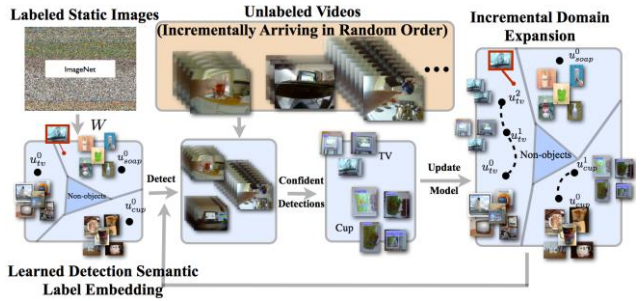
Structured Sparsity

Sung Ju Hwang

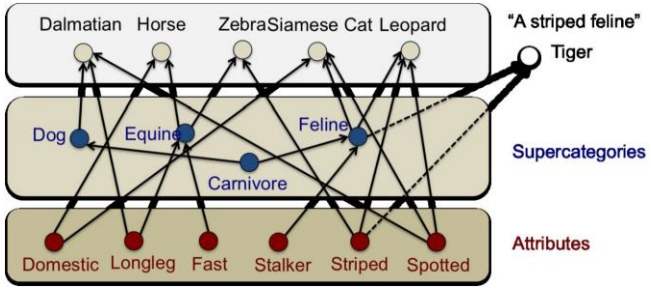
Ulsan National Institute of Science and Technology

Lecturer Introduction

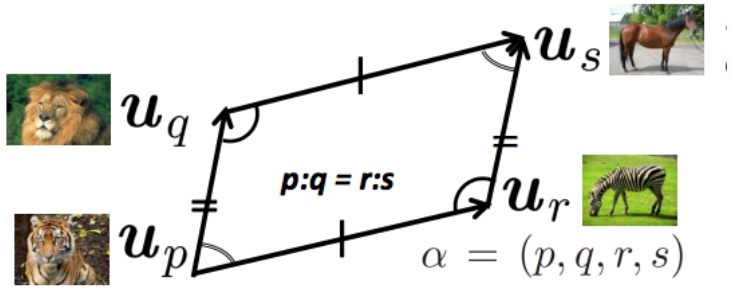
Works on the intersection between Computer Vision and Machine Learning



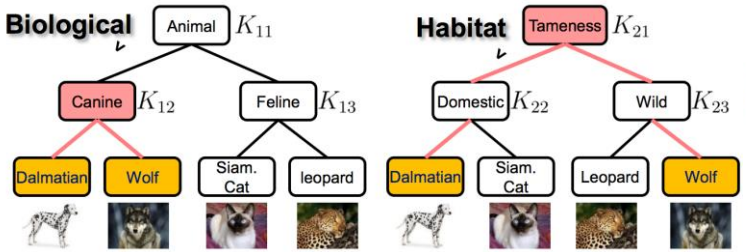
Expanding Object Detector's Horizon: Incremental Learning Framework for Object Detection in Videos, **CVPR 2015**



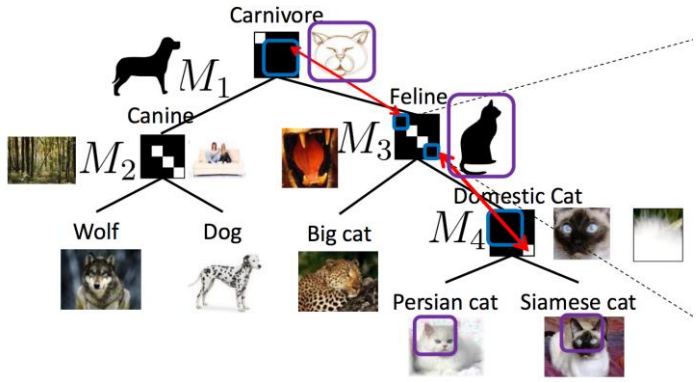
A Unified Semantic Embedding: Relating Taxonomies with Attributes, **NIPS 2014**



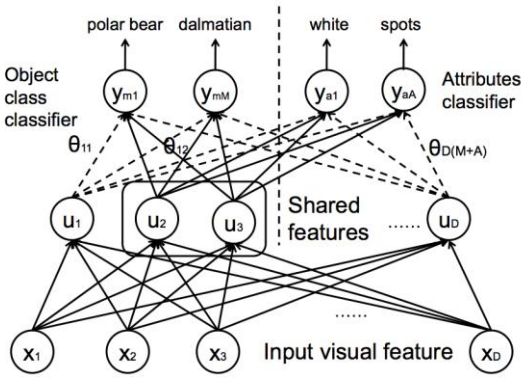
Analogy Preserving Semantic Embedding for Visual Object Categorization, **ICML 2013**



Semantic Kernel Forests from Multiple Taxonomies,**NIPS 2012**



Learning a Tree of Metrics with Disjoint Visual Features, **NIPS 2011**



Sharing Features between Objects and Their Attributes,**CVPR 2011**

Lecture Note Update

http://sjhwang.unist.ac.kr/sparsity.pdf

Current Research Interests

Learning Semantics

- How can we learn high-level semantic knowledge from the given visual and textual data, such that machine's understanding of the world aligns well with how we understand the world?

Interactive Learning

- How can we build a learning model such that machines learn from interacting with humans?

Lifelong Learning

- How can we learn a model that can basically learn forever, while transferring knowledge obtained at earlier stages to the learning for later ones?

Part 1: General Sparsity

- Background: Optimization
- Regularizations
- Sparsity
- Sparsity-inducing Regularization
- Example: lasso
- Numerical optimization – proximal methods
- Alternative sparse methods
- Example: Image restoration
- Example: Self-taught learning

Background: Mathematical Optimization

Selection of the values that minimize/maximize a given function

$$\text{minimize } f_0(w)$$
$$\text{subject to } f_i(w) \leq b_i, i = 1, \ldots, m$$

$$\min_{\mathrm{w}} \quad f_0(w)$$
$$\text{s.t. } f_i(w) \leq b_i, i = 1, \ldots, m$$

$w = (w_1, \ldots, w_n)$ optimization variables

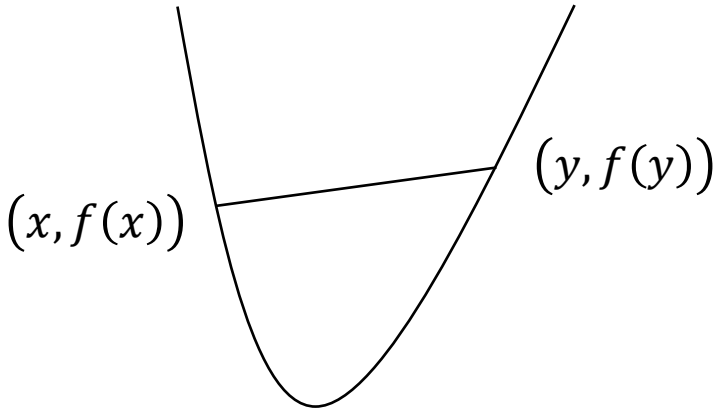
$f_0 : R^n \rightarrow R$ objective function

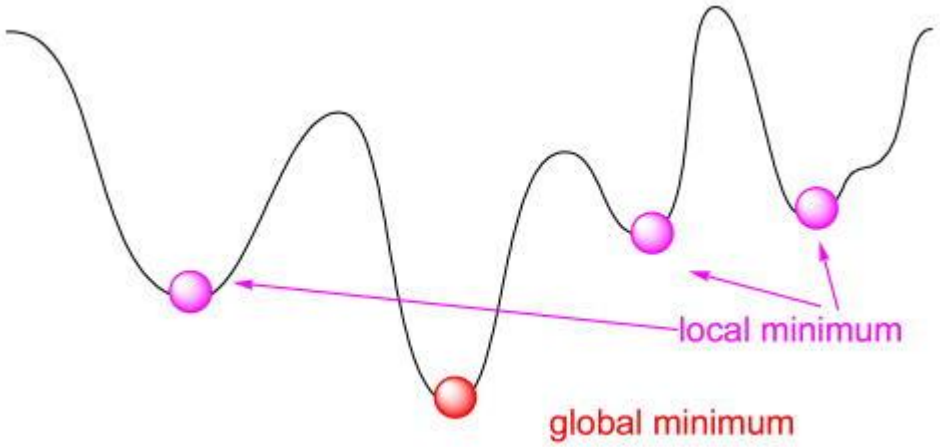
$f_i : R^n \rightarrow R, i = 1, \ldots, m$ constraints function

We want to find the optimal solution w^* that has the smallest value of f_0 among all vectors that satisfy the constraints.

Background: Convex Functions

$f: R^n \to R$ is convex if dom f is a convex set and
$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$





Has a unique global minimum.

f is concave if $-f$ is convex

Regularization

Introduction of bias to better condition the target problem with additional information

$$\min_{w} f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w})$$

feature label regularization parameter

$$\min_{w} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \lambda\Omega(\boldsymbol{w})$$

parameter, weight

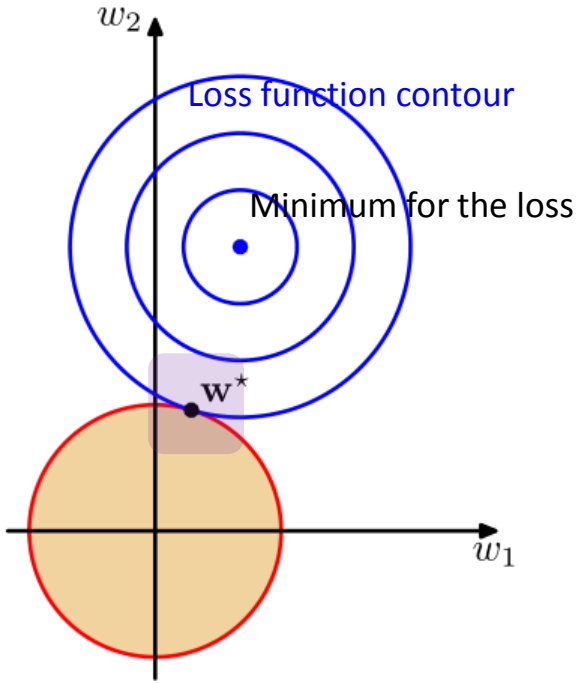
For machine learning, regularization is often used to prevent overfitting.

Regularization

L2 regularizations, based on 2 norm, is one the most common types of regularizations

$$\min_{w} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2$$

$$\|\boldsymbol{w}\|_2 = \sqrt{\sum_{w_j} w_j^2}$$

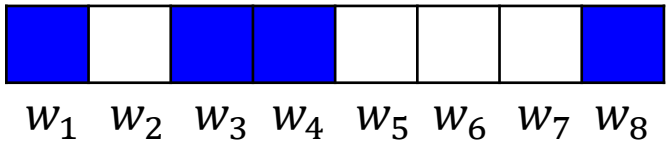


Shrinks the value of the variables while favoring similar weights among them

Sparsity

We define the L0-psuedonorm as follows:

$$\|\boldsymbol{w}\|_0 = \#\{i = 1, \dots, p \mid w_i \neq 0\}$$

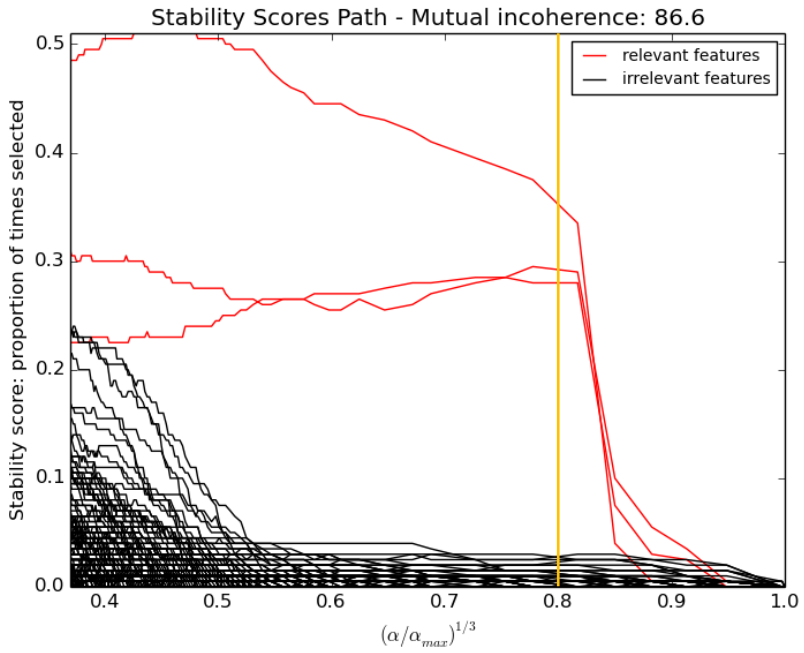


This is a measure of how many of the variables are non-zero.

L0-regularization results in learning **sparse** models, by selecting variables

Why is Sparsity Good?

Feature selection - Identifies features that are truly relevant to the target task.



Useful for high-dimensional learning and data-driven methods

Why is Sparsity Good?

Better Interpretability – The learned model can be better explained in terms of selected non-zero entries.

Category	Ground-truth attributes	Supercategory + learned attributes
 Otter	An animal that swims, fish, water, new world, small, flippers, furry, black, brown, tail, . . .	A musteline mammal that is quadrapedal, flippers, furry, ocean
 Skunk	An animal that is smelly, black, stripes, white, tail, furry, ground, quadrapedal, new world, walks, . . .	A musteline mammal that has stripes
 Deer	An animal that is brown, fast, horns, grazer, forest, quadrapedal, vegetation, timid, hooves, walks, . . .	A deer that has spots, nestspot, longneck, yellow, hooves
 Moose	An animal that has horns, brown, big, quadrapedal, new world, vegetation, grazer, hooves, strong, ground,. . .	A deer that is arctic, stripes, black
Equine	N/A	An odd-toed ungulate, that is lean and active
Primate	N/A	An animal, that has hands and bipedal

[Hwang et al.] S. J. Hwang and L. Sigal, A Unified Semantic Embedding: Relating Taxonomies with Attributes, NIPS 2014

Why is Sparsity Good?

Model compression – With most of the parameters set to zero, sparsity can greatly reduce the memory and computational requirements.

$$\begin{pmatrix} 1.0 & 0 & 5.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.0 & 0 & 0 & 0 & 0 & 11.0 & 0 \\ 0 & 0 & 0 & 0 & 9.0 & 0 & 0 & 0 \\ 0 & 0 & 6.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.0 & 0 & 0 & 0 & 0 \\ 2.0 & 0 & 0 & 0 & 0 & 10.0 & 0 & 0 \\ 0 & 0 & 0 & 8.0 & 0 & 0 & 0 & 0 \\ 0 & 4.0 & 0 & 0 & 0 & 0 & 0 & 12.0 \end{pmatrix}$$

8x8 matrix – requires 512 bytes to store in double precision

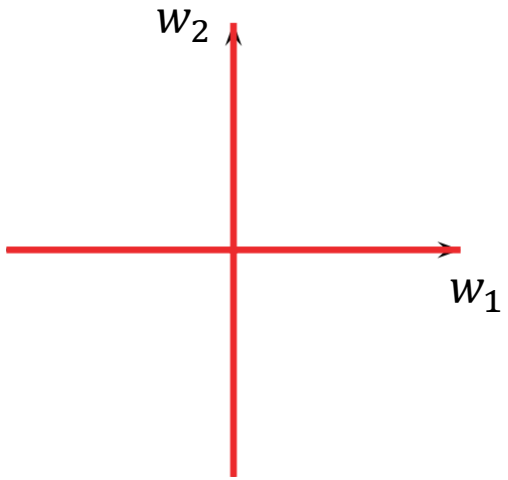
12 nonzero entries – requires 96 bytes + additional memory to store indices.

L0-regularization

Directly solving for L0-regularization is difficult as it should try all possible subsets of variables.

$$\min_{w} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_0$$

$$\|\boldsymbol{w}\|_0 = \#\{i = 1, \dots, p \mid w_i \neq 0\}$$



Are there more efficient ways to obtain sparse models?

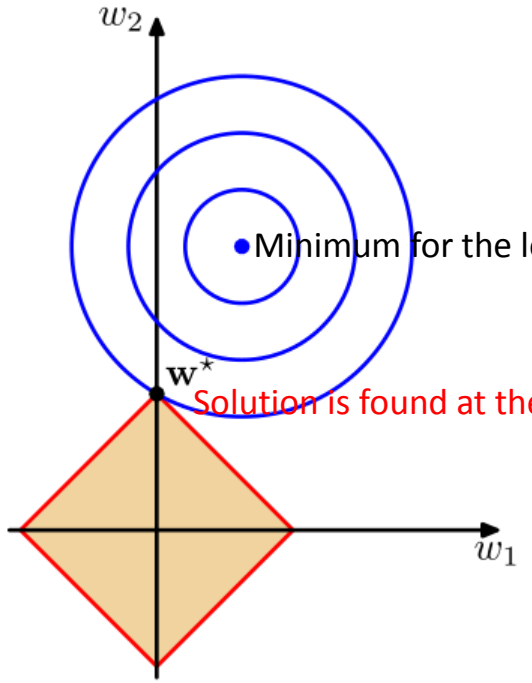
L1 regularization

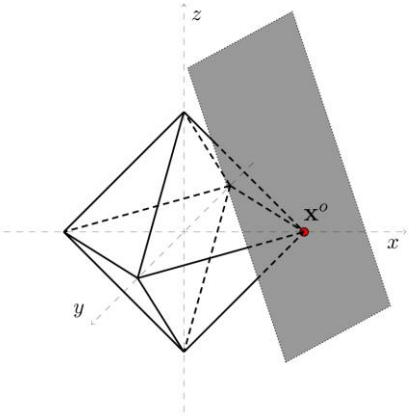
Convex relaxation of L0 regularization.

$$\min_{w} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1$$

$$\|\boldsymbol{w}\|_1 = \sum_{w_j} |w_j|$$

$$\|\boldsymbol{w}\|_1 \leq \lambda$$





As the dimensionality of w increases, the norm ball will have increasingly more number of corners.

