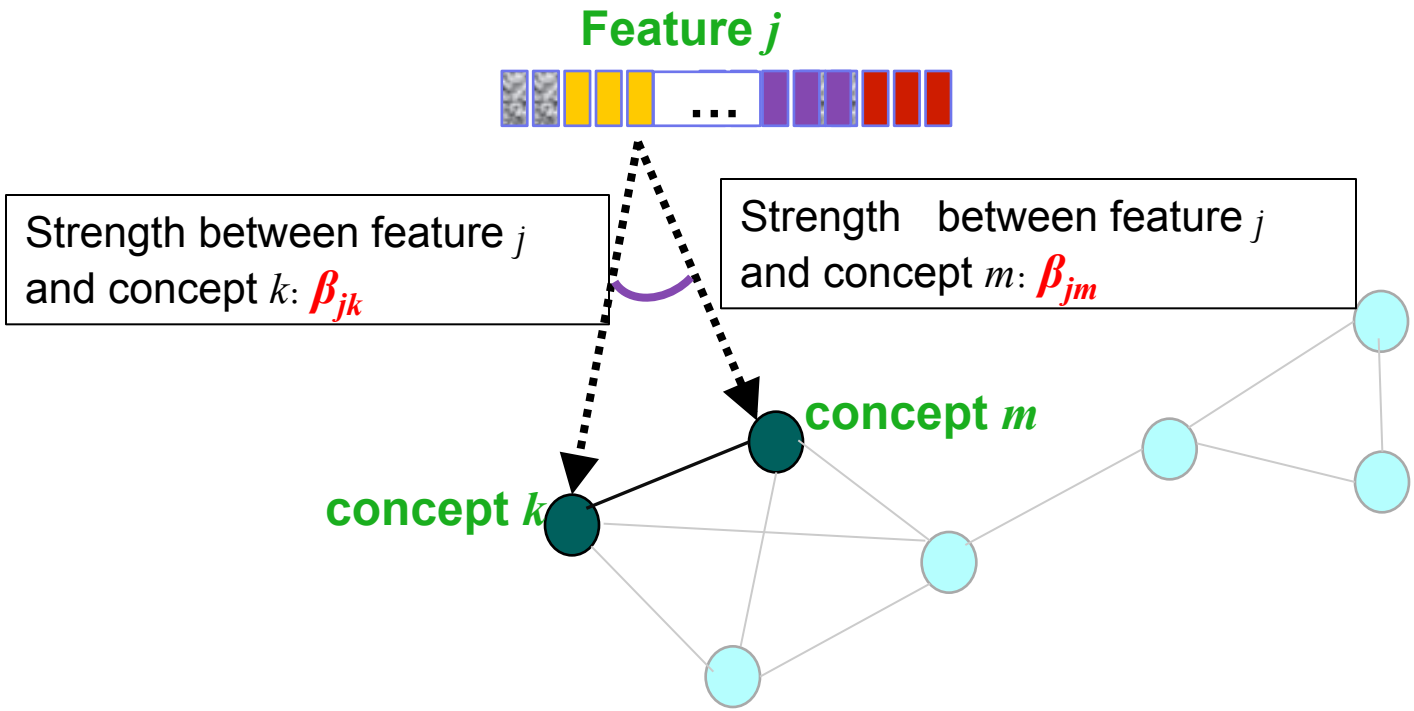
$$\beta^* = \arg\min_{\beta} \sum_{i} (y_i - X_i \beta_i)^T (y_i - X_i \beta_i) + \lambda \sum_{i,j} |\beta_{j,i}|$$

\+

We introduce

Graph- or tree-guided penalty

Graph-Guided Fusion Penalty

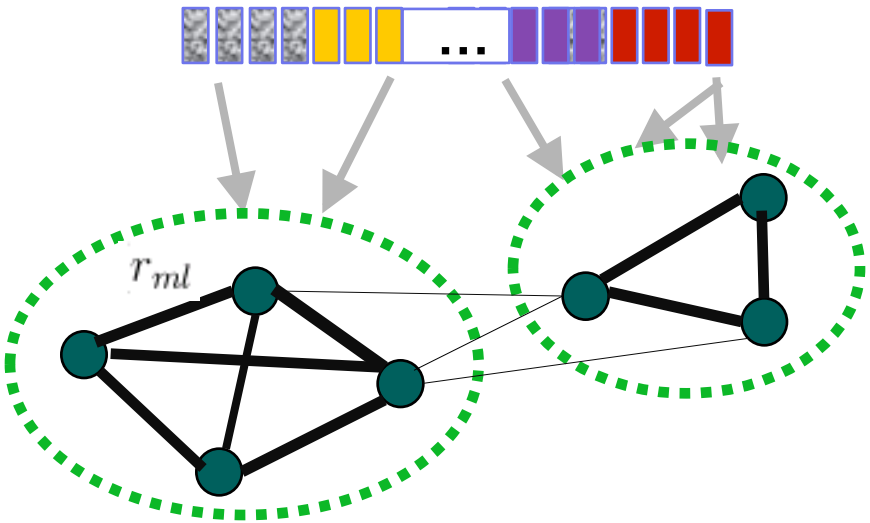


- Fusion Penalty: $|\beta_{jk} - \beta_{jm}|$
- For two correlated concepts (connected in the network), the association strengths may have similar values.
  - Fusion effect propagates to the entire network
  - Association between features and subnetworks of concepts