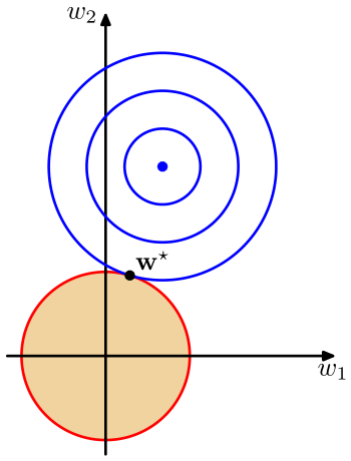
Results in parameter selection

L2- vs L1-regularization

L2 regularization promotes **grouping** – results in equal weights for correlated features

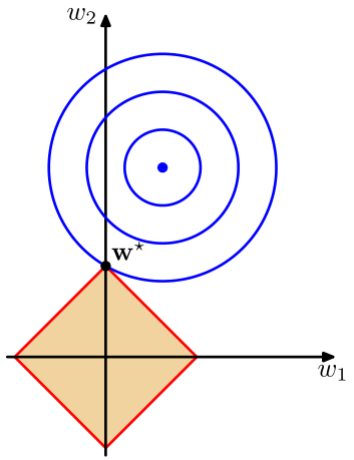
L1 regularization promotes **sparsity** – selects few informative features

L2-regularization



$$\Omega(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

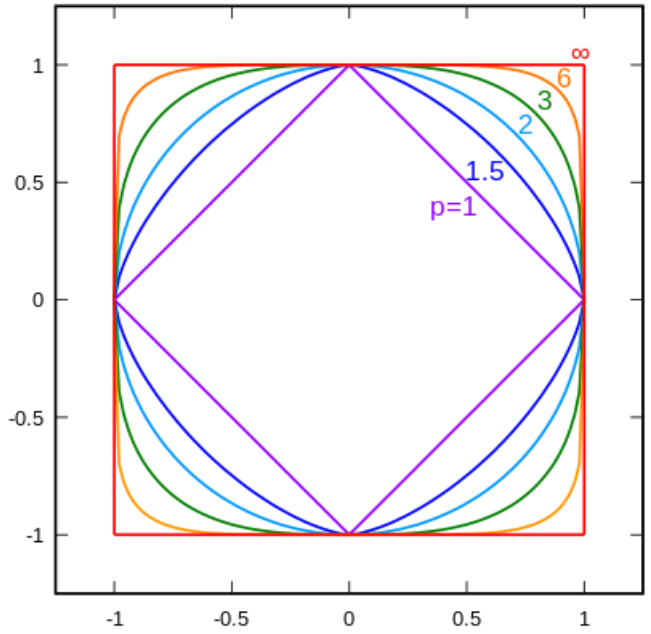
L1-regularization



$$\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$$

Lp-Norms

General case $\|\boldsymbol{w}\|_{\mathrm{p}}$ where p can be any non-negative number



Lp norms with p < 2 promotes sparsity

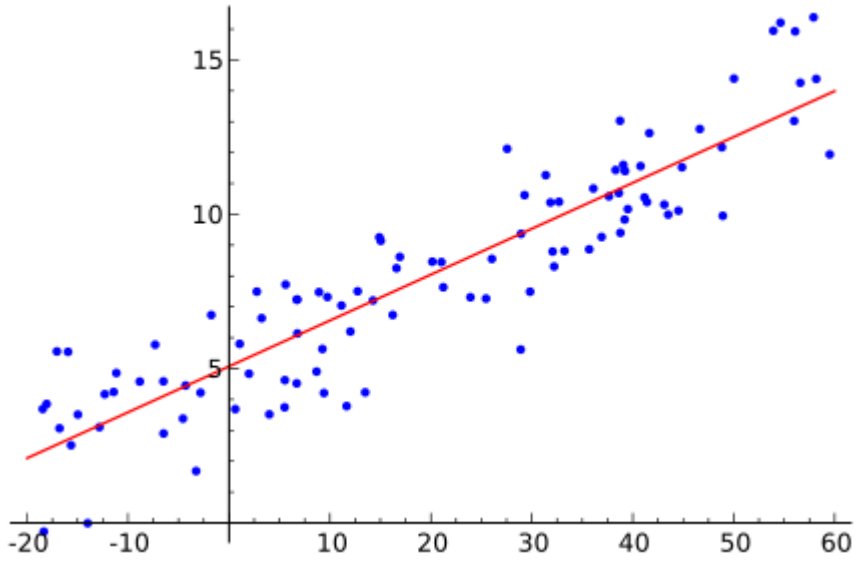
Lp norms with p > 2 promotes grouping

Example: Linear Regression

Fit a linear model $\boldsymbol{w}$, that minimizes the residual between the observed and predicted values

$$\min_{w} \frac{1}{N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T y$$



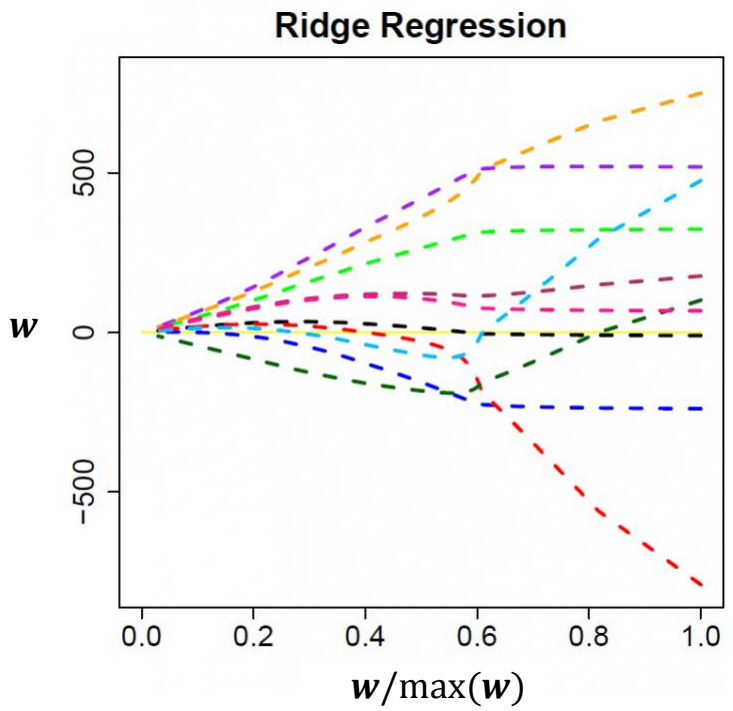
Ridge Regression

Linear regression with L2-regularization

$$\min_w \frac{1}{N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\Omega(\boldsymbol{w})$$

$$\Omega(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T y$$



Shrinks all variables to zero – reduces variance while introducing bias.

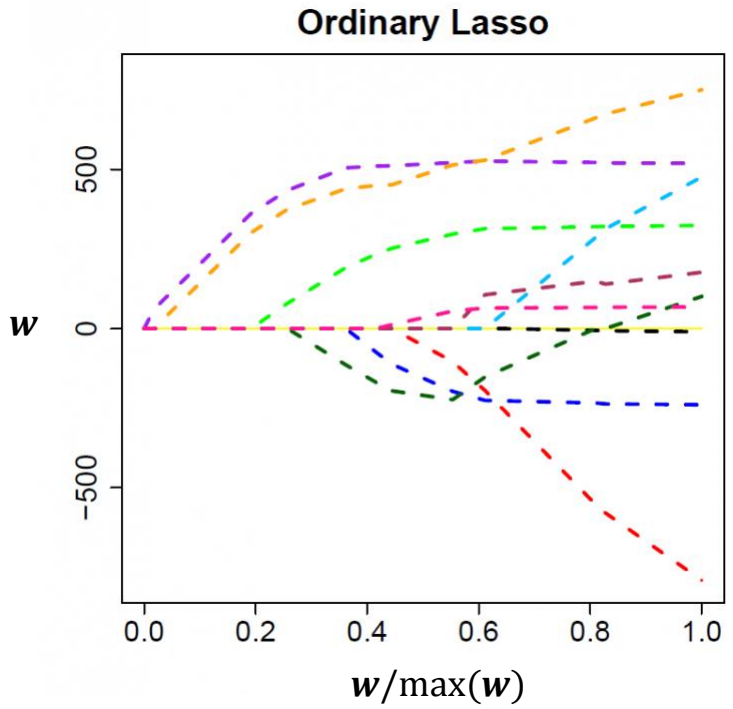
Lasso (L1-regularized Linear Regression)

Shorthand for Least Absolute Shrinkage and Selection Operator

Linear regression with L1-regularization

$$\min_{w} \frac{1}{N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\Omega(\boldsymbol{w})$$

$$\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$$



Not all variables become zero at the same time – results in variable selection

L1-Regularization for General Loss

L1-regularization can be coupled with various types of loss to obtain sparse models
e.g.) logistic regression, SVM

$$\min_{w} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \lambda \Omega(\boldsymbol{w})$$

$$l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} [(1 - y_i) \boldsymbol{w}^T \boldsymbol{x}_i + \log(1 + \exp(-\boldsymbol{w}^T \boldsymbol{x}_i))]$$

$$\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 \qquad \|\boldsymbol{w}\|_1 \leq \lambda$$

How can we then **solve** for such l1-regularized objectives?

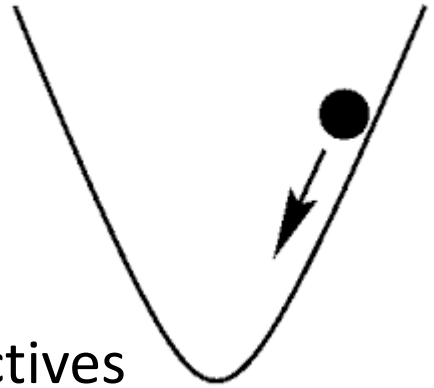
Gradient Descent

Determine the descent direction as the gradient at the point, determine the step size t, and then move to that direction.

given a starting point $x \in \mathbf{dom}\, f$.
repeat

1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

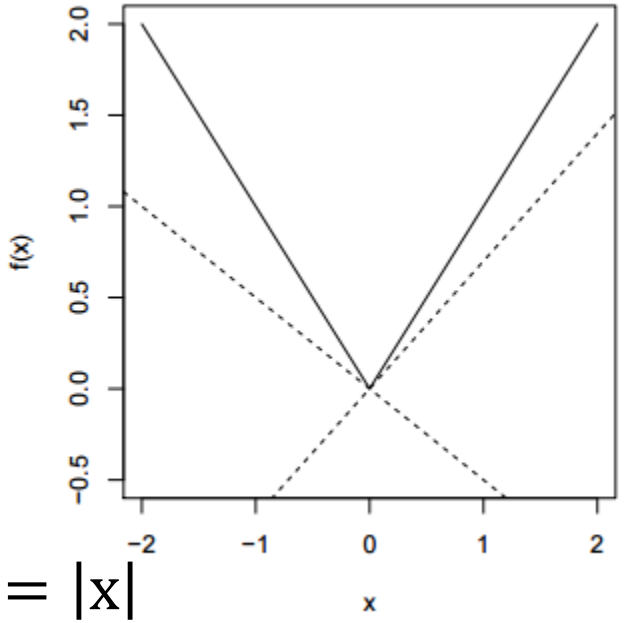


Repeat this process until a stopping criterion is met.

Cannot be applied to optimization of non-smooth objectives

Subgradient Method

Compute a set of gradient, defining the gradient on non-smooth points



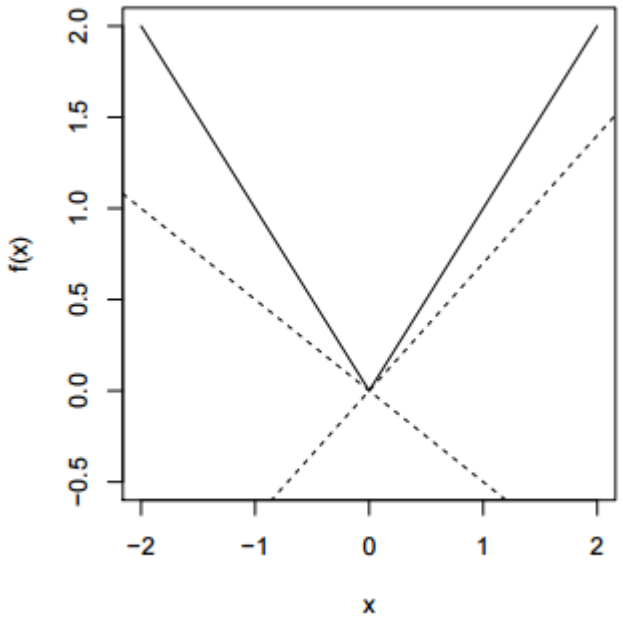
$f: R \to R, \mathrm{f(x)} = |\mathrm{x}|$

For $x \neq 0$, the subgradient $g = sign(x)$

For $x = 0$, subgradient g is any element of [-1,1].

Subgradient Method

A vector g is a subdifferential of f at x, if $f(y) = f(x) + d^T(y - x), \forall y$



Suffers from slow convergence, and solutions obtained are usually non-sparse

Proximal Gradient

Specifically tailored for regularized optimization problems where $g(\boldsymbol{w})$ is differentiable but $h(\boldsymbol{w})$ is a general convex function.

smooth

smooth or nonsmooth

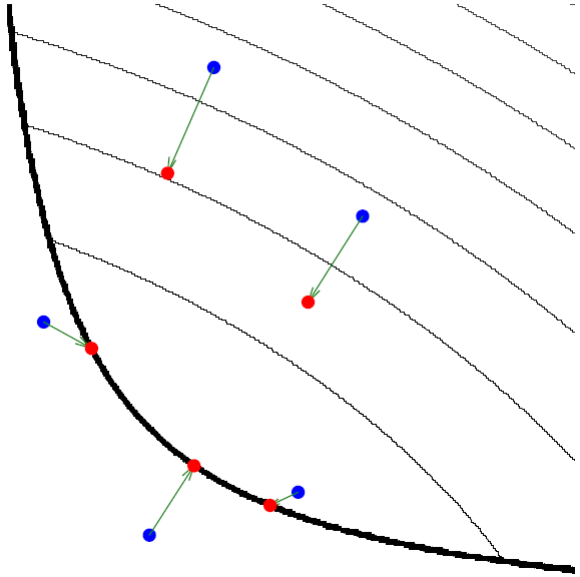
$$\min_{\boldsymbol{w}} g(\boldsymbol{w}) + h(\boldsymbol{w})$$

$$\boldsymbol{w}_{t+1} = prox(\boldsymbol{w}_t - \alpha \nabla g(\boldsymbol{x}_t))$$

Solution obtained by taking a gradient step only on g(w)

$$prox(\boldsymbol{w}) = argmin_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}^*\|_2^2 + \alpha h(\boldsymbol{w})$$

Euclidean Projection



The proximal operator, $prox(\boldsymbol{w})$ should be efficient – closed form solution preferred

Proximal Operator for L1-norm regularization

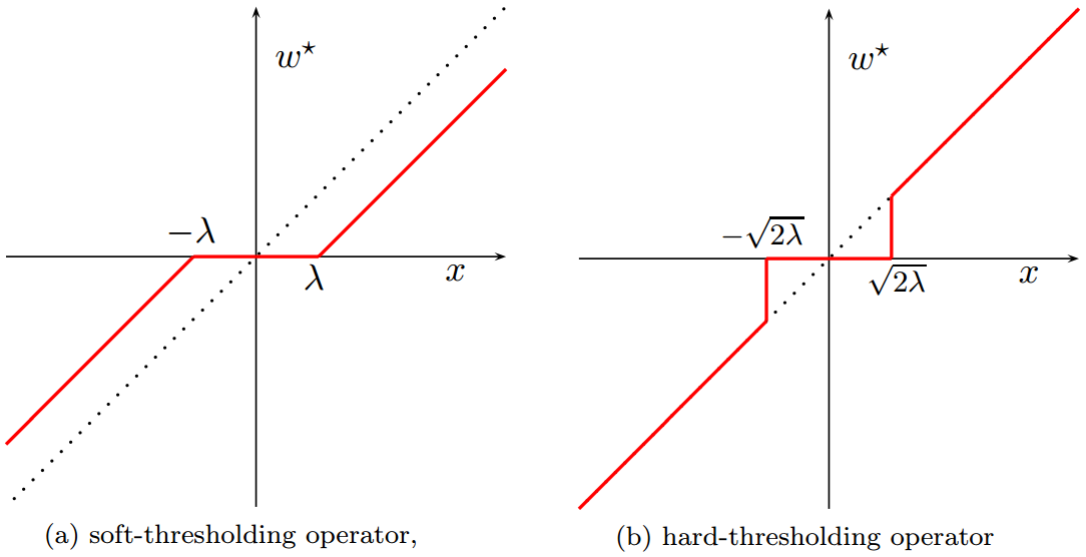
Iterative soft-thresholding operator – reduce the absolute value of each component of $\boldsymbol{w}$ by lambda, and if the resulting value is below zero, set it to zero.

$$\min_{\boldsymbol{w}} g(\boldsymbol{w}) + h(\boldsymbol{w})$$

$$h(\boldsymbol{w}) = \lambda \|\boldsymbol{w}\|_1$$

$$\boldsymbol{w}^* = \boldsymbol{w}_t - \alpha \nabla g(\boldsymbol{x}_t)$$

$$prox(\boldsymbol{w}) = sign(\boldsymbol{w}^*)[|\boldsymbol{w}^*| - \lambda\,]_+$$



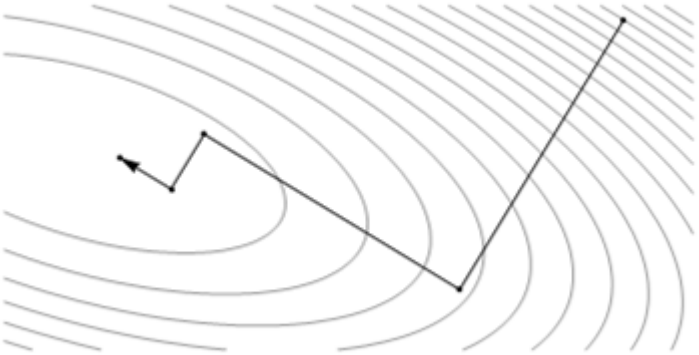
(a) soft-thresholding operator, (b) hard-thresholding operator

Coordinate Descent

Optimize one variable at a time while fixing all others.

$$\min_{w_j} \nabla_j f(\boldsymbol{w}^t)\left(\boldsymbol{w}_j - \boldsymbol{w}_j^t\right) + \frac{1}{2}\nabla_{jj}^2 f(\boldsymbol{w}^t)\left(\boldsymbol{w}_j - \boldsymbol{w}_j^t\right) + \lambda\left|\boldsymbol{w}_j\right|$$

$$\boldsymbol{w}_j^* = prox_{\frac{\lambda}{\nabla_{jjf}}|\cdot|}\left(\boldsymbol{w}_{\mathrm{j}}^{\mathrm{t}} - \frac{\nabla_j f\left(\boldsymbol{w}_j^t\right)}{\nabla_{jj}^2 f}\right)$$



$\boldsymbol{w}_j^*$ can be obtained by solving for the loss with coordinate j and soft-thresholding the solution.

[Bach et. al] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, Optimization with Sparsity Inducing Penalties, F & T in Machine Learning, 2012

Stochastic Subgradient Descent

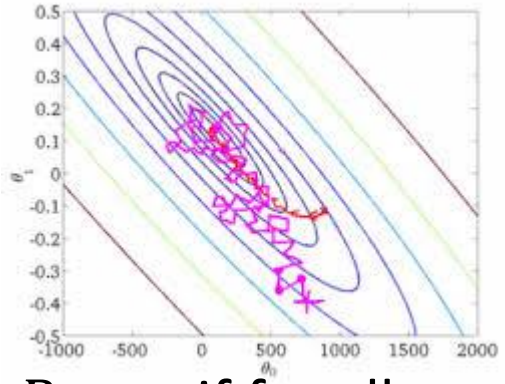
Subgradient method using noisy unbiased subgradients for scalable optimization

step size at iteration t

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - a_t \widetilde{g_t}$$

gradient direction at iteration t

$$E(\widetilde{g_t}|\boldsymbol{w}_t) = g_t \in \partial f(\boldsymbol{w}_t)$$



A random vector is a noisy unbiased subgradient for $f: R \to R$ at x, if for all z

$$f(\boldsymbol{z}) \geq f(\boldsymbol{w}) + (E\,\tilde{g})^T(\boldsymbol{z} - \boldsymbol{w})$$

Define the optimal value as $f^* = \min\{f(\boldsymbol{w}_1), \ldots, f(\boldsymbol{w}_t)\}$

Scales and works well for ML problems

However, SGD often does not generate sparse solutions for l1-regularized objectives

Regularized Dual Averaging Method

Consider all past subgradients of the loss and the whole regularization term when solving for a regularized objective

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left\{ \langle \overline{\boldsymbol{g}_t}, w \rangle + \Omega(\boldsymbol{w}) + \frac{\beta_t}{t} h(\boldsymbol{w}) \right\}$$

Positive sequence - $\gamma\sqrt{t}$

auxiliary Strongly convex function

$$\overline{\boldsymbol{g}_t} = \frac{t-1}{t}\overline{\boldsymbol{g}_{t-1}} + \frac{1}{t}\boldsymbol{g}_t$$

Average of all gradients over t iterations

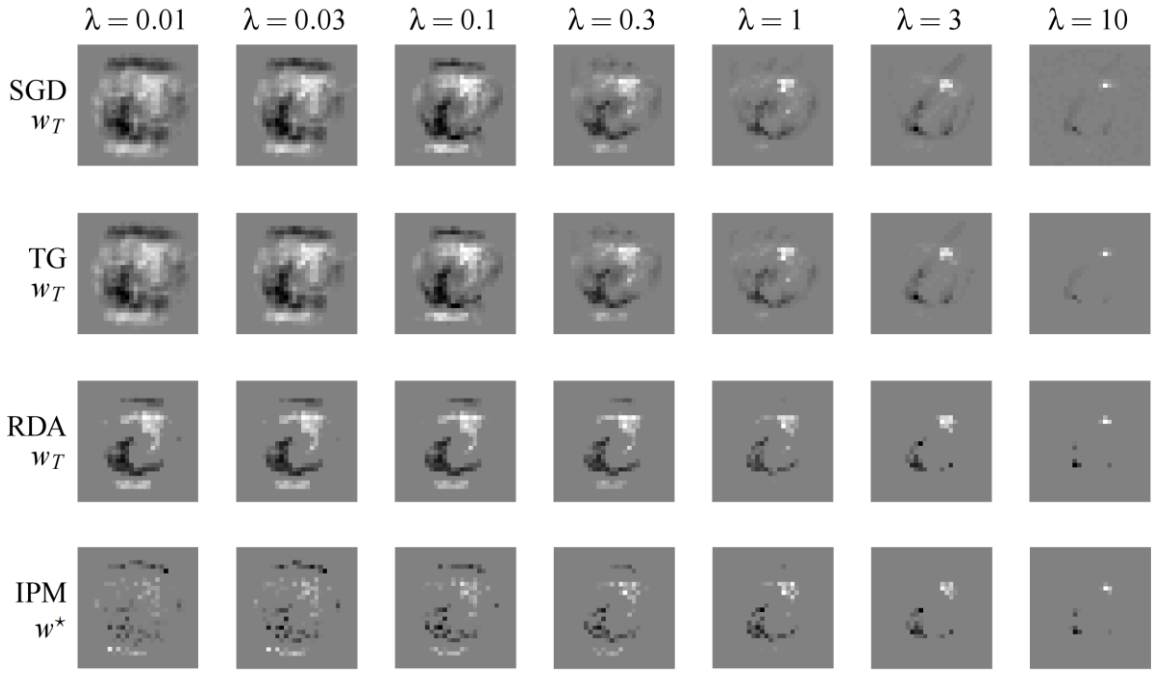
$$w_{t+1}^i = \begin{cases} 0, & \left|g_t^i\right| \le \lambda \\ -\frac{\sqrt{t}}{\gamma}\left(g_t^i - \lambda sign\left(g_t^i\right)\right), & otherwise \end{cases}$$

For l1-regularization RDA has a closed form solution

[Xiao et. al] L. Xiao, Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization, JMLR 2010

Regularized Dual Averaging Method

RDA obtains sparser solutions compared to those obtained by SGD and proximal gradient

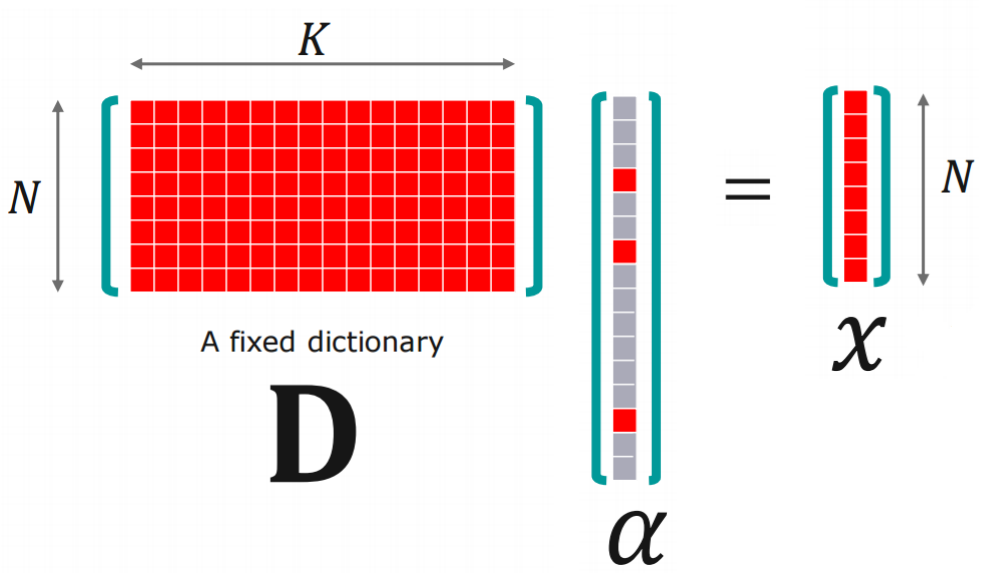


RDA has a convergence rate of $1/\sqrt{T}$ for general convex regularizers.

[Xiao et. al] L. Xiao, Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization, JMLR 2010

Sparse Coding

Find sparse decomposition of each data instance x_i, using some known dictionary atoms



$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\Omega(\boldsymbol{\alpha})$$

data instance

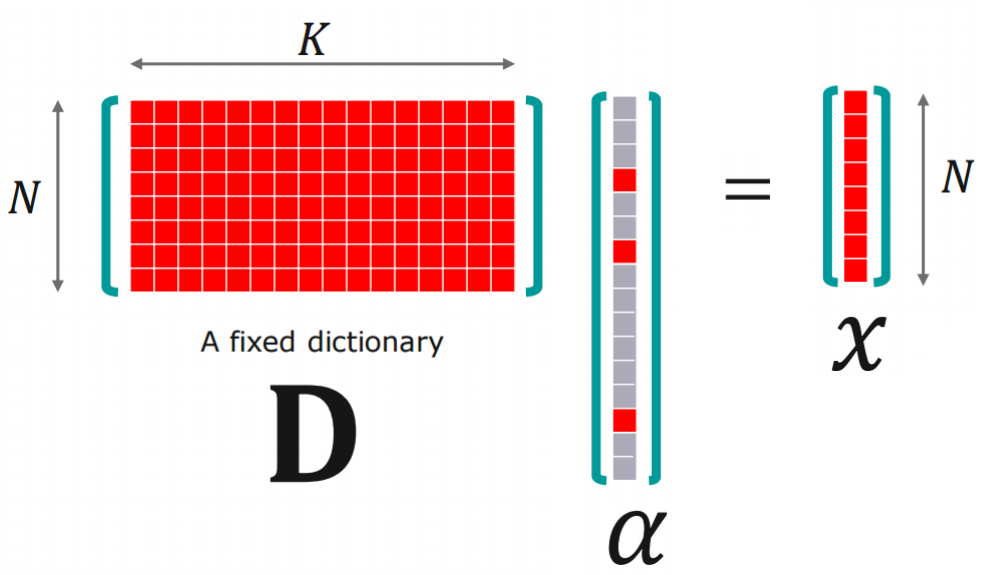
sparse code

Sparsity-inducing regularization

dictionary

Sparse Coding and Dictionary Learning

Find both a sparse decomposition $\boldsymbol{\alpha}_i$ of each data instance $\boldsymbol{x}_i$, and the bases $\boldsymbol{D}$



$$\min_{\boldsymbol{\alpha},\boldsymbol{D}} \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\Omega(\boldsymbol{\alpha})$$

data instance

sparse code

Sparsity-inducing regularization

dictionary

Can use alternating optimization to solve for each variable at a time.

Alternative Sparse Methods

Use greedy algorithms to directly solve for l0-norm

$$\min_{w} \frac{1}{N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 \qquad s.t.\ \|\boldsymbol{w}\|_0 \leq \lambda$$

$$\min_{\alpha} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 \qquad s.t.\ \|\boldsymbol{\alpha}\|_0 \leq \lambda$$

Enforce sparsity by setting the desired number of nonzero variables.

- Matching Pursuit, Orthogonal Matching Pursuit

Matching Pursuit

At each step, select the column that is most correlated with the input

$$\min_{\alpha} \|\boldsymbol{x} - \boldsymbol{D\alpha}\|_2^2 \qquad s.t.\ \|\boldsymbol{\alpha}\|_0 \leq \lambda$$

$\boldsymbol{\alpha} \leftarrow 0$

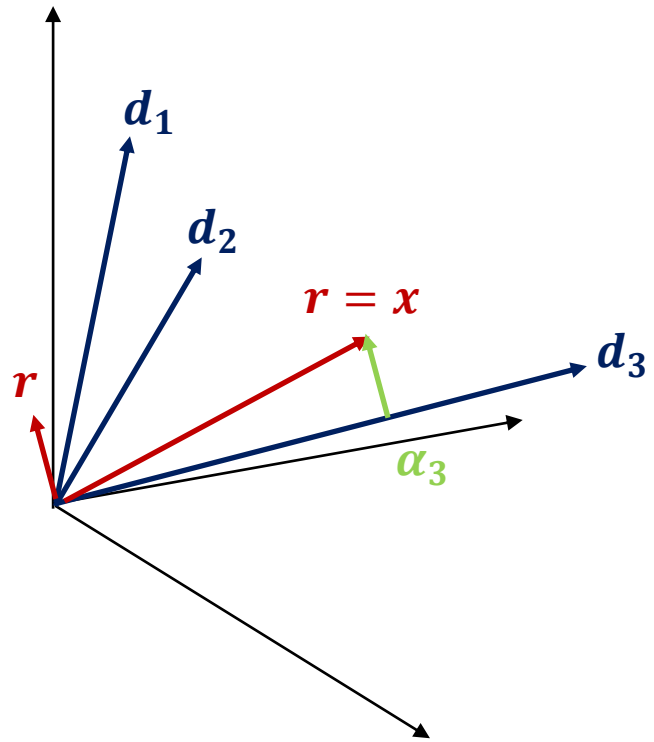
$\boldsymbol{r} \leftarrow \boldsymbol{x}$

while $\|\boldsymbol{\alpha}\|_0 \leq \lambda$

$\boldsymbol{i} \leftarrow argmax_{\{i=1,\ldots,p\}} \left|\boldsymbol{d}_i^T \boldsymbol{r}\right|$

$\boldsymbol{\alpha_i} \leftarrow \boldsymbol{\alpha_i} + \boldsymbol{d_i^T r}$

$\boldsymbol{r} \leftarrow \boldsymbol{r} - \left(\boldsymbol{d_i^T r}\right)\boldsymbol{d_i}$



Orthogonal Matching Pursuit

At each step, select the column that helps reduce the objective the most

$$\min_{\alpha} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 \qquad s.t.\ \|\boldsymbol{\alpha}\|_0 \leq \lambda$$

$\Gamma = \emptyset$

for $i = 1, \dots, \lambda$ do

$$i = argmin_{i \in \Gamma^c} \left\{ \min_{\alpha'} \left\| \boldsymbol{x} - \boldsymbol{D}_{\Gamma \cup \{i\}} \boldsymbol{\alpha}' \right\|_2^2 \right\}$$

$$\Gamma \leftarrow \Gamma \cup \{i\}$$

$$r \leftarrow \left(I - \boldsymbol{D}_\Gamma \left(\boldsymbol{D}_\Gamma^{\mathrm{T}} \boldsymbol{D}_\Gamma \right)^{-1} \boldsymbol{D}_\Gamma^T \right) \boldsymbol{x}$$

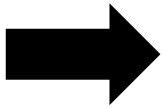
$$\alpha_\Gamma \leftarrow \left(\boldsymbol{D}_\Gamma^{\mathrm{T}} \boldsymbol{D}_\Gamma \right)^{-1} \boldsymbol{D}_\Gamma^T \boldsymbol{x}$$

All coefficients extracted so far are updated at each step

Image Restoration

Eliminate noise from the original image

Original

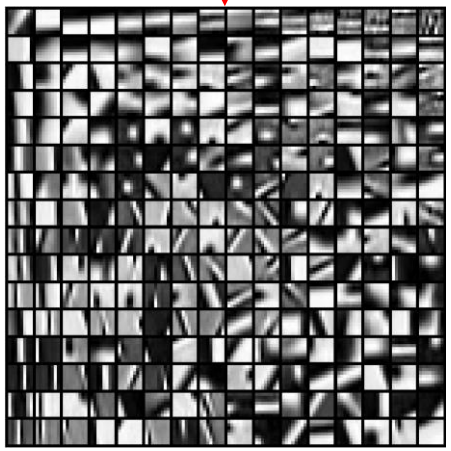


Restored

Image Restoration with Sparse Coding

Reconstruct the image with sparse combination of learned bases

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1$$



Bases

Image Inpaiting

Works with even larger level of noise

Original

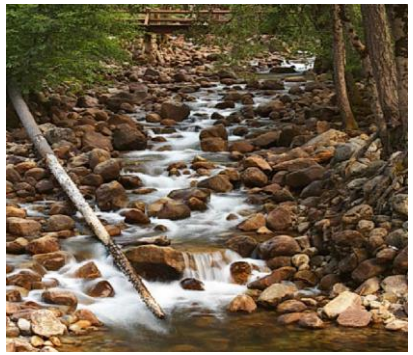
Restored

Self-taught Learning

Transfer learning when source domain is **unlabeled** and only remotely related to the target task – based on the intuition that human learning is largely unsupervised.

Source

$\{x_u^{(i)}\}_{i=1}^{k} \quad x_u^{(i)} \in R^n, k >> m$



Target

$\{(x_l^{(i)}, y^{(i)})\}_{i=1}^{m} \quad x_l^{(i)} \in R^n, y^{(i)} \in \{1,\ldots,T\}$

Learn a higher-level, more abstract feature for the given feature type (modality)

[Raina et al.] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007

Self-taught Learning with Sparse Coding

Given unlabeled data x, find good bases b using sparse coding and dictionary learning.

Reconstruction error

Bases

Sparsity Regularization

L2 regularization

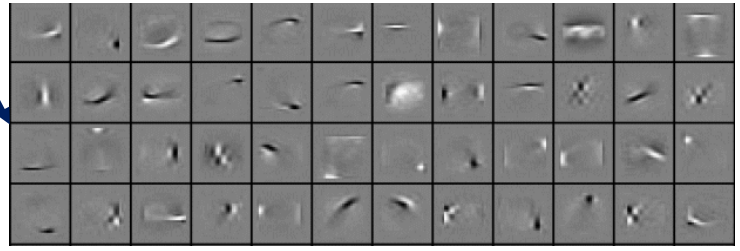
$$\min_{a,b} \sum_i \left|\left|x_u^{(i)} - \sum a_j^{(i)} b_j\right|\right|_2^2 + \beta \sum_i ||a^{(i)}||_1 \quad ||b_j||_2^2 \leq 1, \forall j \in 1, \ldots, s$$

Sparse codes

Natural Image

Edges

Handwritten Characters



Strokes

[Raina07] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007

Self-taught Learning with Sparse Coding

Use alternating optimization - Solve for each variable while fixing the other, and alternate the process until convergence (Efficient algorithm introduced in 07)

1) Fix b and solve for a → L1-regularized least squares problem

$$\min_{a} \sum_i ||x_u^{(i)} - \sum a_j^{(i)} b_j||_2^2 + \beta \sum_i ||a^{(i)}||_1$$

Fixed

2) Fix a and solve for b → L2-constrained least squares problem

$$\min_{b} \sum_i ||x_u^{(i)} - \sum a_j^{(i)} b_j||_2^2 \quad ||b_j||_2^2 \leq 1, \forall j \in 1, \ldots, s$$

Fixed

Alternate Until Convergence

[Raina07] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007

[Lee07] H. Lee, A. Battle, R. Raina, A. Y. Ng, Efficient Sparse Coding Algorithms, NIPS 2007

Self-taught Learning with Sparse Coding

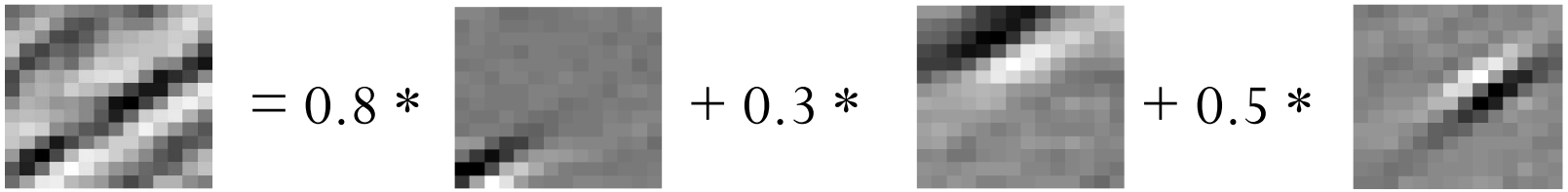
Then reconstruct the examples in the target dataset using the learned bases.