Kim and Xing, PLOS G 2009

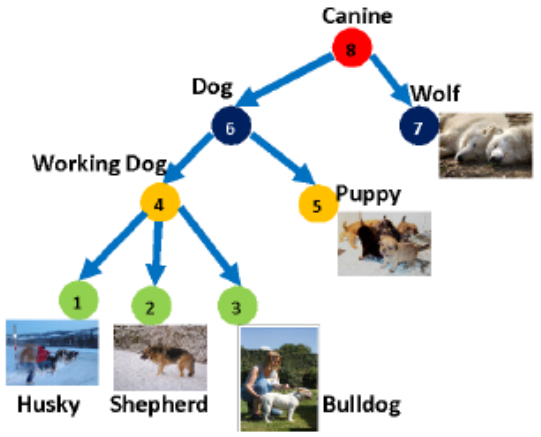
Graph-Weighted Fused Lasso

Overall effect



- Subnetwork structure is embedded as a densely connected nodes with large edge weights
- Edges with small weights are effectively ignored

Tree-guided Group Lasso

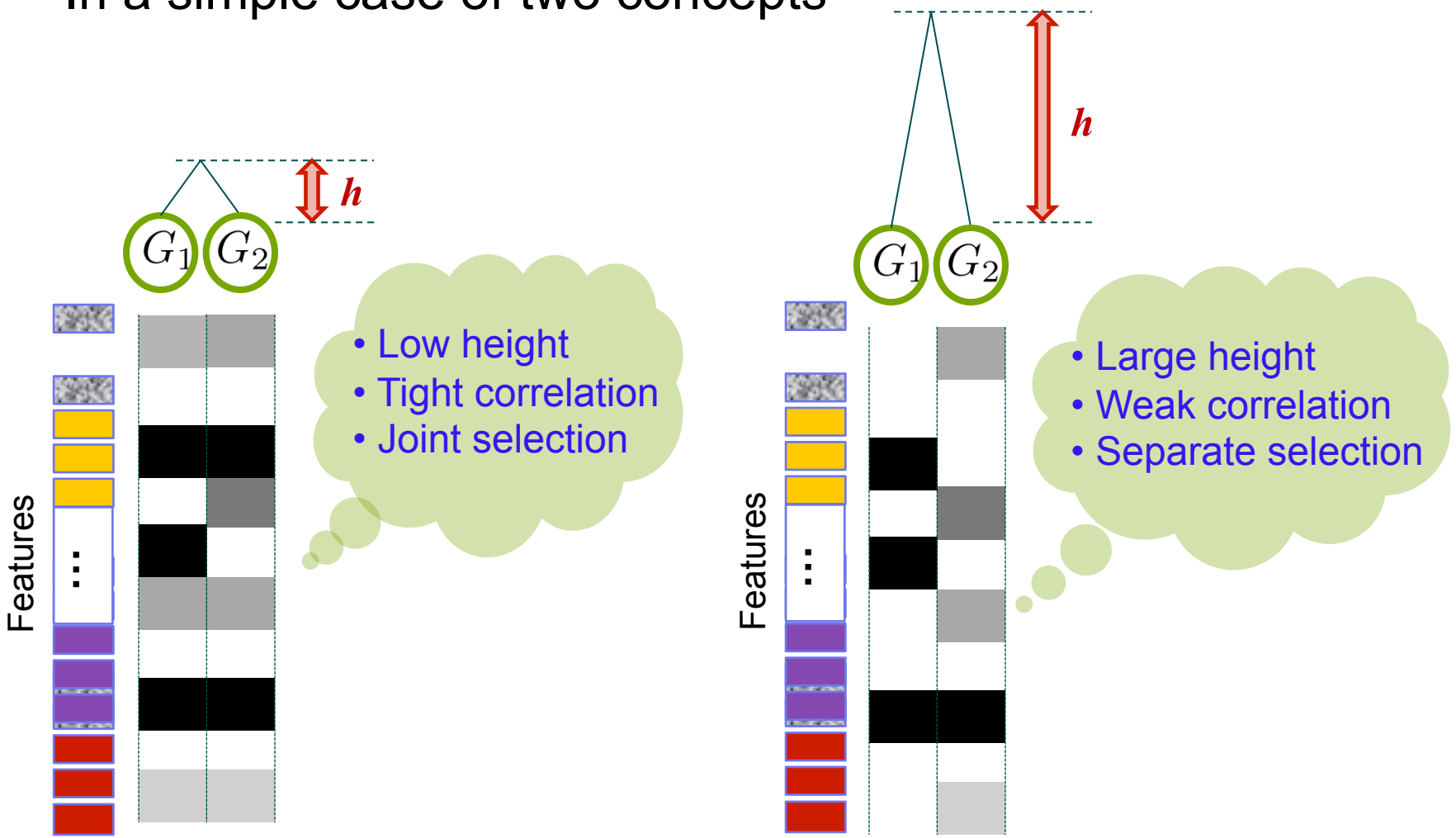


- Why tree?
  - Tree represents a clustering structure
  - Scalability to a very large number of phenotypes
    - Graph : $O(|V|^2)$ edges
    - Tree : $O(|V|)$ edges
  - Capturing Image categories in the ImageNet
    - Agglomerative hierarchical clustering is a popular tool

Kim and Xing, ICML 2010

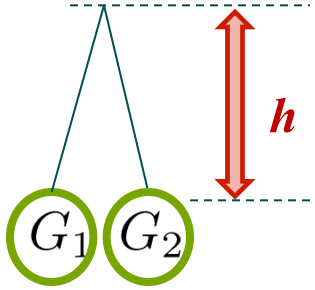
Tree-Guided Group Lasso

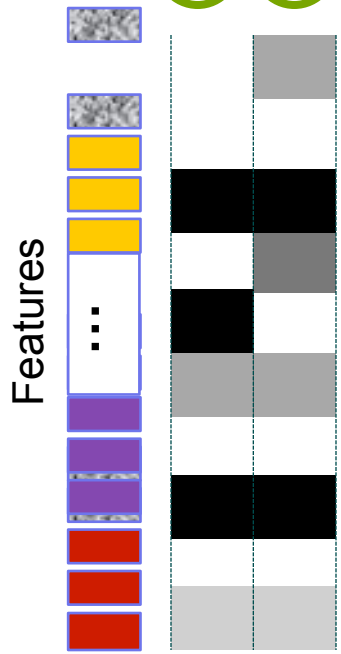
- In a simple case of two concepts



Tree-Guided Group Lasso

- In a simple case of two concepts





$C_1 = \{\beta_{j1}, \beta_{j2}\}$

β_{j1} β_{j2}

Select the child nodes jointly or separately?

Tree-guided group lasso

$$\text{argmin} \ (y - X\beta)' \cdot (y - X\beta) + \lambda \sum_j \left[h\left(|\beta_{j1}| + |\beta_{j2}|\right) + (1-h)\left(\sqrt{\beta_{j1}^2 + \beta_{j2}^2}\right) \right]$$

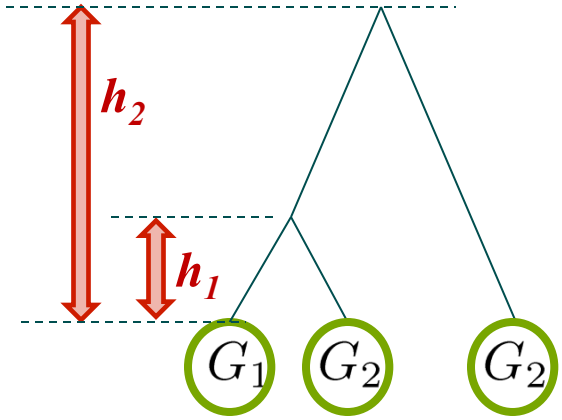
L_1 penalty
- Lasso penalty
- **Separate** selection

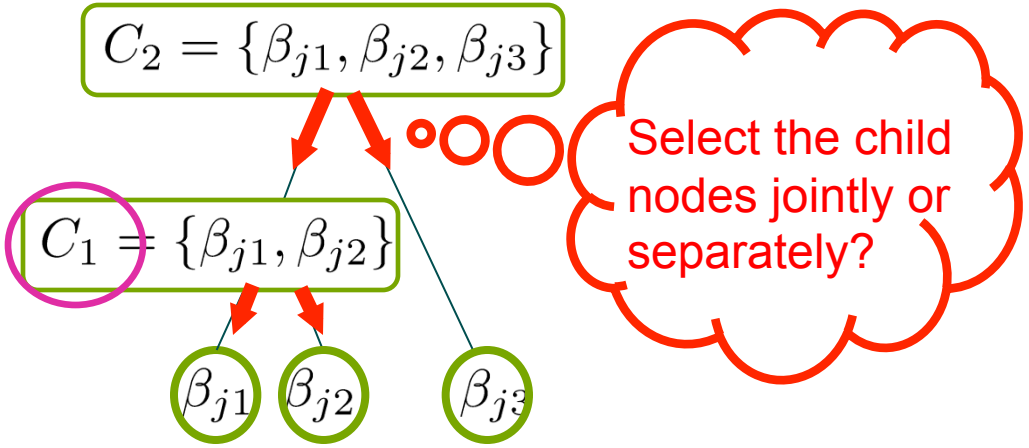
L_2 penalty
- Group lasso
- **Joint** selection