Reconstruction error

Sparse codes

$$\min_{a,b} \sum_i \left|\left| x_u^{(i)} - \sum a_j^{(i)} b_j \right|\right|_2^2 + \beta \sum_i ||a^{(i)}||_1 \qquad ||b_j||_2^2 \leq 1, \forall j \in 1, \ldots, s$$

Bases



$$x_l = 0.8 * b_{87} + 0.3 * b_{376} + 0.5 * b_{411}$$

Then train a classifier (such as a SVM) over the obtained sparse codes

[Raina et al.] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007

Self-taught Learning with Sparse Coding

Results show that information gained from unlabeled data is actually useful.

Image Classification

Method	Accuracy
Baseline	16%
PCA	37%
Sparse coding	**47%**

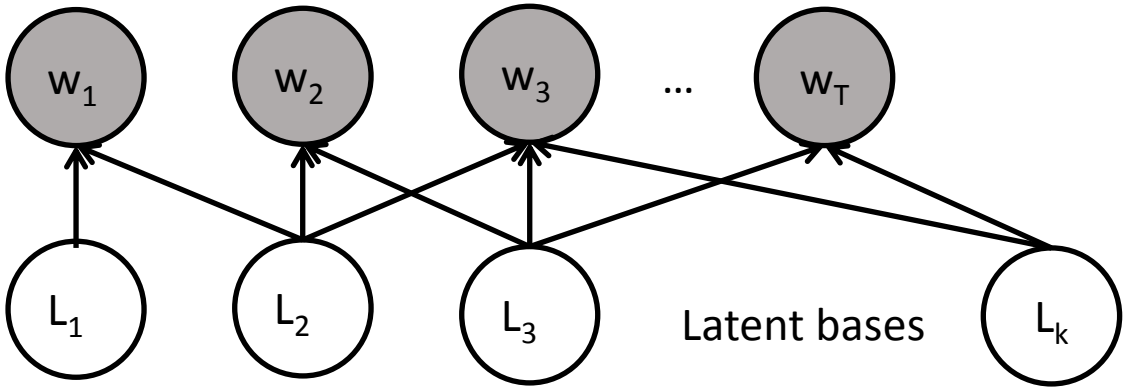
Handwritten Character Recognition

Method	Accuracy
Raw	54.8%
PCA	54.8%
Sparse coding	**58.5%**

[Raina et al.] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007

Learning Task Grouping in Multitask Learning

Allow the learners to selectively share the information across the tasks.

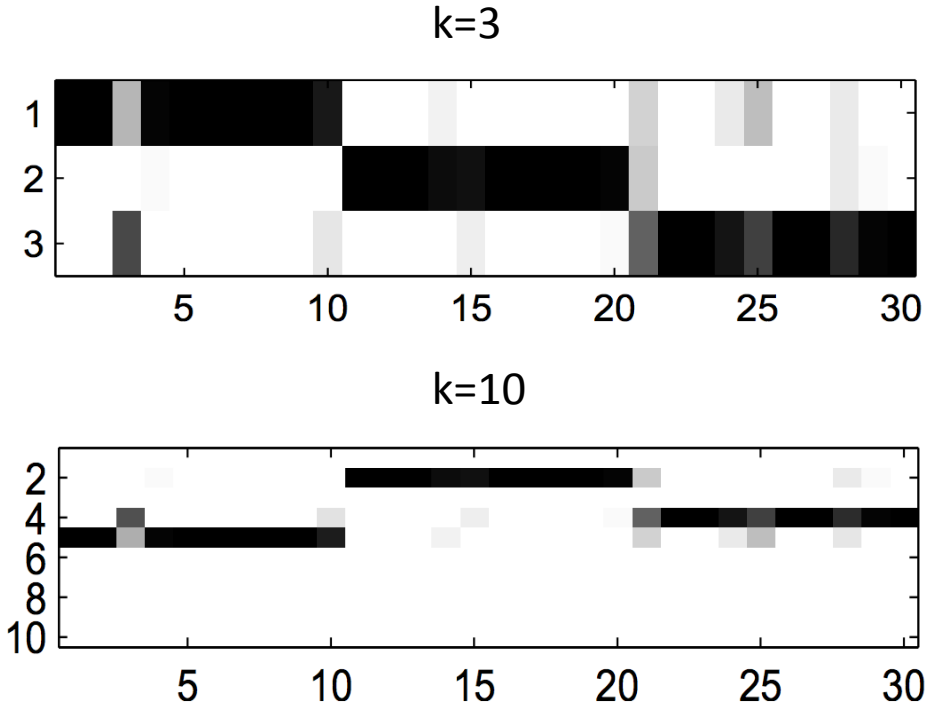


Shared parameter bases

$$\sum_t \sum_{(\boldsymbol{x}_{ti},\boldsymbol{y}_{ti})\in Z_t} \mathcal{L}(y_{ti}, \boldsymbol{s}_t' \boldsymbol{L}' \boldsymbol{x}_{ti}) + \mu||\boldsymbol{S}||_1 + \lambda||\boldsymbol{L}||_F^2$$

Task-specific sparse weight

$$\boldsymbol{w}_t = \boldsymbol{L}\boldsymbol{s}_t$$

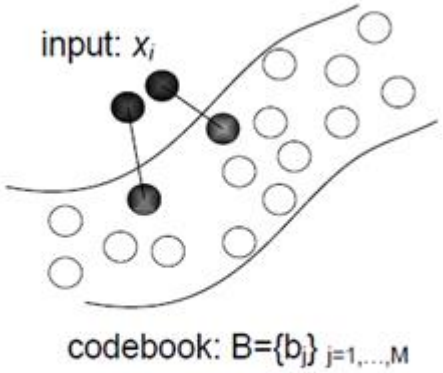


Can discover true support without having to provide the number of groups

[Kumar13] A. Kumar, H. Daume III, Learning Task Grouping and Overlap in Multi-Task Learning

Sparse Feature Encoding

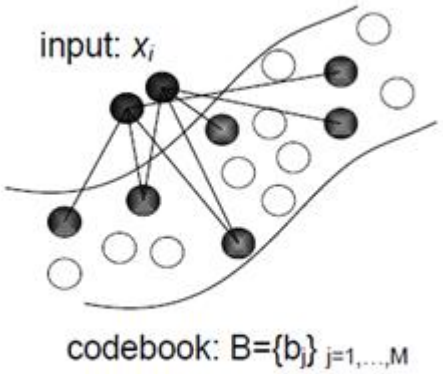
Soft-Encoding – encodes each feature as a sparse combination of visual words.



VQ

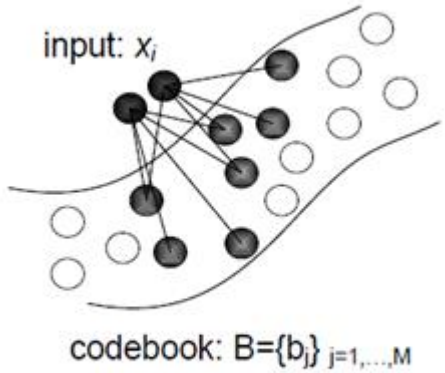
$$\arg\min_{\mathbf{C}} \sum_{i=1}^{N} ||\mathbf{x}_i - \mathbf{B}\mathbf{c}_i||^2$$

$$s.t. \|\mathbf{c}_i\|_{\ell^0} = 1, \|\mathbf{c}_i\|_{\ell^1} = 1, \mathbf{c}_i \succeq 0, \forall i$$



SC

$$\arg\min_{\mathbf{C}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}\mathbf{c}_i\|^2 + \lambda\|\mathbf{c}_i\|_{\ell^1}$$



LLC

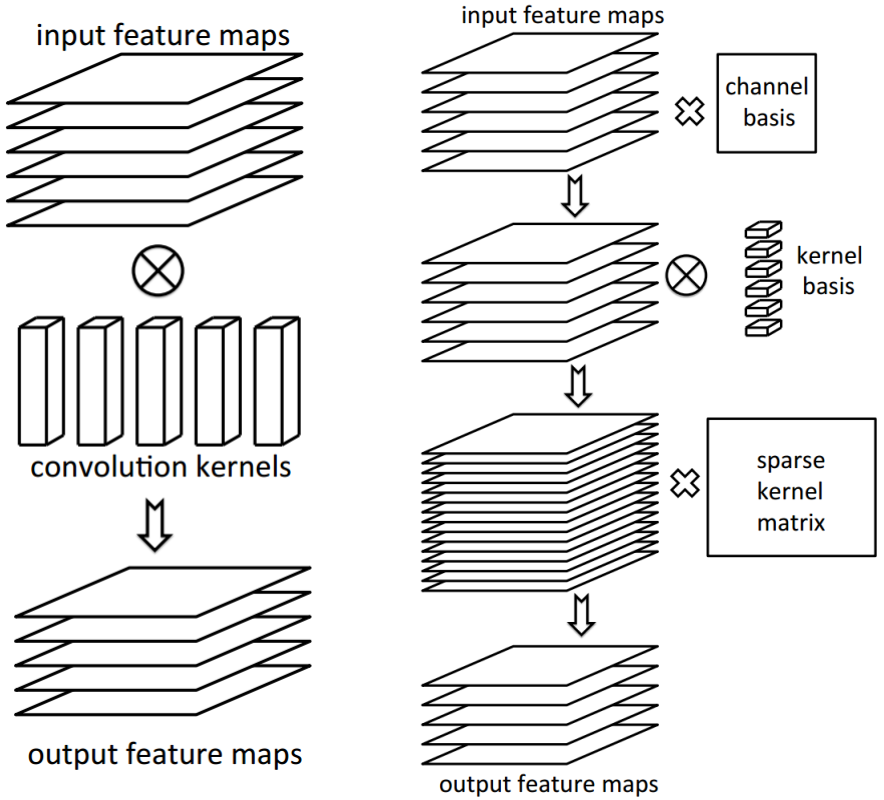
$$\min_{\mathbf{C}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{B}\mathbf{c}_i\|^2 + \lambda||\mathbf{d}_i \odot \mathbf{c}_i||^2$$

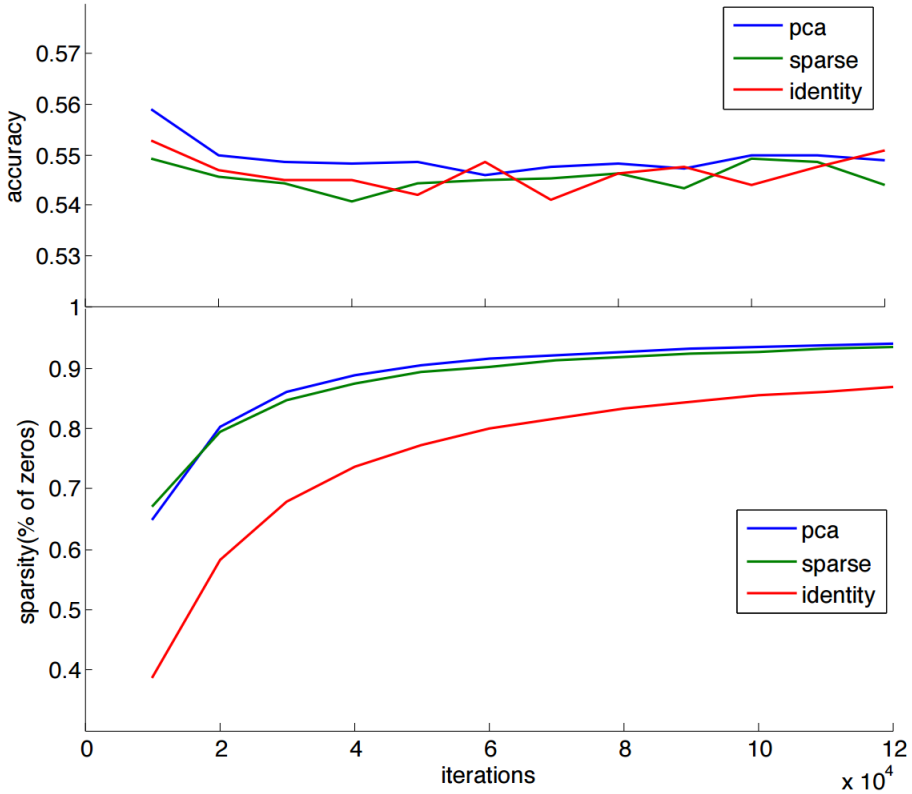
$$s.t.\ \mathbf{1}^\top \mathbf{c}_i = 1,\ \forall i$$

[Wang et. al] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, Locality-Constrained Linear coding for Image Classification, CVPR 2010

Learning Sparse Filters for a CNN

Use sparse coding to reduce the number of convolutional kernels

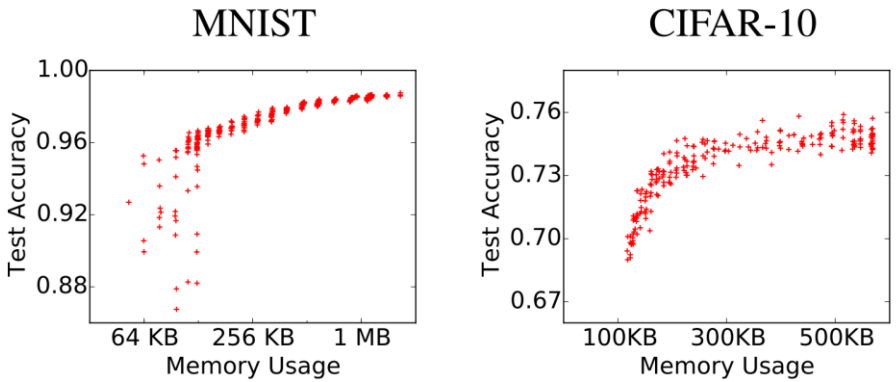




[Liu et. al] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Penksy, Sparse Convolutional Neural Networks, CVPR 2015

Learning Sparse Weights for a CNN

Learn sparse network weights to reduce memory usage



ImageNet

Model	Top-1	Top-5	Memory
Reported [11]	59.3%	81.8%	-
Caffe Version [9]	57.28%	80.44%	233 MB
Sparse (ours)	55.60%	80.40%	58 MB

[Collins et. al] M. D. Collins and P. Kohli, Memory Bounded Deep Convolutional Networks, arXiv:1412.1442

Summary

Sparsity regularizations result in learning a model with ***few nonzero*** parameters. It is beneficial for ***feature selection***, ***model analysis***, and ***model compression***.

L1-regularization uses 1-norm for regularization, which enduces sparsity. Since 1-norm is ***non-differentiabl***e, we can solve it through optimization methods such as ***subgradient descent***, ***proximal gradient***, and ***regularized dual averaging method***.

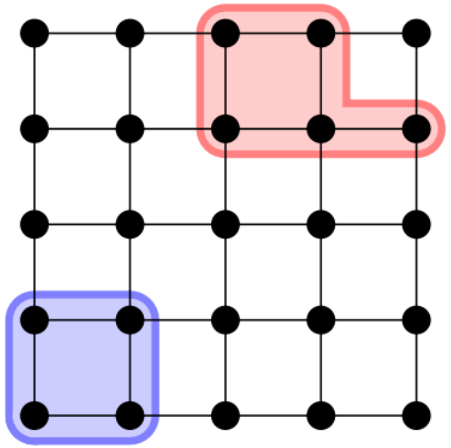
Sparse coding encodes an input variable as a ***sparse combination of some bases***. It is useful for multiple applications such as ***image restoration*** and ***transfer learning***.

Part 2: Structured Sparsity

- Introduction to Structured Sparsity
- Group-structured sparsity
- Block-structured sparsity
- Optimizing for (2,1)-norm
- Tree-structured sparsity
- Graph-structured sparsity
- Example: Image inpainting with hierarchical dictionary learning
- Example: Multi-task learning
- Example: Learning decorrelated attributes

Structured Sparsity

Sparsity regularizations that prefer ***certain structure*** among the variables over others.