Elastic net

Tree-Guided Group Lasso

- For a general tree





Tree-guided group lasso

$$\text{argmin} \ (y - X\beta)' \cdot (y - X\beta)$$

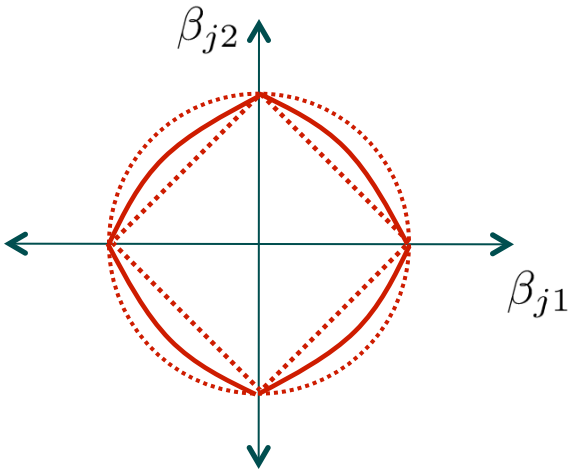
$$+\lambda \sum_j \left[(1-h_2)\left(\sqrt{\beta_{j1}^2 + \beta_{j2}^2 + \beta_{j3}^2}\right) + h_2\left(|C_1| + |\beta_{j3}|\right) \right]$$

$$(1-h_1)\left(\sqrt{\beta_{j1}^2 + \beta_{j2}^2}\right) + h_1\left(|\beta_{j1}| + |\beta_{j2}|\right)$$

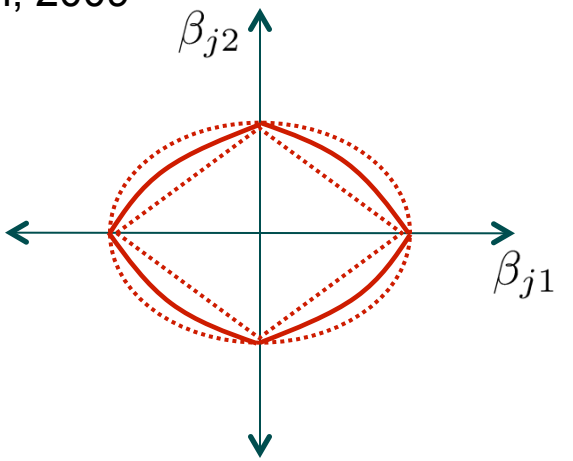
Joint selection **Separate selection**

Proposition 1 *For each of the k-th output (gene), the sum of the weights w_v for all nodes $v \in V$ in T whose group G_v contains the k-th output (gene) as a member equals one. In other words, the following holds:*

$$\sum_{v:k\in G_v} w_v = \prod_{m\in Ancestors(v_k)} h_m + \sum_{l\in Ancestors(v_k)} (1-h_l) \prod_{m\in Ancestors(v_l)} h_m = 1.$$



Previously, in Jenatton, Audibert & Bach, 2009



Exploiting Hierarchical Semantic Structure in ImageNET

- Augmented loss function
  - Weigh differently for different misclassification outcomes
  - Example: classify a “pony” as a “horse” should be penalized less than classifying it as a “car”
- Overlapping group lasso regularization
  - Highly correlated categories should share a common set of features
  - Weakly related categories less likely to be affected by same features

Zhao, Fei-Fei and Xing, in preparation

Augmented Loss Function

- Logistic regression
  - X: *J×N input matrix*
  - *Y: N×1 output vector*
  - *Conditional Likelihood*

$$P(y|\mathbf{x}_i, \mathbf{W}) = \frac{\exp(\mathbf{w}_y^T \mathbf{x}_i)}{\sum_k \exp(\mathbf{w}_k^T \mathbf{x}_i)}$$

$$y^* = \arg\max_{y \in \{1,\ldots,k\}} P(y|\mathbf{x}, \mathbf{W})$$

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} -\sum_{i=1}^{N} \ln P(y|\mathbf{x}_i, \mathbf{W}) + \lambda\Omega(\mathbf{W})$$

Augmented Loss Function

- Semantic relatedness matrix $\mathbf{S} \in \mathbb{R}^{K \times K}$
- Augmented conditional likelihood

$$\hat{P}(y|\mathbf{x}_i, \mathbf{W}) \propto \sum_{r=1}^{K} \mathbf{S}_{y,r} P(r|\mathbf{x}_i, \mathbf{W})$$

$$\hat{P}(y|\mathbf{x}_i, \mathbf{W}) = \frac{\sum_{r=1}^{K} \mathbf{S}_{y,r} \exp(\mathbf{w}_r^T \mathbf{x}_i)}{\sum_{r=1}^{K} \sum_{k=1}^{K} \mathbf{S}_{k,r} \exp(\mathbf{w}_r^T \mathbf{x}_i)}$$

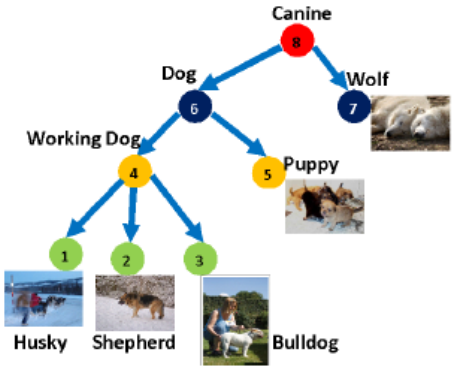
Semantic Relatedness Matrix

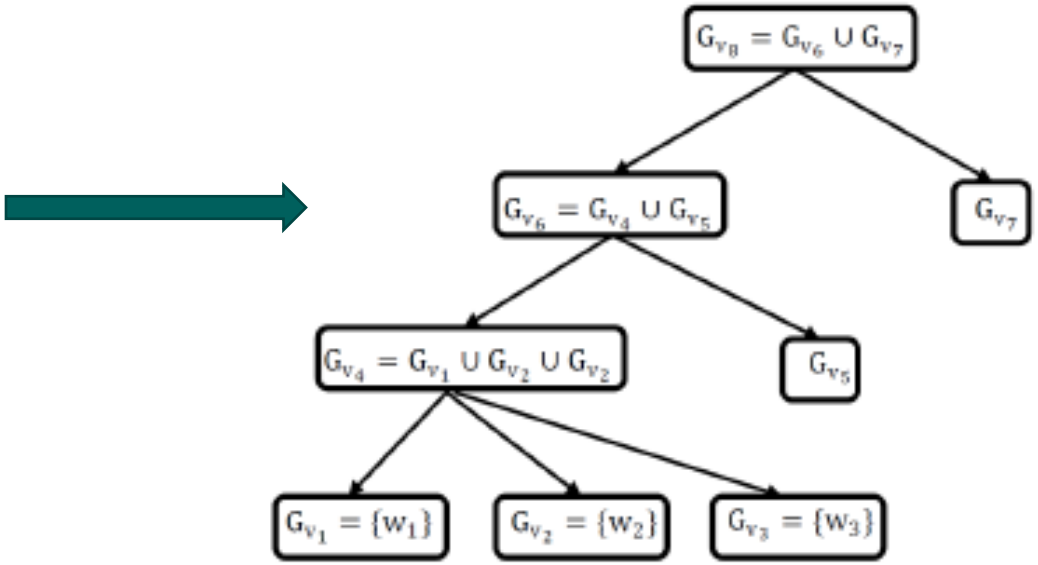
- Semantic distance $\mathbf{D}_{ij}$ between class *i* and class *j*

$$\mathbf{D}_{ij} = \frac{\text{intersect}(\text{path}(i), \text{path}(j))}{\max(\text{length}(\text{path}(i)), \text{length}(\text{path}(j)))}$$

- path(*i*): path from root node to node *I*
- Intersect(*s*,*t*): number of nodes shared by two paths *s* and *t*
- Semantic relatedness matrix S = exp(-k(I-D))

Tree-Guided Sparse Feature Coding





Overlapping-group lasso penalty:

$$\Omega(\mathbf{W}) = \sum_j \sum_{v \in \mathcal{V}} \gamma_v ||\mathbf{w}_{jG_v}||_2$$

Optimization

- Non-smoothness of overlapping-group-lasso penalty
  - Proximal gradient
- Large number of training examples
  - Parallel computation
  - Map-Reduce on computing gradient
    - Map: calculate gradient on single example
    - Reduce: gather gradients computed by all map procedures, and calculate the sum

Proximal Gradient Descent

Original Problem:

$$\arg\min_{\boldsymbol{\beta}\in\mathbb{R}^J} f(\boldsymbol{\beta}) \equiv \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \Omega(\boldsymbol{\beta})$$

$$\Omega(\boldsymbol{\beta}) = \max_{\boldsymbol{\alpha}\in\mathcal{Q}} \boldsymbol{\alpha}^T C\boldsymbol{\beta}$$

Approximation Problem:

$$\arg\min_{\boldsymbol{\beta}\in\mathbb{R}^J} \widetilde{f}(\boldsymbol{\beta}) \equiv \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + f_\mu(\boldsymbol{\beta})$$

$$f_\mu(\boldsymbol{\beta}) = \max_{\boldsymbol{\alpha}\in\mathcal{Q}} \boldsymbol{\alpha}^T C\boldsymbol{\beta} - \mu d(\boldsymbol{\alpha})$$