Enables to exploit known structure – e.g.) To reconstruct handwritten characters, we can exploit the fact that they form connected components

(2,1)-norm

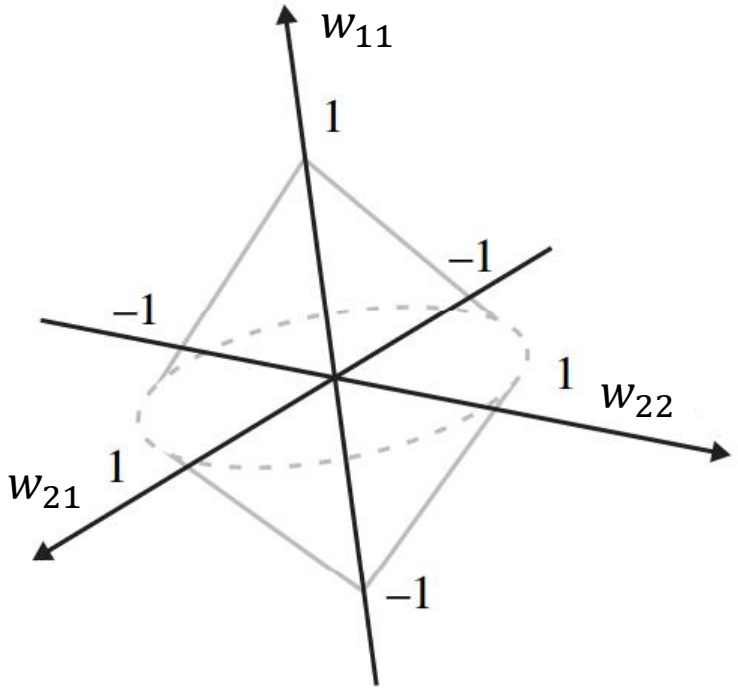
Mixed norm that has 1-norm over 2-norm groups.

$$\min_{\boldsymbol{w}} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_{2,1}$$

$$\|\boldsymbol{w}\|_{2,1} = \sum_{l}^{L} \|\boldsymbol{w}_l\|_2$$

2-norm

1-norm

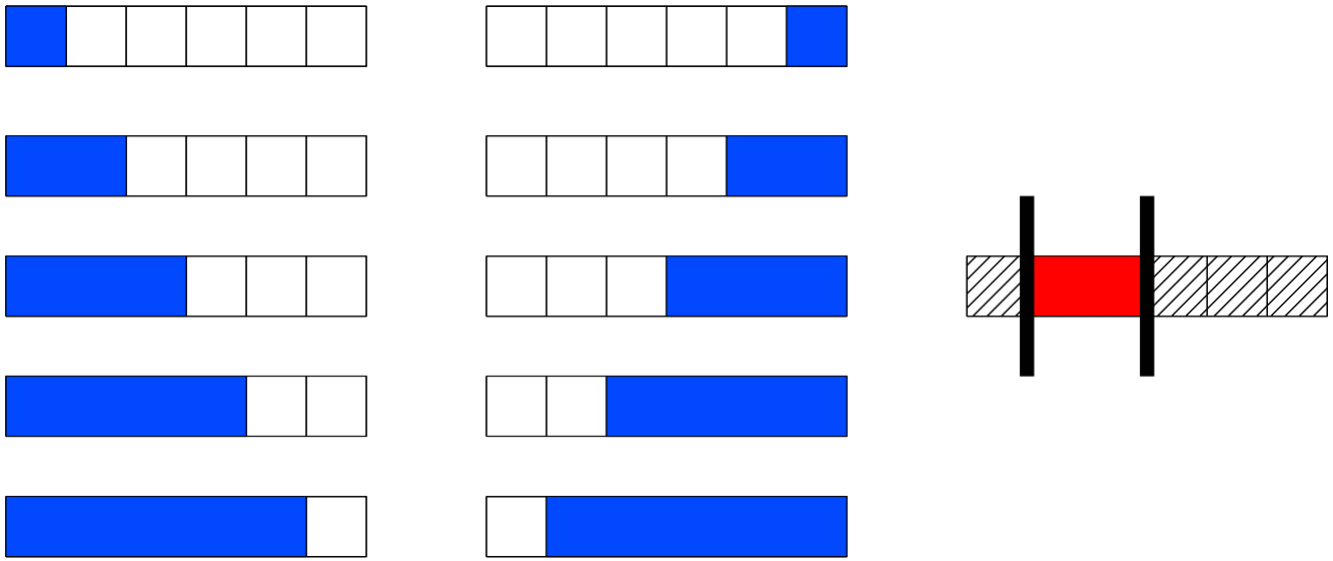


Used to promote sparsity at group level

[Yuan et al.] M. Yuan and Y. Lin, Model Selection and Estimation in Regression with Grouped Variables, Journal of Royal Statistics Society, 2006

Group Sparsity

Structured sparsity with groups of variables

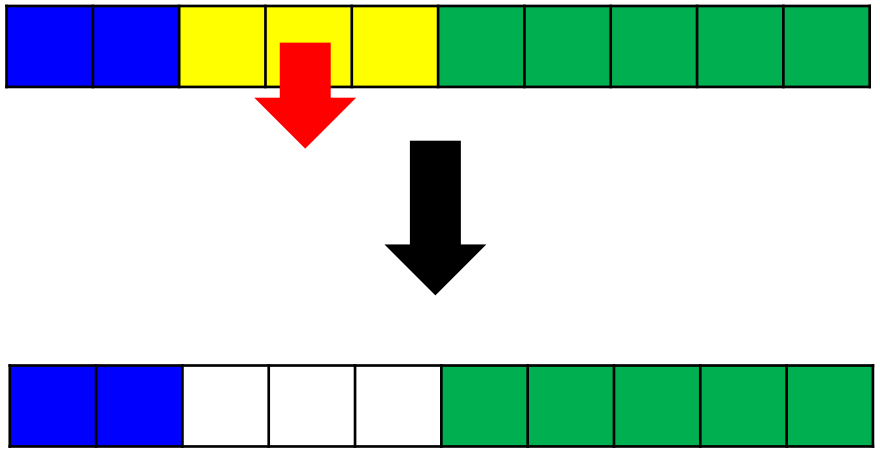


Group Sparsity

Use L2/L1-regularization to select / drop variables in a group

$$\min_{\boldsymbol{w}} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_{2,1}$$

$$\|\boldsymbol{w}\|_{2,1} = \sum_{l}^{L} \|\boldsymbol{w}_l\|_2$$

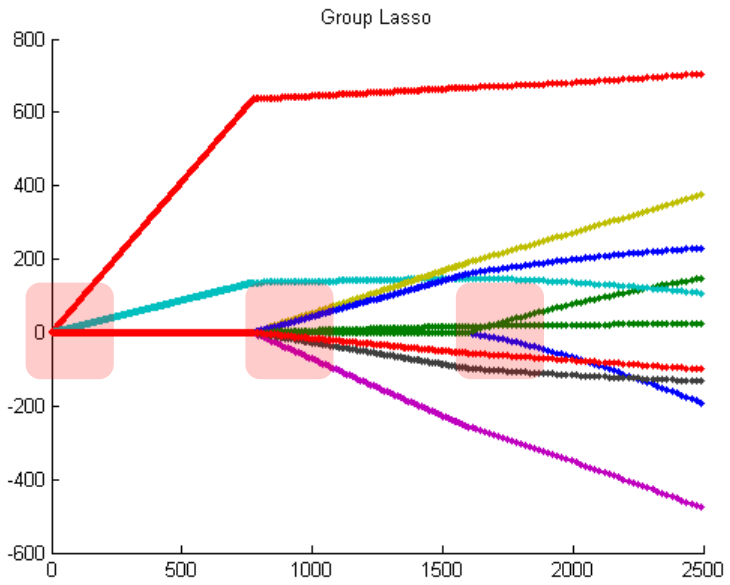


Used to promote sparsity at group level

Group Lasso

Correlated variables gets selected / dropped at the same time

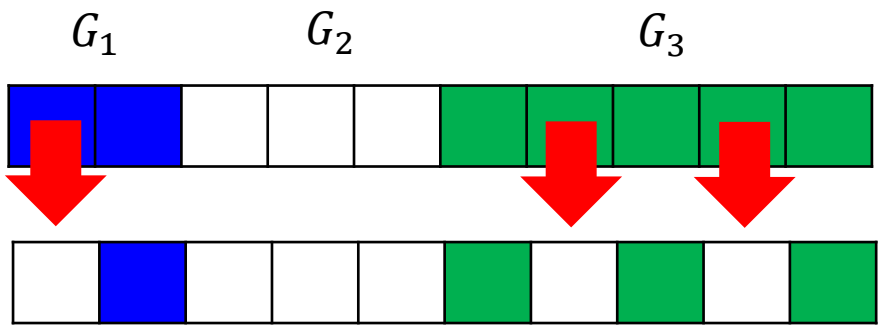
$$\min_{\boldsymbol{w}} \frac{1}{N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \sum_{l=1}^{L} \|\boldsymbol{w}_l\|_2$$



Sparse Group Lasso

Group lasso does not yield sparsity within a group. All variables selected will have non-zero values.

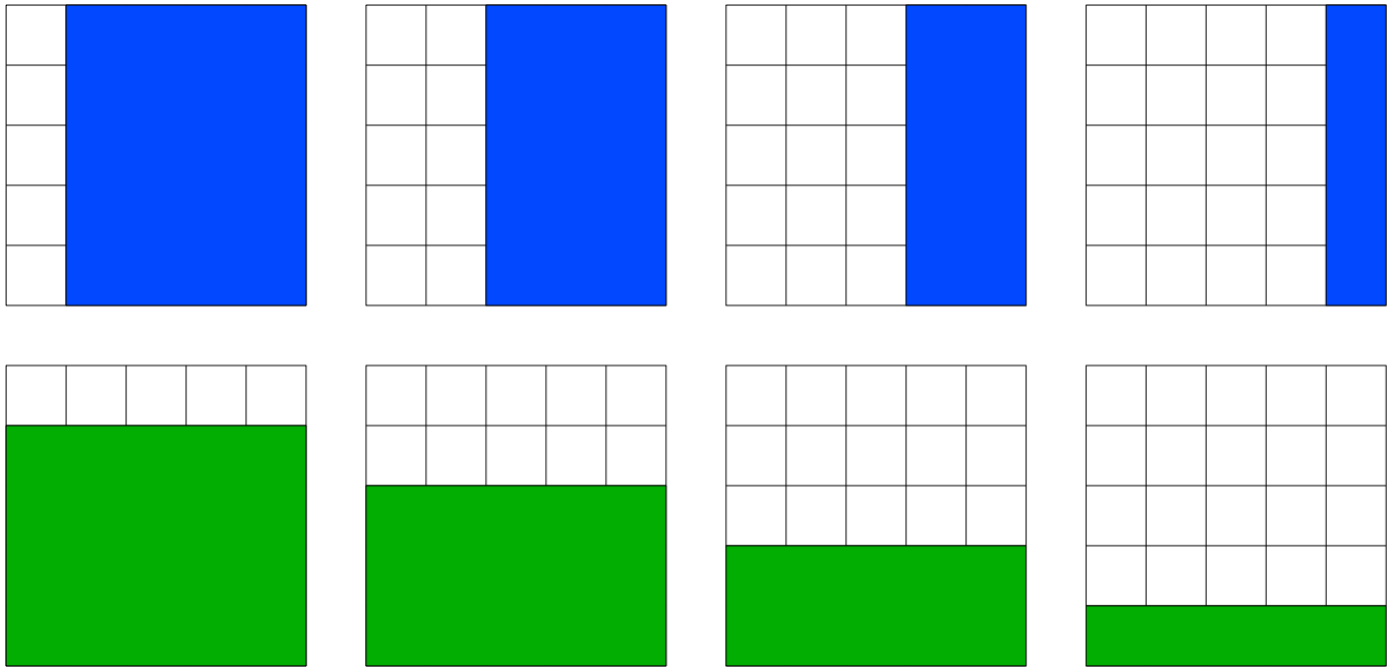
$$\min_{w} \frac{1}{N} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda_1 \sum_{l=1}^{L} \|\boldsymbol{w}_l\|_2 + \lambda_2 \|\boldsymbol{w}\|_1$$

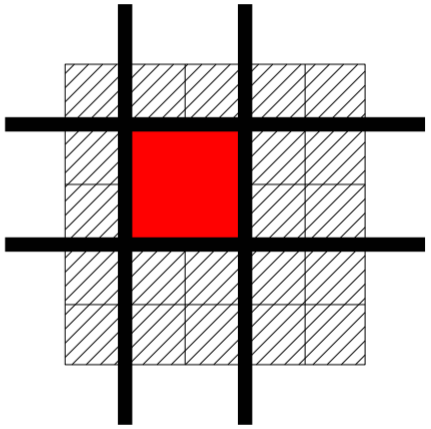


Use additional l1-regularization to yield sparsity at individual feature level.

Block Sparsity

Structured Sparsity on a 2D Grid



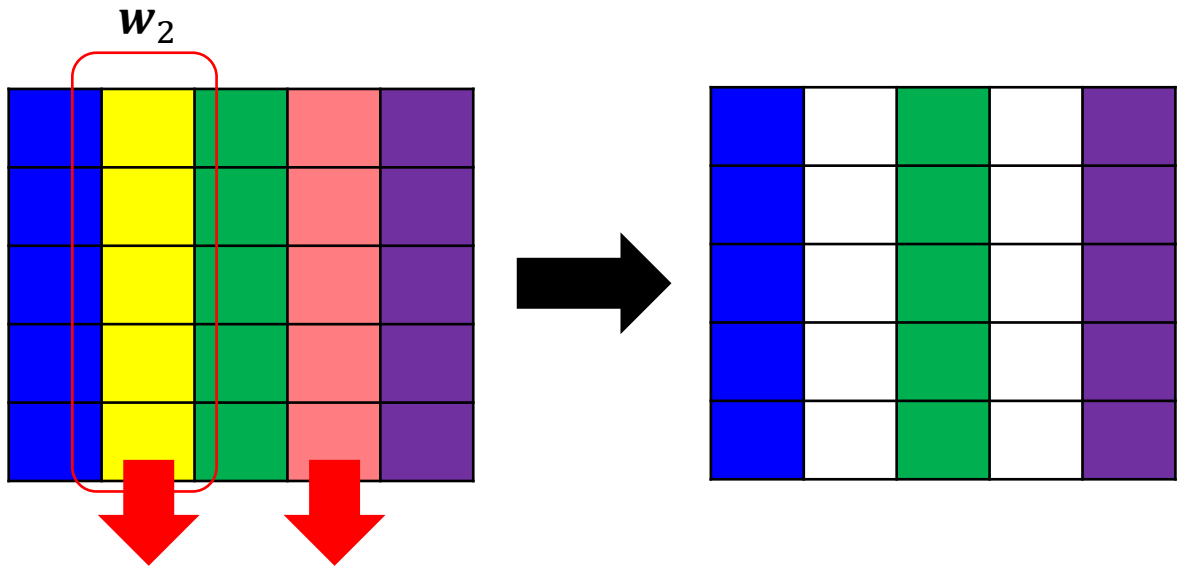


Block Sparsity

Same as group sparsity applied to a sequence. Group each row or columns using 2-norm, and apply 1-norm on the groups

$$\min_{\boldsymbol{w}} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_{2,1}$$

$$\|\boldsymbol{w}\|_{2,1} = \sum_{l}^{L} \|\boldsymbol{w}_l\|_2$$



Exclusive Lasso

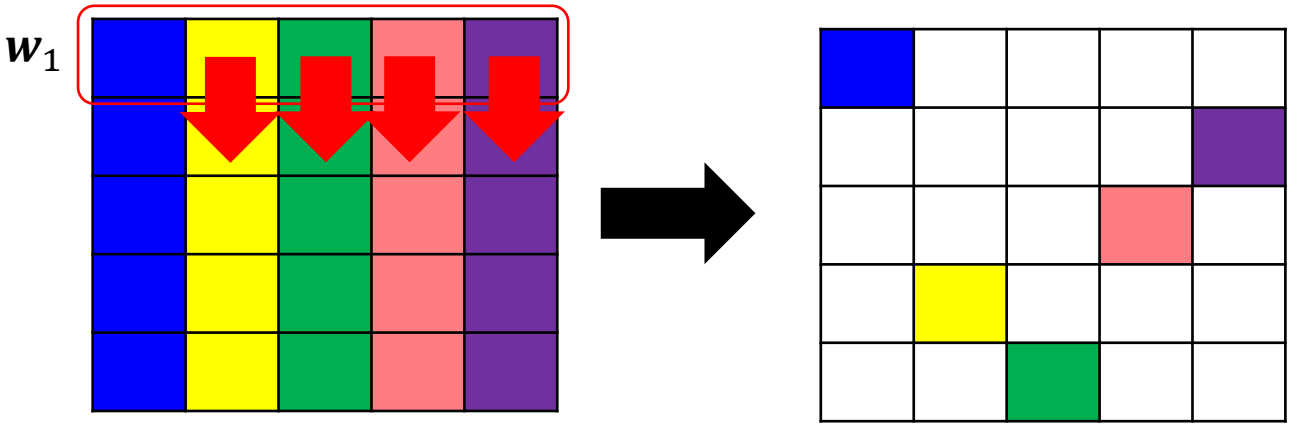
Promote competition for the features between tasks

$$\min_{\boldsymbol{w}} l(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{w}) + \lambda \Omega(\boldsymbol{w})$$

2-norm over the parameter for the same task

$$\Omega(\boldsymbol{w}) = \sum_{j}^{d} \left(\sum_{k}^{m} \left| \boldsymbol{w}_{\mathrm{k}}^{\mathrm{j}} \right| \right)^{2}$$

1-norm over the parameters for m tasks on the same feature dimension



[Zhou10] Y. Zhou, R. Jin, and S. C. H. Hoi, Exclusive Lasso for Multi-task Feature Selection, AISTAT 2010

Optimization for L2/L1-Regularized Objectives

(2,1)-norm is nonsmooth.

proximal operator

$$prox(\boldsymbol{w})_l = \left[1 - \frac{\lambda}{\|\boldsymbol{w}_l\|}\right]_+ \boldsymbol{w}_l$$

Regularized dual averaging

$$w_{t+1}^i = \begin{cases} 0, & if\ \left|g_t^i\right| \le \lambda \\ -\frac{1}{\sigma}\left(g_t^i - \lambda sign\left(g_t^i\right)\right), & otherwise \end{cases}$$

Can use proximal gradient and regularized dual averaging method

Block Coordinate Descent

Optimize a group of variable at a time, while fixing all others.

$$\boldsymbol{w}_g^* = prox_{\frac{\lambda}{L_t}}\left(\boldsymbol{w}_g - \frac{1}{L_t}\nabla_g f\left(\boldsymbol{w}_g^t\right)\right)$$

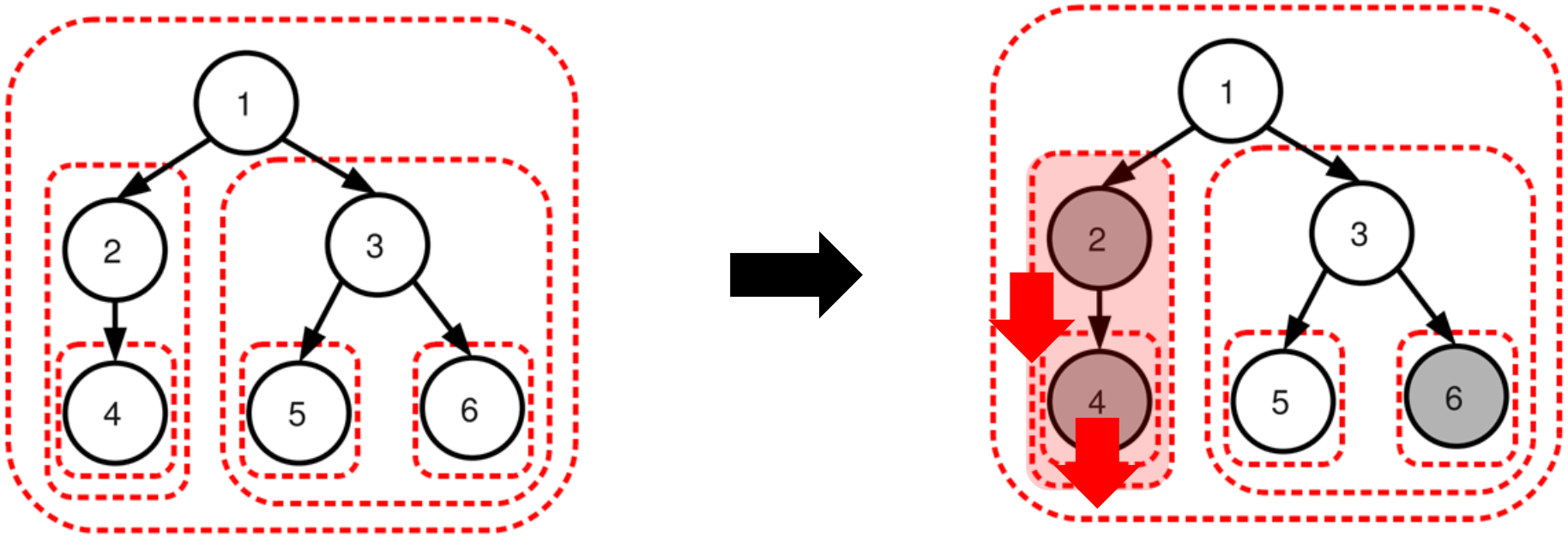
$$prox(\boldsymbol{w}) = \left[1 - \frac{\lambda}{\|\boldsymbol{w}\|}\right]_+ \boldsymbol{w}$$

The solution $\boldsymbol{w}_g^*$ can be obtained through group-soft thresholding

[Blondel et al.] M. Blondel, K. Seki, K. Uehara, Block Coordinate Descent Algorithms for Large-Scale Sparse Multiclass Classification

Tree Sparsity

Structural sparsity on a tree

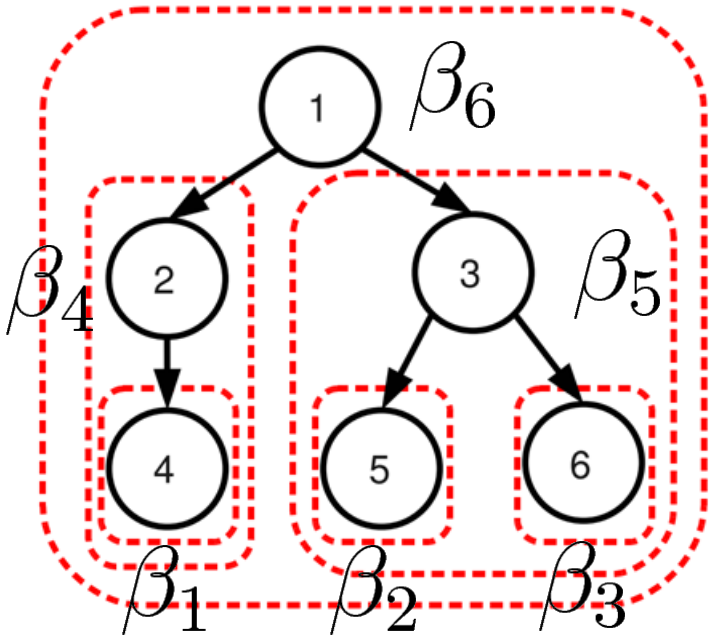


If a node is removed, then all its descendant nodes are dropped.