Gradient of the Approximation:

$$\nabla\widetilde{f}(\boldsymbol{\beta}) = \mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}-\mathbf{y}) + C^T\boldsymbol{\alpha}^*$$

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}\in\mathcal{Q}} \boldsymbol{\alpha}^T C\boldsymbol{\beta} - \mu d(\boldsymbol{\alpha})$$

$\nabla\widetilde{f}(\boldsymbol{\beta})$ is Lipschitz continuous with the Lipschitz constant L

$$L = \lambda_{\max}(\mathbf{X}^T\mathbf{X}) + L_\mu$$

Chen et al and Xing, UAI 2011

Reformulate the Penalty

- Reformulate $||\mathbf{w}_{j\mathbf{g}_i}||_2$ as $||\mathbf{w}_{j\mathbf{g}_i}||_2 = \max_{||\boldsymbol{\alpha}_{j\mathbf{g}_i}||_2 \leq 1} \boldsymbol{\alpha}_{j\mathbf{g}_i}^T \mathbf{w}_{j\mathbf{g}_i}$
- Define $\sum_{\mathbf{g} \in \mathcal{G}} |\mathbf{g}| \times J$ matrix

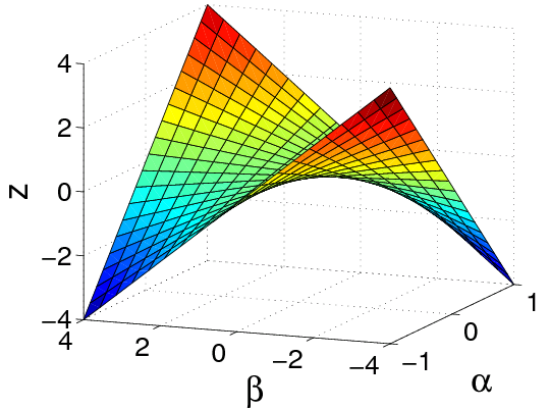
$$\mathbf{A} = \begin{pmatrix} \boldsymbol{\alpha}_{1\mathbf{g}_1} & \cdots & \boldsymbol{\alpha}_{J\mathbf{g}_1} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\alpha}_{1\mathbf{g}_{|\mathcal{G}|}} & \cdots & \boldsymbol{\alpha}_{J\mathbf{g}_{|\mathcal{G}|}} \end{pmatrix}$$

- Overlapping-group-lasso penalty reformulated as

$$\Omega(\mathbf{W}) = \sum_j \sum_i \gamma_i \max_{||\boldsymbol{\alpha}_{j\mathbf{g}_i}||_2 \leq 1} \boldsymbol{\alpha}_{j\mathbf{g}_i}^T \mathbf{w}_{j\mathbf{g}_i} = \max_{A \in \mathcal{O}} \langle \mathbf{C}\mathbf{W}^T, \mathbf{A} \rangle$$

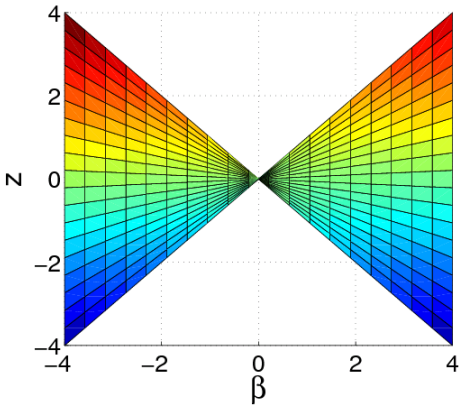
Geometric Interpretation

- Smooth approximation



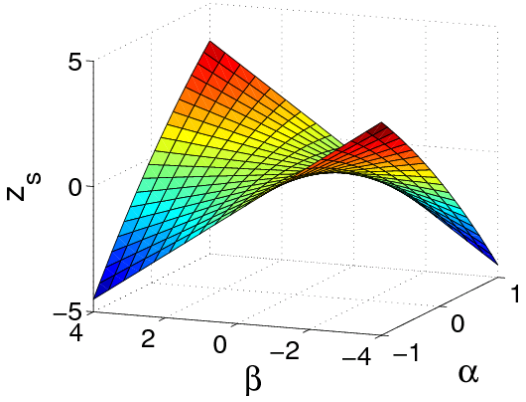
$z(\alpha, \beta) = \alpha\beta$

Projection onto $z - \beta$ Plane



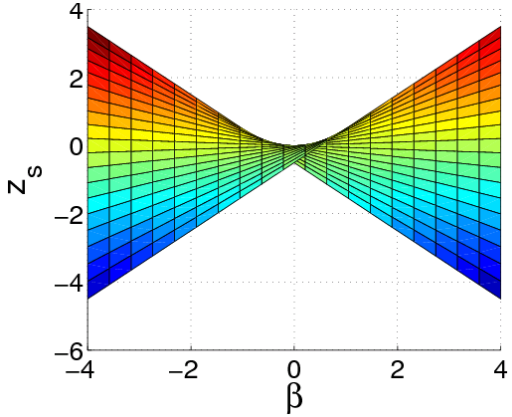
$f_0(\beta) = \max_{\alpha \in [-1,1]} z(\alpha, \beta) = |\beta|$

Uppermost Line Nonsmooth



$z_s(\alpha, \beta) = \alpha\beta - \frac{1}{2}\alpha^2$

Projection onto $z_s - \beta$ Plane



$f_1(\beta) = \max_{\alpha \in [-1,1]} z_s(\alpha, \beta)$

Uppermost Line Smooth

Proximal Gradient

- Introduce auxiliary function to construct smooth approximation

$$f_\mu(\mathbf{W}) = \max_{\mathbf{A}\in\mathcal{O}} \langle \mathbf{C}\mathbf{W}^T, \mathbf{A}\rangle - \mu d(\mathbf{A}) \qquad (13)$$

Theorem 1 *For any $\mu > 0$, $f_\mu(\mathbf{W})$ is a convex and continuously differentiable function in $\mathbf{W}$, and the gradient of $f_\mu(\mathbf{W})$ takes the following form*

$$\nabla f_\mu(\mathbf{W}) = \mathbf{A}^{*T}\mathbf{C}$$

where A^* is the optimal solution to (13). Moreover, the gradient $\nabla f_\mu(\mathbf{W})$ is Lipschitz continuous with the Lipschitz constant $L_\mu = \frac{||\mathbf{C}||^2}{\mu}$, where $||\mathbf{C}||$ is a special norm defined as $||\mathbf{C}|| = \max_{||\mathbf{V}||_F \le 1} ||\mathbf{V}\mathbf{C}||_F$.

Convergence Rate

Theorem: If we require $f(\boldsymbol{\beta}^t) - f(\boldsymbol{\beta}^*) \leq \epsilon$ and set $\mu = \frac{\epsilon}{2D}$, the number of iterations is upper bounded by:

$$t \leq \sqrt{\frac{4\|\boldsymbol{\beta}^*\|_2^2}{\epsilon}\left(\lambda_{\max}(\mathbf{X}^T\mathbf{X}) + \frac{2D\|\Gamma\|^2}{\epsilon}\right)} = O(\frac{1}{\epsilon})$$

Remarks: state of the art IPM method for for SOCP converges at a rate $O(\frac{1}{\epsilon^2})$

Multi-Task Time Complexity

- Pre-compute:

$$\mathbf{X}^T\mathbf{X},\ \mathbf{X}^T\mathbf{Y}\text{:}\quad O(J^2N + JKN)$$

- Per-iteration Complexity (computing gradient)

Tree:

IPM for SOCP	$O\left(J^2(K+\lvert\mathcal{G}\rvert)^2(KN + J(\sum_{g\in\mathcal{G}}\lvert g\rvert))\right)$
Proximal-Gradient	$O(J^2K + J\sum_{g\in\mathcal{G}}\lvert g\rvert)$

Graph:

IPM for SOCP	$O\left(J^2(K+\lvert E\rvert)^2(KN + JK + J\lvert E\rvert)\right)$
Proximal-Gradient	$O(J^2K + J\lvert E\rvert)$

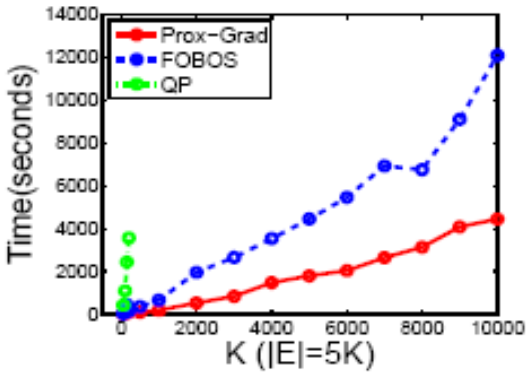
Proximal-Gradient: Independent of Sample Size