[Jenatton11a] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, Proximal Methods for Sparse Hierarchical Dictionary Learning, ICML 2010

Hierarchical Sparsity-Inducing Norm

Perform group lasso where each group $\mathcal{G}_j$ contains node j and all its descendants.



$$\Omega(\beta) = \sum_{g \in \mathcal{G}} w_g ||\boldsymbol{\beta}_g||_2$$

$$\boldsymbol{\beta}_1 = \beta_1 \quad \boldsymbol{\beta}_2 = \beta_2 \quad \boldsymbol{\beta}_3 = \beta_3$$

$$\boldsymbol{\beta}_4 = \{\beta_1, \beta_4\} \qquad \boldsymbol{\beta}_5 = \{\beta_2, \beta_3, \beta_5\}$$

$$\boldsymbol{\beta}_6 = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5\}$$

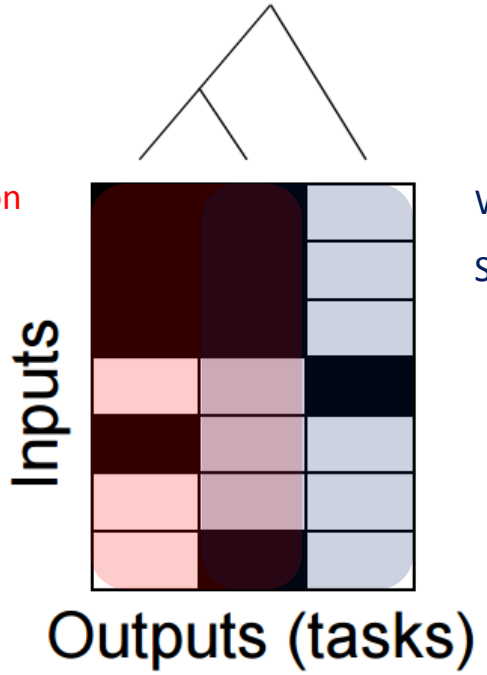
[Jenatton11a] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, Proximal Methods for Sparse Hierarchical Dictionary Learning, ICML 2010

Tree-guided Group Lasso

Exploit a given tree structure on the output values to recover the true structure in the parameters

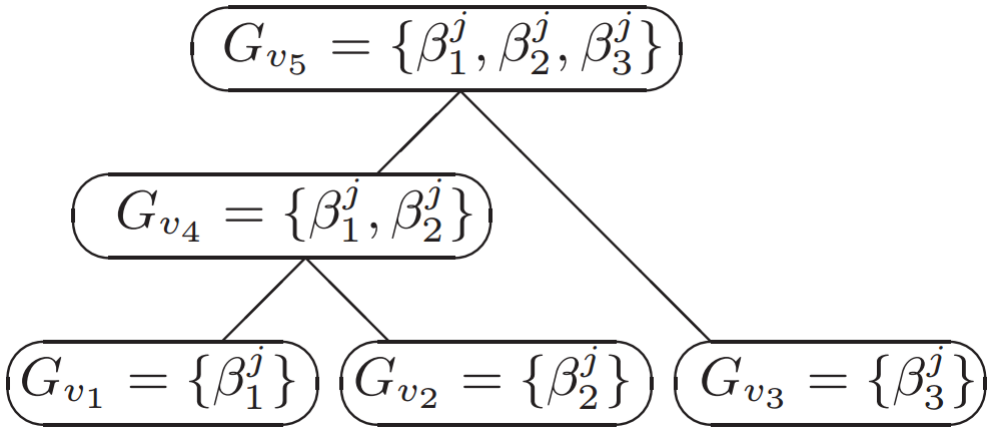
Tight correlation

Joint selection



Weak Correlation

Separate selection



Idea: use overlapping groups in group lasso

[Kim et al.] S. Kim and E. P. Xing, Tree Guided Group Lasso for Multi-Task Regression with Structured Sparsity, ICML 2010

Tree-guided Group Lasso

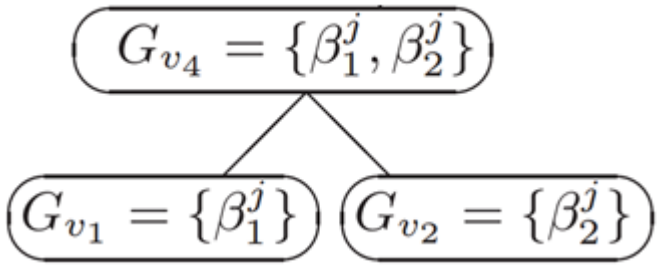
Promotes sharing and competition between tasks in hierarchical manner.

L1 regularization – promotes competition

$$\sum_j \left[h\left(|\beta_1^j| + |\beta_2^j|\right) + (1-h)\left(\sqrt{\left(\beta_1^j\right)^2 + \left(\beta_2^j\right)^2}\right) \right]$$

L2 regularization – promotes sharing

Decides the degree of competition and sharing



Simplest case where there are only two outputs

[Kim et al.] S. Kim and E. P. Xing, Tree Guided Group Lasso for Multi-Task Regression with Structured Sparsity, ICML 2010

Tree-guided Group Lasso

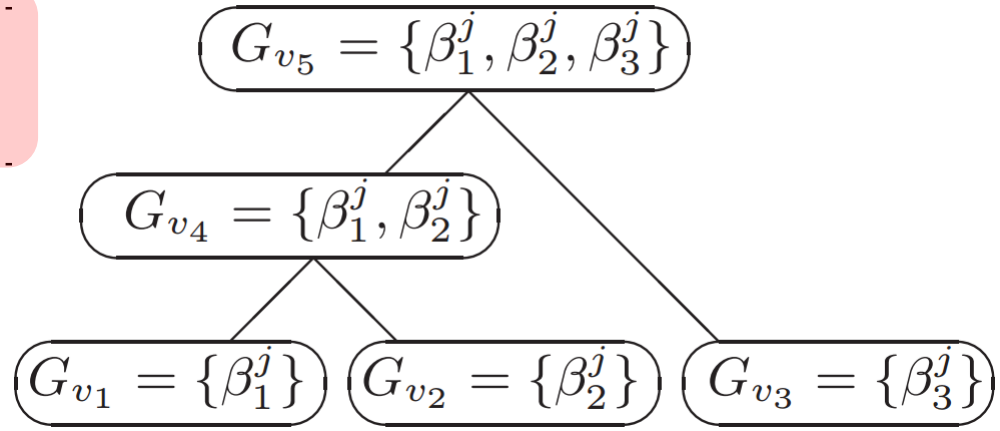
Promotes sharing and competition between tasks in hierarchical manner.

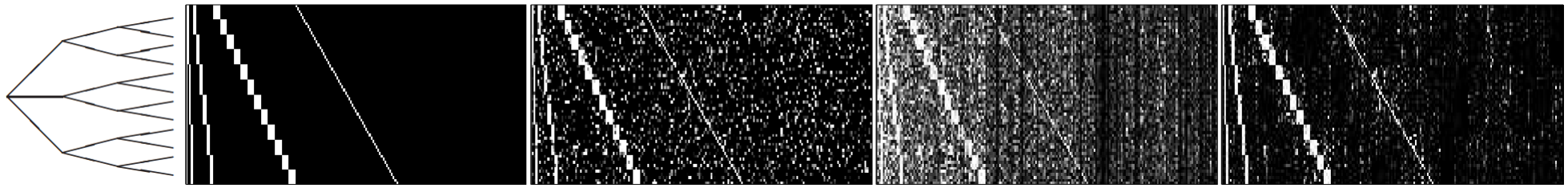
competition

sharing

$$\sum_j \left[h_2\left(\left|G_{v_4}\right| + \left|\beta_3^j\right|\right) + (1-h_2)\left(\sqrt{\left(\beta_1^j\right)^2 + \left(\beta_2^j\right)^2 + \left(\beta_3^j\right)^2}\right) \right]$$

$$\sum_j \left[h\left(\left|\beta_1^j\right| + \left|\beta_2^j\right|\right) + (1-h)\left(\sqrt{\left(\beta_1^j\right)^2 + \left(\beta_2^j\right)^2}\right) \right]$$



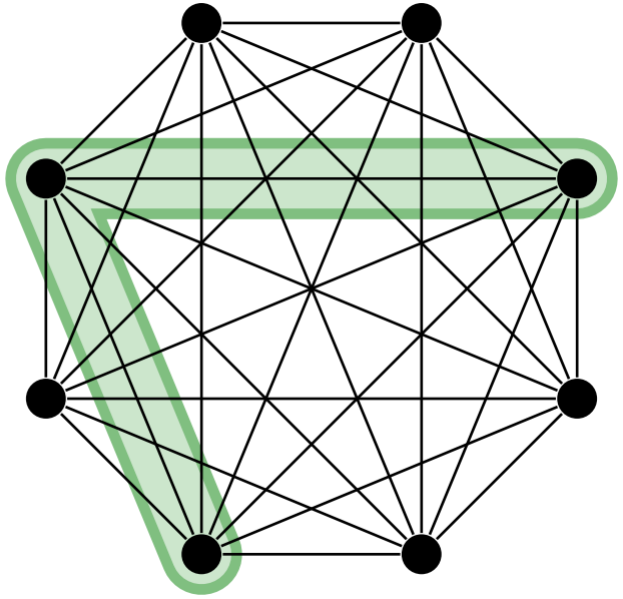


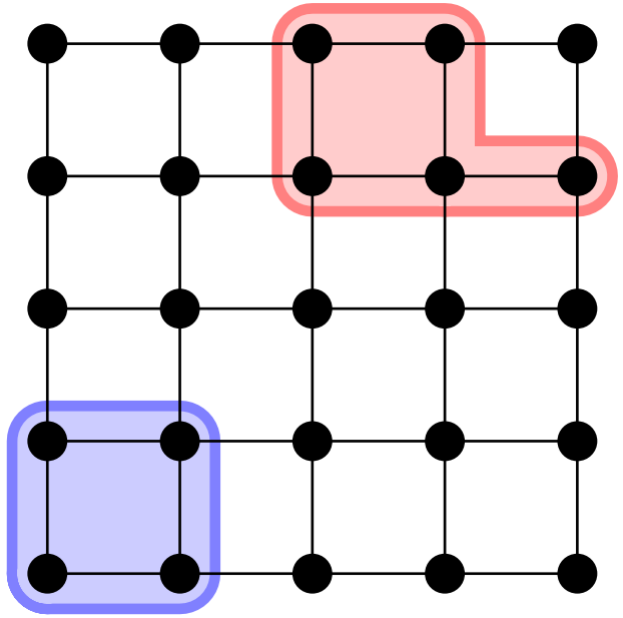
True coefficients | Lasso | Group lasso | Tree-guided Group lasso

[Kim et al.] S. Kim and E. P. Xing, Tree Guided Group Lasso for Multi-Task Regression with Structured Sparsity, ICML 2010

Graph-Structured Sparsity

Structured sparsity on a generic graph





[Hegde15] C. Hegde, P. Indyk, and L. Schmidt, A Nearly-Linear Time Framework for Graph-Structured Sparsity, ICML 2015

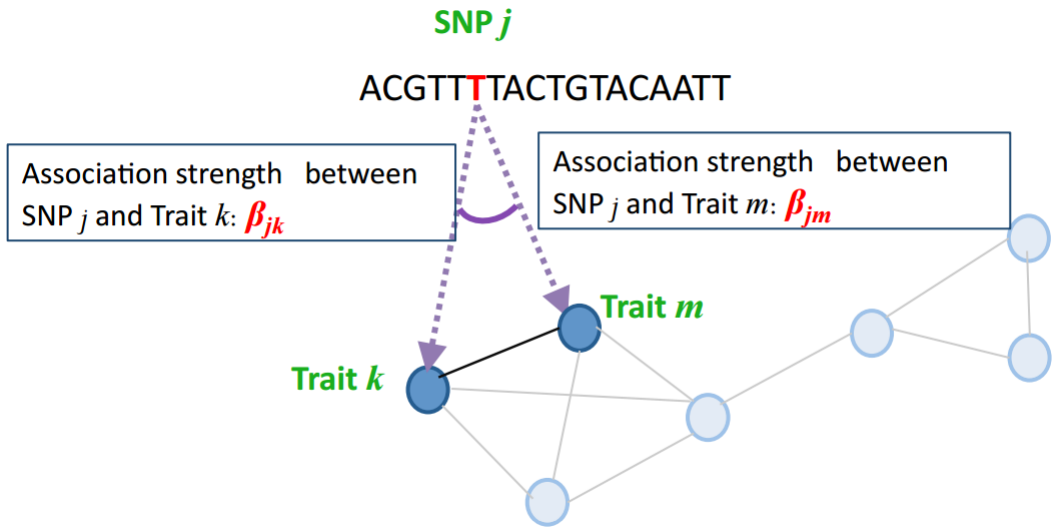
Graph-Guided Fused Lasso

Use known graph structure among the output to constrain correlated variables to have similar parameters

$$\Omega(B) = \sum_k \sum_j |\beta_{jk}| + \gamma \sum_{m,l \in E} \sum_j |\beta_{jm} - sgn(r_{ml})\beta_{jl}|$$

sparsity

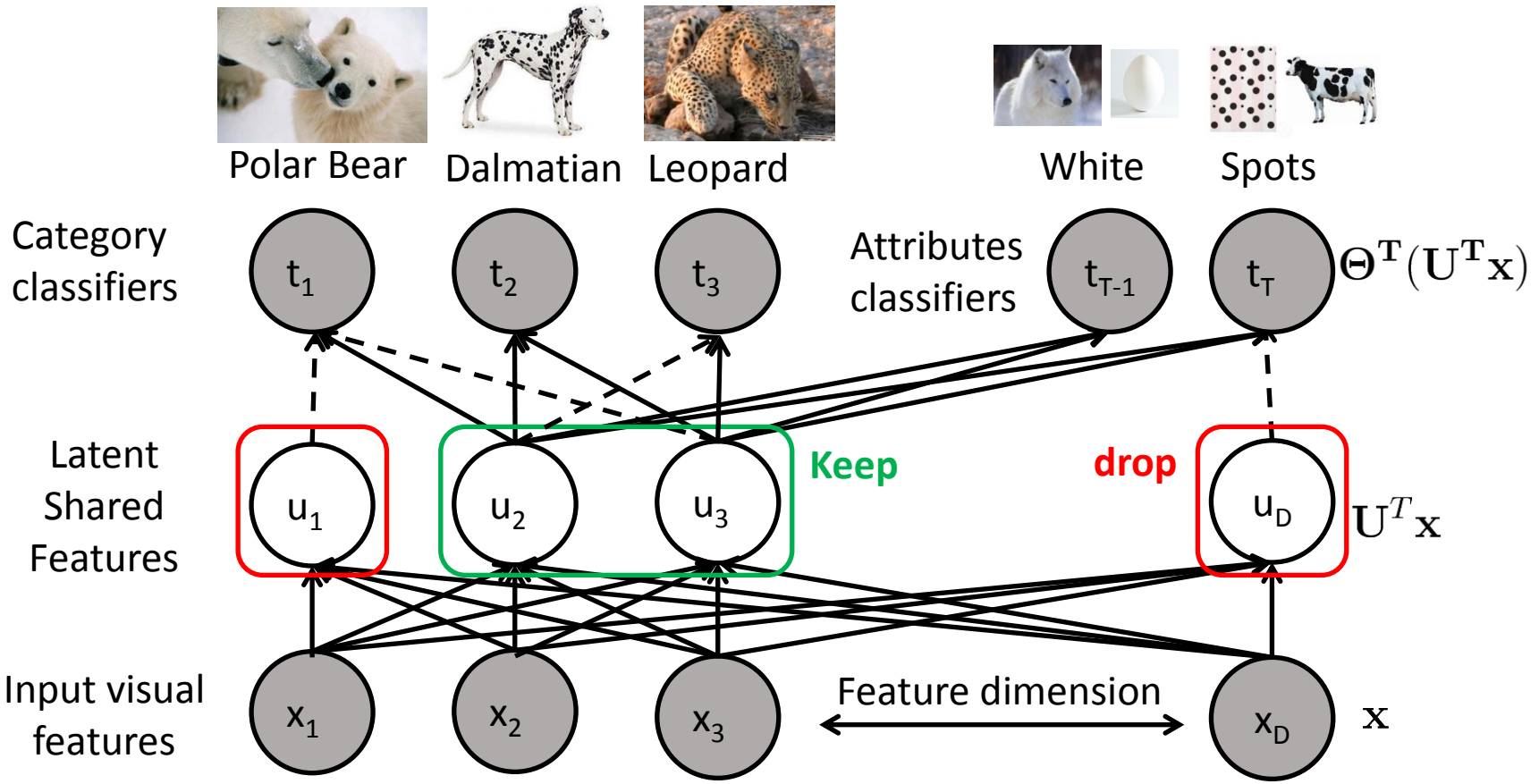
Fusion penalty



[Chen et al.] X. Chen, S. Kim, Q. Lin, J. G. Carbonell, E. P. Xing, arXiv:1005.3579

Multitask Feature Learning

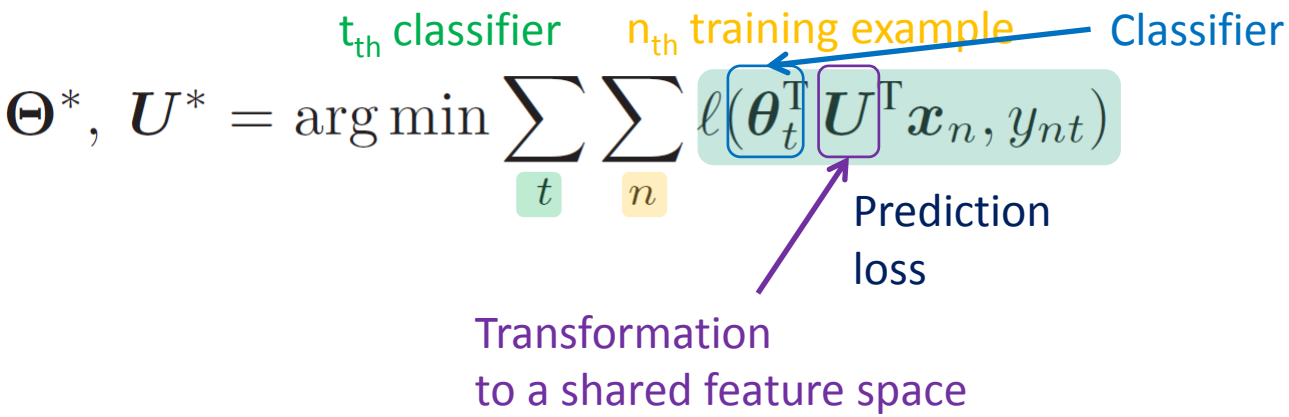
Assume that there exist some latent shared features that are shared across multiple tasks, and learn them.