Linear in #.of concepts

Parallelizable

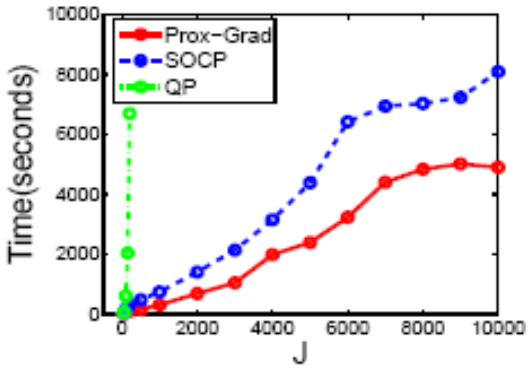
Experiments

- Time complexity



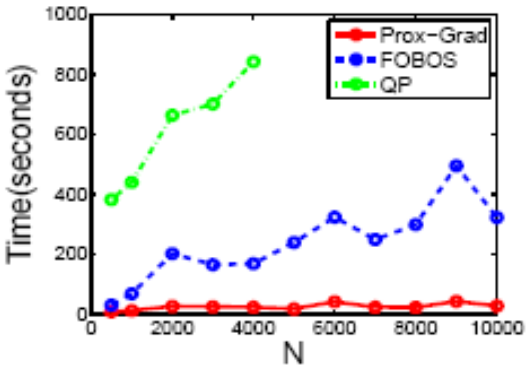
(a)

$N = 500, J = 100$



(b)

$N = 1000, K = 50$



(c)

$J = 100, K = 50$

$\mu = 10^{-4}, \rho = 0.5$

Empirical Study

- ILSVRC10: 1.2 million images / 1000 categories
- 1000 visual words in dictionary
- Locality-constrained linear coding
- Max pooling on spatial pyramid

- Each image represented as a vector in 21000 dimensional space

Zhao, Fei-Fei and Xing, in preparation

Classification Results

- Flat error & hierarchical error

Table 1: Classification results (both flat and hierarchical errors) of various algorithms.

	Flat Error					Hierarchical Error				
Algorithm	Top 1	Top 2	Top 3	Top 4	Top 5	Top 1	Top 2	Top 3	Top 4	Top 5
LR	0.797	0.726	0.678	0.639	0.607	8.727	6.974	5.997	5.355	4.854
ALR	0.796	0.723	0.668	0.624	0.587	8.259	6.234	5.061	4.269	3.659
GroupLR	0.786	0.699	0.642	0.600	0.568	7.620	5.460	4.322	3.624	3.156
APPLET	0.779	0.698	0.634	0.589	0.565	7.208	4.985	3.798	3.166	3.012

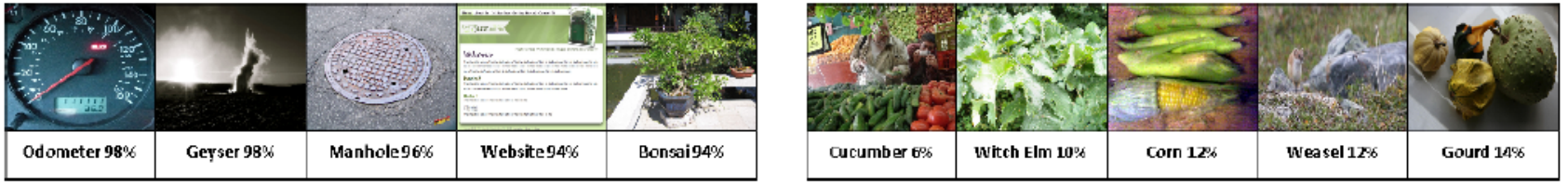


Figure 2: Left: image classes with highest accuracy. Right: image classes with lowest accuracy.

Effects of Augmented Loss Function

- APPLET vs. LR

True class	laptop	linden	gordon setter	gourd	bullfrog	volcano	odometer	earthworm
APPLET	laptop(0)	live oak(3)	Irish setter(2)	acorn(2)	woodfrog(2)	volcano(0)	odometer(0)	earthworm(0)
LR	laptop(0)	log wood(3)	alp(11)	olive(2)	water snake(9)	geyser(4)	odometer(0)	slug(8)

Table 2: Example prediction results of *APPLET* and *LR*.

- Classification results of APPLET significantly more informative

Summary: why care about structure?

- Theoretically, it increase the power [Mladen and Xing, 2010]

$$\mathbb{P}[\mathcal{M}_* \subseteq \mathcal{M}^{(m^*_{\max})}] \geq 1 - C_1 \exp\left(-C_2 \frac{n^{1-6\delta_s-6\delta_{\min}}}{\max\{\log(p), \log(T)\}}\right) ;$$

Summary: Toward Large Scale Problems:

- Large data size:
  - Stochastic methods
  - Parallel computation, e.g., Map-Reduce
- Large feature dimension:
  - Sparsity-inducing regularization
  - Sparse coding
  - Structured sparsity
- Large concept space:
  - Multi-task and transfer learning
  - Structured sparsity

SAILING LAB
Laboratory for Statistical Artificial InteLligence & INtegrative Genomics

Acknowledgement

SAILING Lab, CMU

Bin Zhao
Jun Zhu
Xi Chen
Seyoung Kim Ph.D.
(now, Assi Prof , SCS@CMU)

Vision Lab, Stanford Univ.

Fei-Fei Li PI
Jia Li
Hao Su