[Hwang11a] S. J. Hwang, F. Sha, K. Grauman, Sharing Features between Objects and Their Attributes, CVPR 2011

Multitask Feature Learning

Given N input features x and label y_t for each task t, Simultaneously learn the transformation U and model parameter θ_t for each class t



1. Learn classifiers on the transformed features in shared feature space

2. Promote a common sparsity pattern in the new parameters

[Hwang11a] S. J. Hwang, F. Sha, K. Grauman, Sharing Features between Objects and Their Attributes, CVPR 2011

Sharing features via Sparsity Regularization

Using group sparsity regularization, enforce each learner to use features that are informative across multiple tasks.

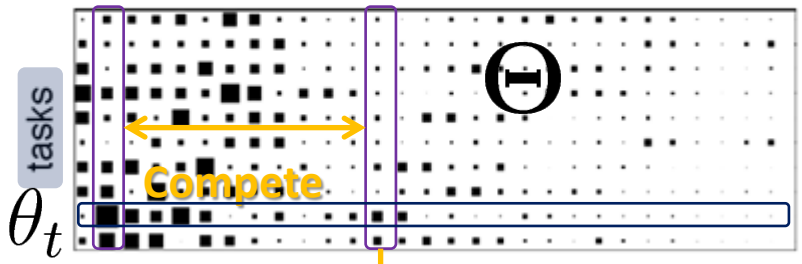
$$||\Theta||_{2,1} = \sum_{d=1}^{D} \sqrt{\sum_{t} \theta_{td}^2}$$

2-norm across tasks

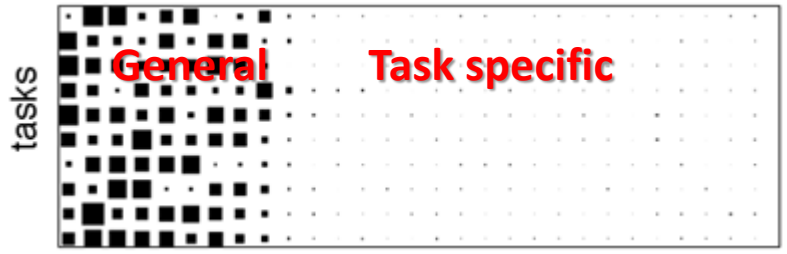
task

features

No Sharing



Sharing



[Hwang11a] S. J. Hwang, F. Sha, K. Grauman, Sharing Features between Objects and Their Attributes, CVPR 2011

Convex Multitask Feature Learning

Previous formulation is non-smooth, which makes it challenging to solve. Thus we solve an equivalent form instead.→ Replace features with a covariance matrix Ω that measures the relative effectiveness of each dimension

w_t : model parameter for task t, on **original** features.

Ω : covariance matrix

$$\boldsymbol{W}^*, \boldsymbol{\Omega}^* = \arg\min \sum_t \sum_n \ell(\boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{x}_n, y_{nt}) + \gamma \sum_t \boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{\Omega}^{-1} \boldsymbol{w}_t + \gamma\epsilon \,\mathrm{Trace}(\boldsymbol{\Omega}^{-1})$$

Model prediction loss in the **original** feature space

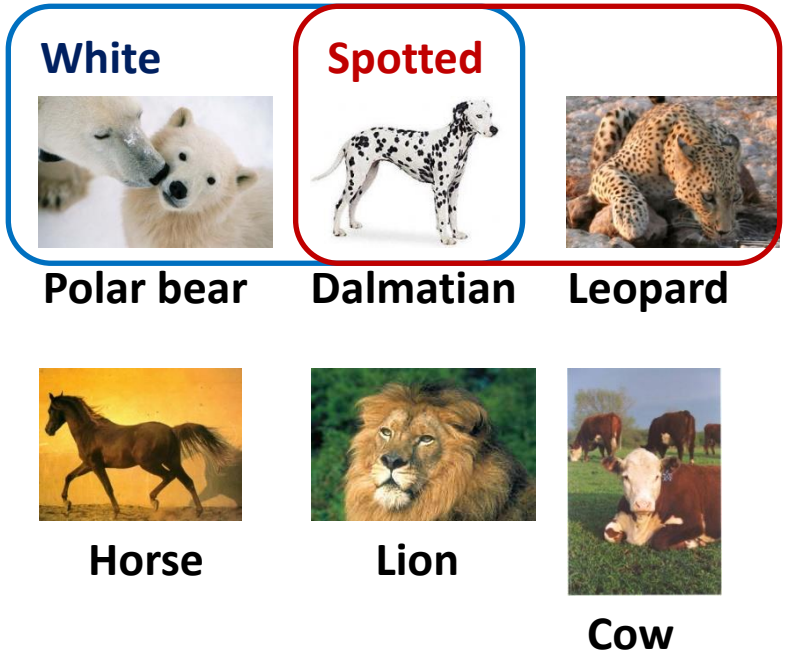
Trace norm
(sum of the diagonal entries for a PSD matrix)

[Argyriou08] A. Argyriou, T. Evgeniou and M. Pontil, Convex Multi-Task Feature Learning, Machine Learning, 2008

Multitask Feature Learning - Result

Dataset

Animals with Attributes (AWA)
30,475 images, 50 classes, 85 attributes

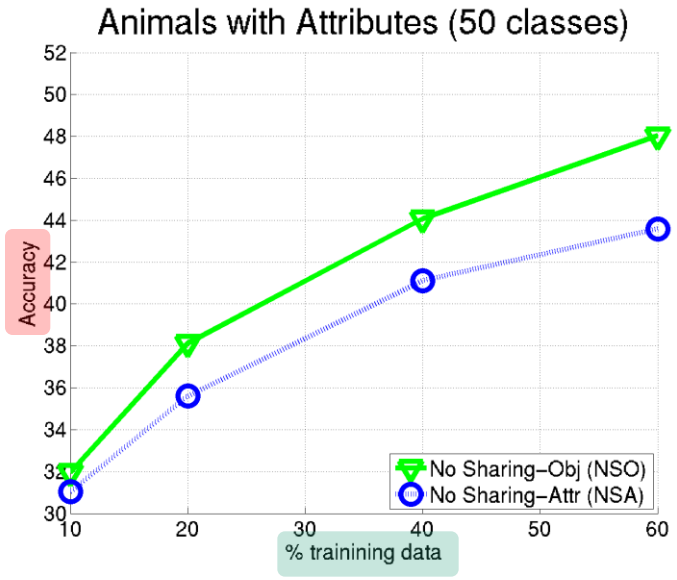


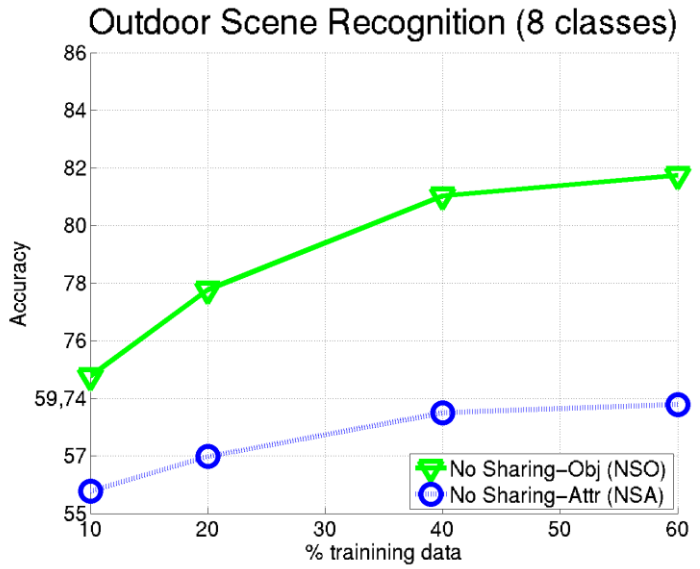
Outdoor Scene Recognition (OSR)
2,688 images, 8 classes, 6 attributes

[Hwang11a] S. J. Hwang, F. Sha, K. Grauman, Sharing Features between Objects and Their Attributes, CVPR 2011

Multitask Feature Learning - Result

No Sharing (Visual Features) > Attributes-based Prediction





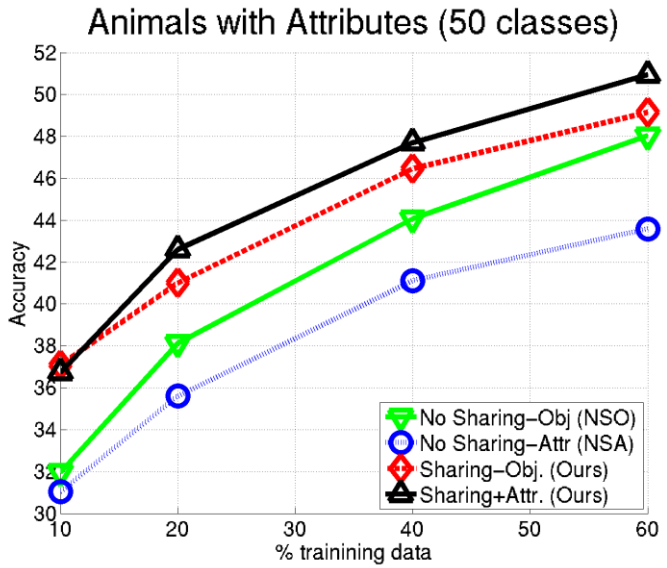
No sharing-Obj.: Independent SVMs trained on visual features

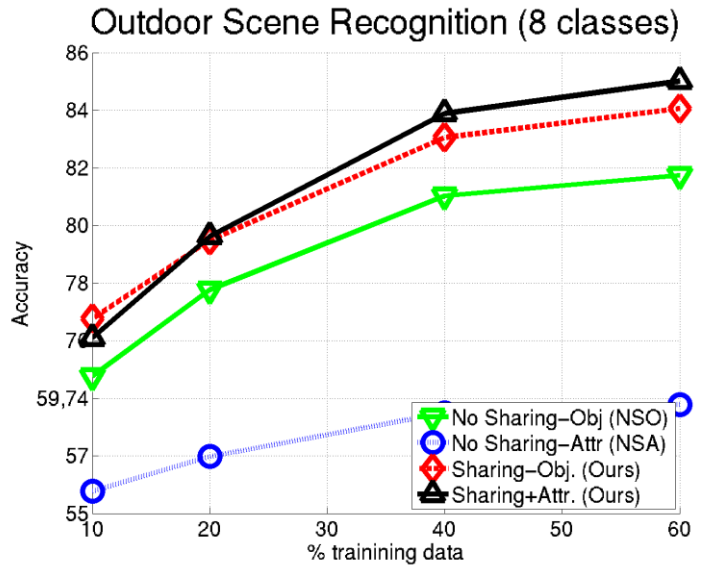
No sharing-Attr. : Object recognition on predicted attributes as in [Lampert09].

[Hwang11a] S. J. Hwang, F. Sha, K. Grauman, Sharing Features between Objects and Their Attributes, CVPR 2011

Multitask Feature Learning - Result

Shared Features (Object & Attributes) > Shared Features (Objects) > No Sharing





Sharing-Obj.: Multitask Feature Learning on object classifiers

Sharing-Attr. : Multitask Feature Learning on object + attribute classifiers

[Hwang11a] S. J. Hwang, F. Sha, K. Grauman, Sharing Features between Objects and Their Attributes, CVPR 2011

Multitask Feature Learning - Result

Predicted object categories Red: incorrect prediction

More robustness to background clutter from features refined with attributes

No Sharing Dolphin

Ours Grizzly Bear

No Sharing Polar Bear

Ours Dalmatian

Even when our method fails, it often makes more semantically “close” predictions.

No Sharing Giant Panda

Ours Rhinoceros

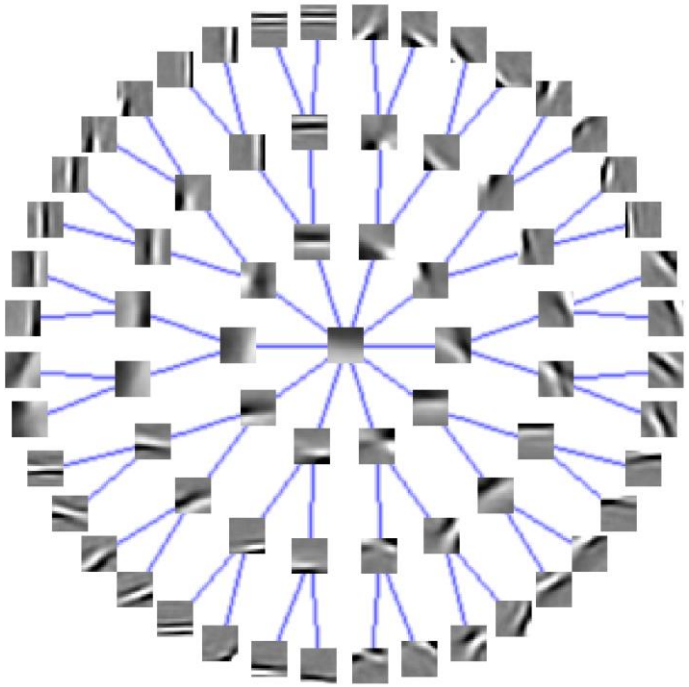
No Sharing Cow

Ours Wolf

[Hwang11a] S. J. Hwang, F. Sha, K. Grauman, Sharing Features between Objects and Their Attributes, CVPR 2011

Image Inpainting with Hierarchical Sparse Coding

Use hierarchical sparsity inducing-norm to reconstruct an input image with a hierarchy of image patches



[Jenatton11a] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, Proximal Methods for Sparse Hierarchical Dictionary Learning, ICML 2010

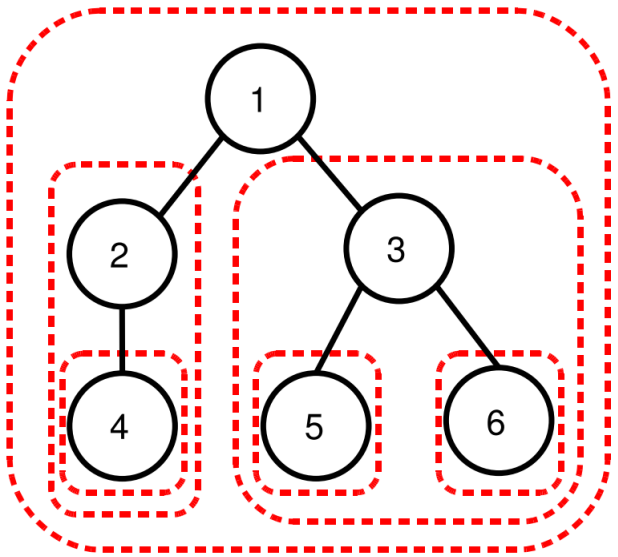
Hierarchical Sparse Coding

Use proximal operator to solve for the hierarchical sparsity-inducing norm

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}\in\mathcal{A}} \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2}||\mathbf{x}^i - \mathbf{D}\boldsymbol{\alpha}^i||_2^2 + \lambda\Omega(\boldsymbol{\alpha}^i)\right]$$

$$\Omega(\boldsymbol{\alpha}) \triangleq \sum_{g\in\mathcal{G}} w_g \|\boldsymbol{\alpha}_{|g}\|$$

$$\min_{\mathbf{v}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{u}-\mathbf{v}\|_2^2 + \lambda\Omega(\mathbf{v})$$



Problem – No closed form solution exists for group lasso with overlapping groups

[Jenatton11a] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, Proximal Methods for Sparse Hierarchical Dictionary Learning, ICML 2010

Hierarchical Sparse Coding

Use primal-dual approach to solve for the dual problem

Primal

$$\min_{\mathbf{v}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{u}-\mathbf{v}\|_2^2 + \lambda\Omega(\mathbf{v})$$

Dual

$$\max_{\boldsymbol{\xi}\in\mathbb{R}^{p\times|\mathcal{G}|}} -\frac{1}{2}\left(\|\mathbf{u}-\sum_{g\in\mathcal{G}}\boldsymbol{\xi}^g\|_2^2 - \|\mathbf{u}\|_2^2\right)$$

$$s.t.\ \forall g\in\mathcal{G},\ \|\boldsymbol{\xi}^g\|_* \leq \lambda w_g \text{ and } \boldsymbol{\xi}_j^g = 0 \text{ if } j\notin g$$

Algorithm 1 Block coordinate ascent in the dual

Inputs: $\mathbf{u}\in\mathbb{R}^p$ and set of groups $\mathcal{G}$.
Outputs: $(\mathbf{v},\boldsymbol{\xi})$ (primal-dual solutions).
Initialization: $\mathbf{v}=\mathbf{u}, \boldsymbol{\xi}=0$.
while (*maximum number of iterations not reached*) **do**
  for $g\in\mathcal{G}$ **do**
    $\mathbf{v}\leftarrow\mathbf{u}-\sum_{h\neq g}\boldsymbol{\xi}^h$.
    $\boldsymbol{\xi}^g\leftarrow\Pi^*_{\lambda w_g}(\mathbf{v}_{|g})$.
  end for
end while
$\mathbf{v}\leftarrow\mathbf{u}-\sum_{g\in\mathcal{G}}\boldsymbol{\xi}^g$.

Convex constraints for each vector ξ^g are separable

- Can be efficiently solved using block-coordinate ascent

[Jenatton11a] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, Proximal Methods for Sparse Hierarchical Dictionary Learning, ICML 2010

Image Inpainting Results

Hierarchical dictionary learning yields much less reconstruction noise compared to flat dictionary learning with flat l1-regularization.

L1

Tree-sparsity

Noise	50%	90%
Flat	19.3	72.1
Hierarchical	18.6	65.9

[Jenatton11a] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, Proximal Methods for Sparse Hierarchical Dictionary Learning, ICML 2010

Decorrelating Visual Semantic Attributes

Problem: difficult to distinguish between co-occurring attributes

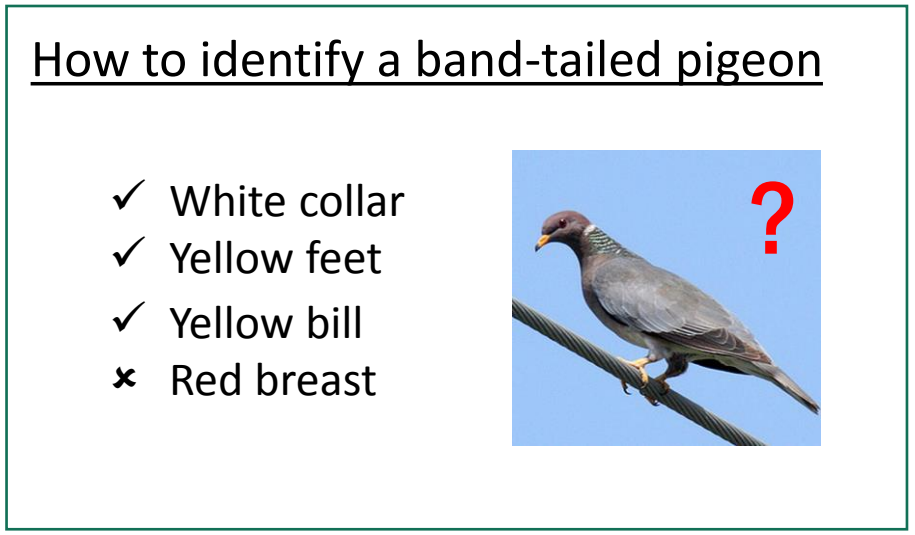
✓ Forest animal? ✓ Brown? ✓ Has ears? ✗ ✗ Combinations?

[Jayaraman et al.] D. Jayaraman, F. Sha, and K. Grauman, Decorrelating Visual Attributes by Resisting Urge to Share, CVPR 2014

Decorrelating Visual Semantic Attributes

Motivation: learning correct attributes is crucial for applications such as image search and zero-shot learning.

Image search

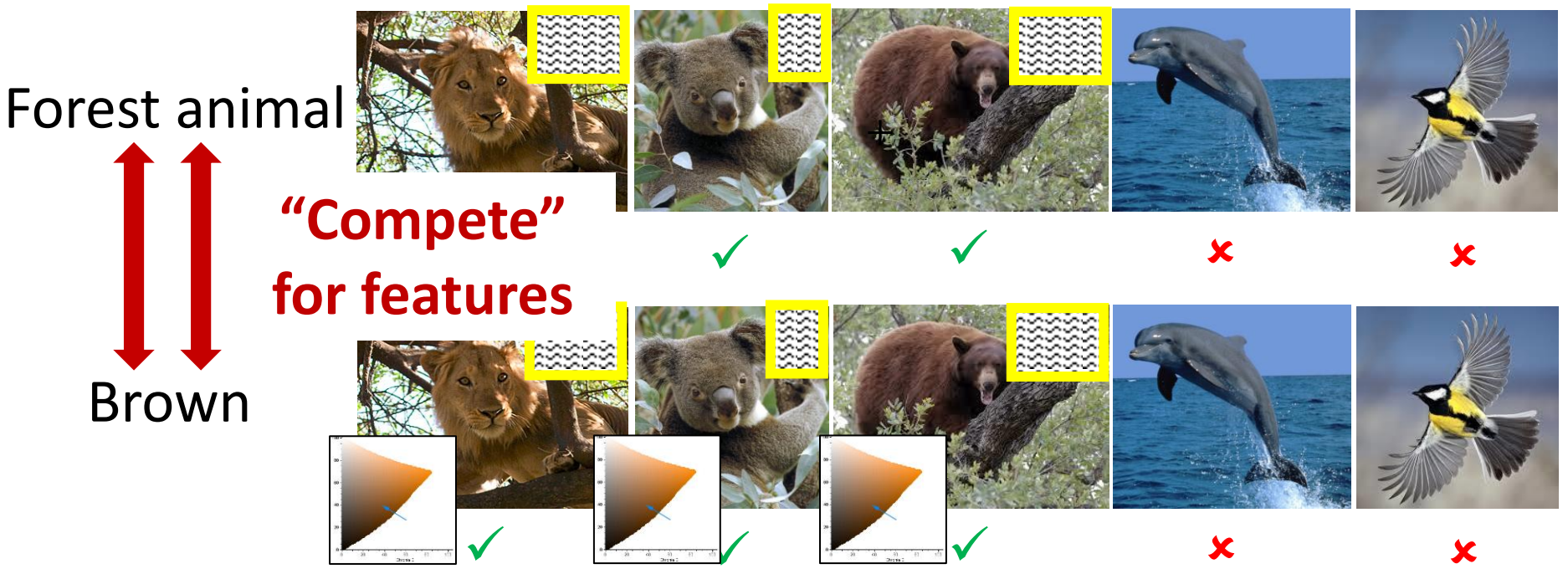


Zero-shot learning

[Jayaraman et al.] D. Jayaraman, F. Sha, and K. Grauman, Decorrelating Visual Attributes by Resisting Urge to Share, CVPR 2014

Decorrelating Visual Semantic Attributes

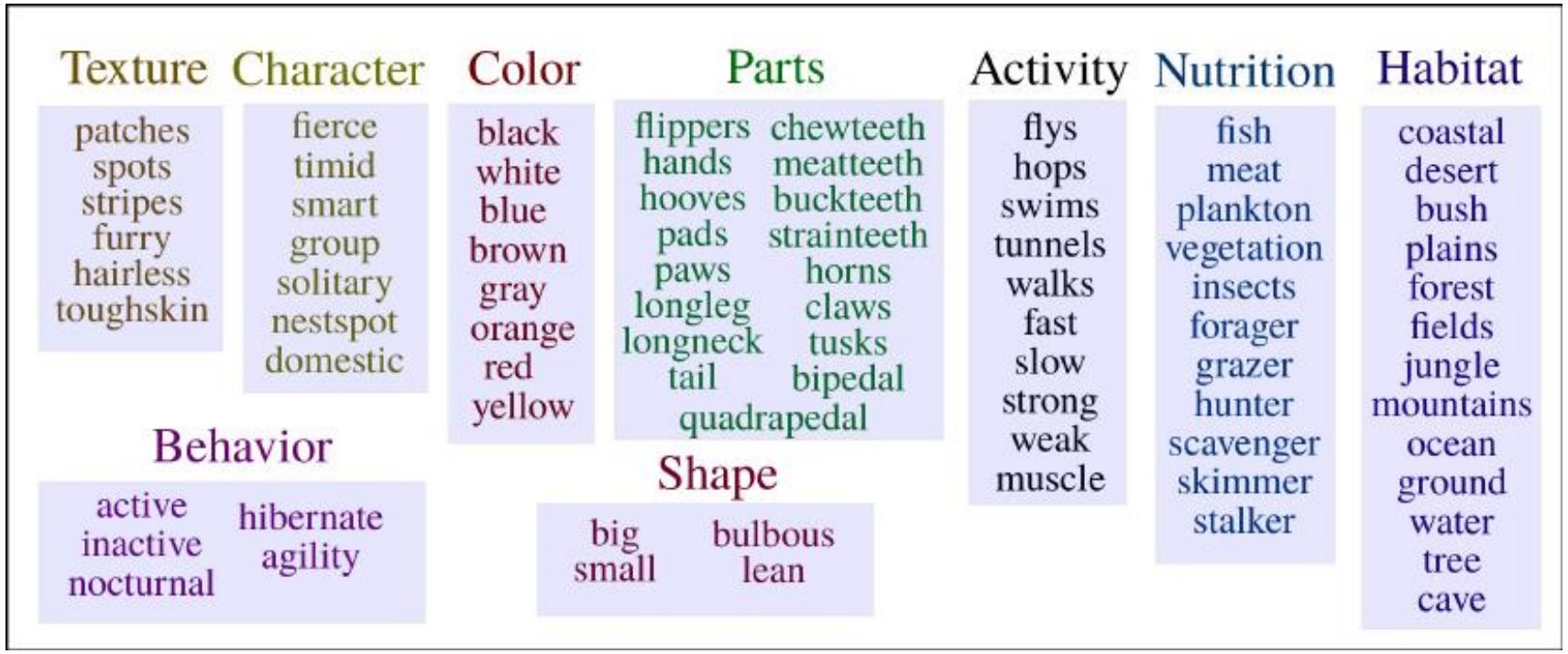
Solution: promote competition between



[Jayaraman et al.] D. Jayaraman, F. Sha, and K. Grauman, Decorrelating Visual Attributes by Resisting Urge to Share, CVPR 2014

Decorrelating Visual Semantic Attributes

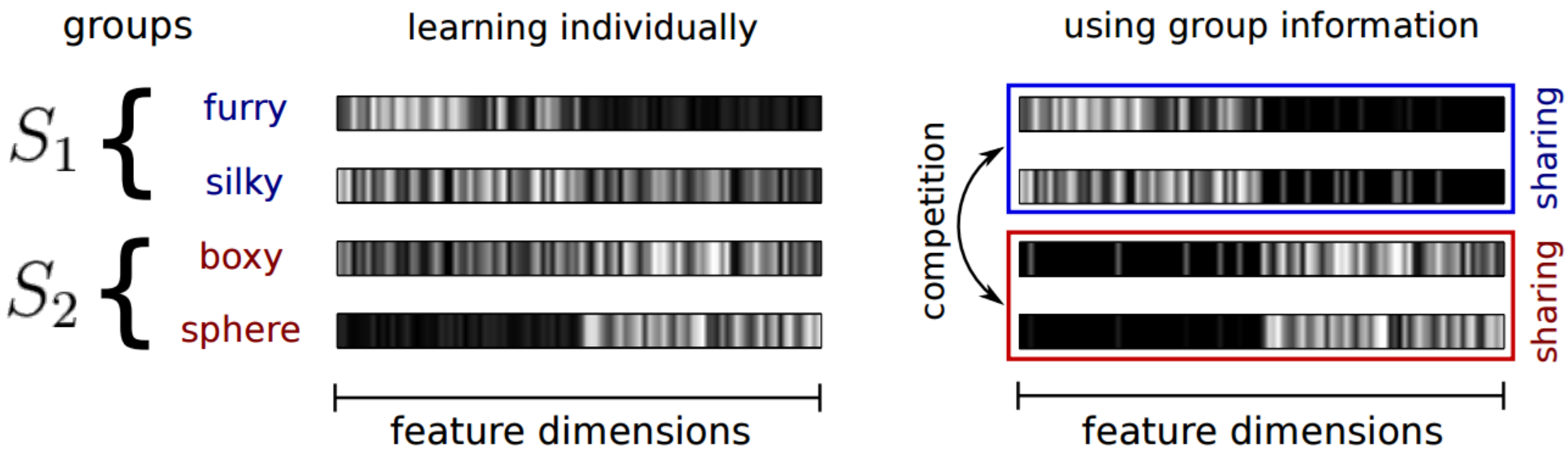
Attributes can be grouped into multiple semantic categories



[Jayaraman et al.] D. Jayaraman, F. Sha, and K. Grauman, Decorrelating Visual Attributes by Resisting Urge to Share, CVPR 2014

Decorrelating Visual Semantic Attributes

Promote sharing between attributes in the same group, while promoting competition for features in different groups



$$\underset{W}{\operatorname{argmin}} \; L(W|X,Y) + \sum_d \sum_l \|w_d^{S_l}\|_2$$

Classification loss

Promotes competition for features support between different groups

Promotes feature sharing between attributes within a same group

[Jayaraman et al.] D. Jayaraman, F. Sha, and K. Grauman, Decorrelating Visual Attributes by Resisting Urge to Share, CVPR 2014

Decorrelating Visual Semantic Attributes

Decorrelation model predicts attributes much better in unusual cases.

Not brown underparts

No eye

Not boxy

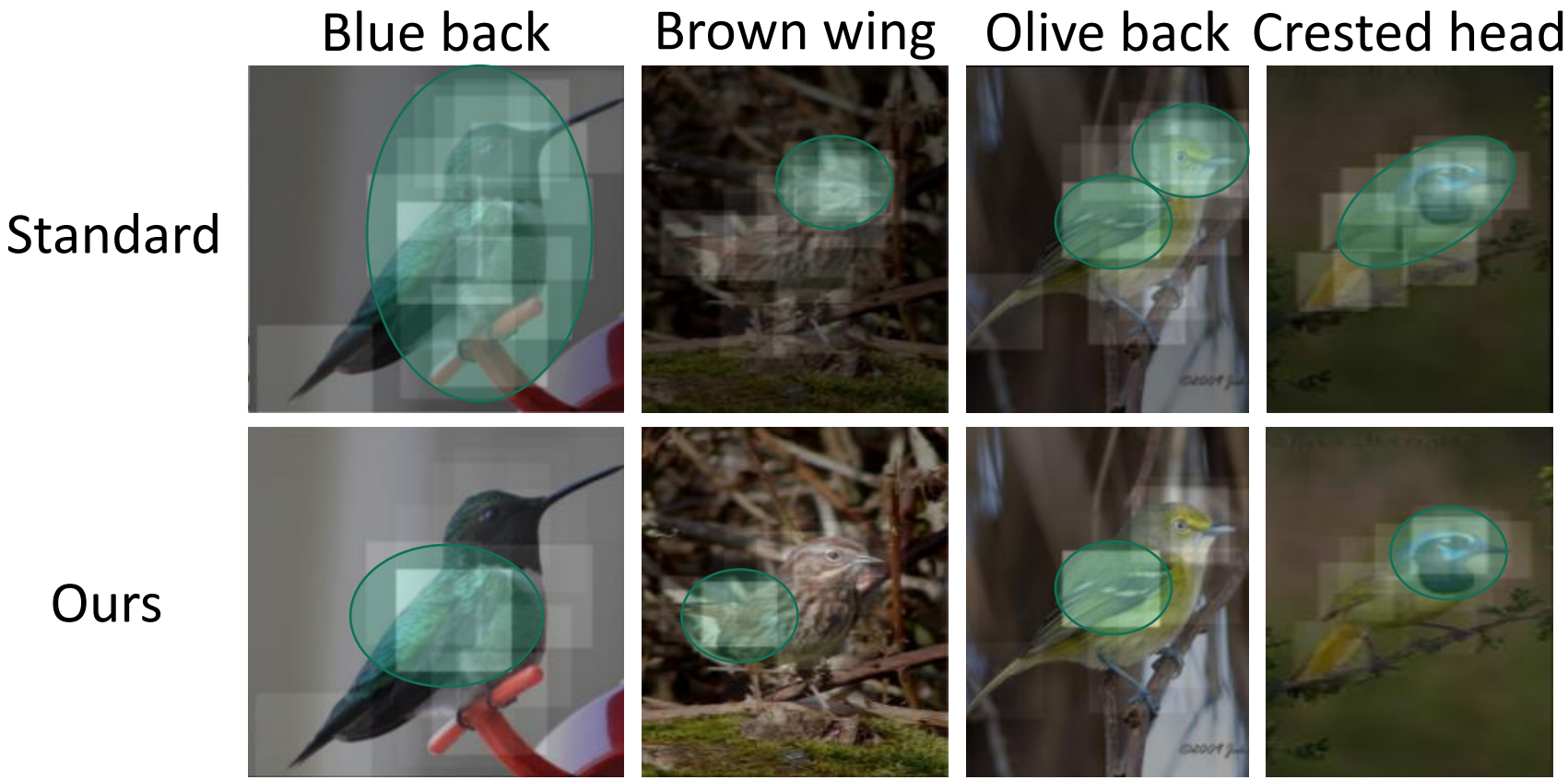
No mouth

No ear

[Jayaraman et al.] D. Jayaraman, F. Sha, and K. Grauman, Decorrelating Visual Attributes by Resisting Urge to Share, CVPR 2014

Decorrelating Visual Semantic Attributes

This model can also accurately localize the part-based attributes.



[Jayaraman et al.] D. Jayaraman, F. Sha, and K. Grauman, Decorrelating Visual Attributes by Resisting Urge to Share, CVPR 2014

SPAMS (SPArse Modeling Software)

A popular optimization toolbox for solving most of the methods introduced in this presentation

http://spams-devel.gforge.inria.fr/

- Implements OMP, Lasso, LARS, coordinate descent and proximal methods for l1-regularization.

- Provides proximal toolbox for solving l2/l1-reg. , sparse group lasso, tree-structured regularization, and structured sparsity with overlapping groups.

[Mairal et al.] J. Mairal, F. Bach, and J. Ponce, Sparse Modeling for Image and Vision Processing, Foundations and Trends in Computer Graphics and Vision, 2014

Summary

Structured sparsity is a kind of sparsity that prefers ***certain structure*** among the variables, based on some ***prior knowledge***.

There are various types of structured sparsity, including ***group sparsity***, ***block sparsity***, ***tree-structured sparsity***, and ***graph-based sparsity***. Structured sparsity is often enforced as ***mixed norm*** regularization, such as (2,1)-norm

Structured sparsity is useful for multiple applications, including ***multi-task learning***, and hierarchical dictionary learning for ***image denoising***.