Multi-label prediction via compressed sensing

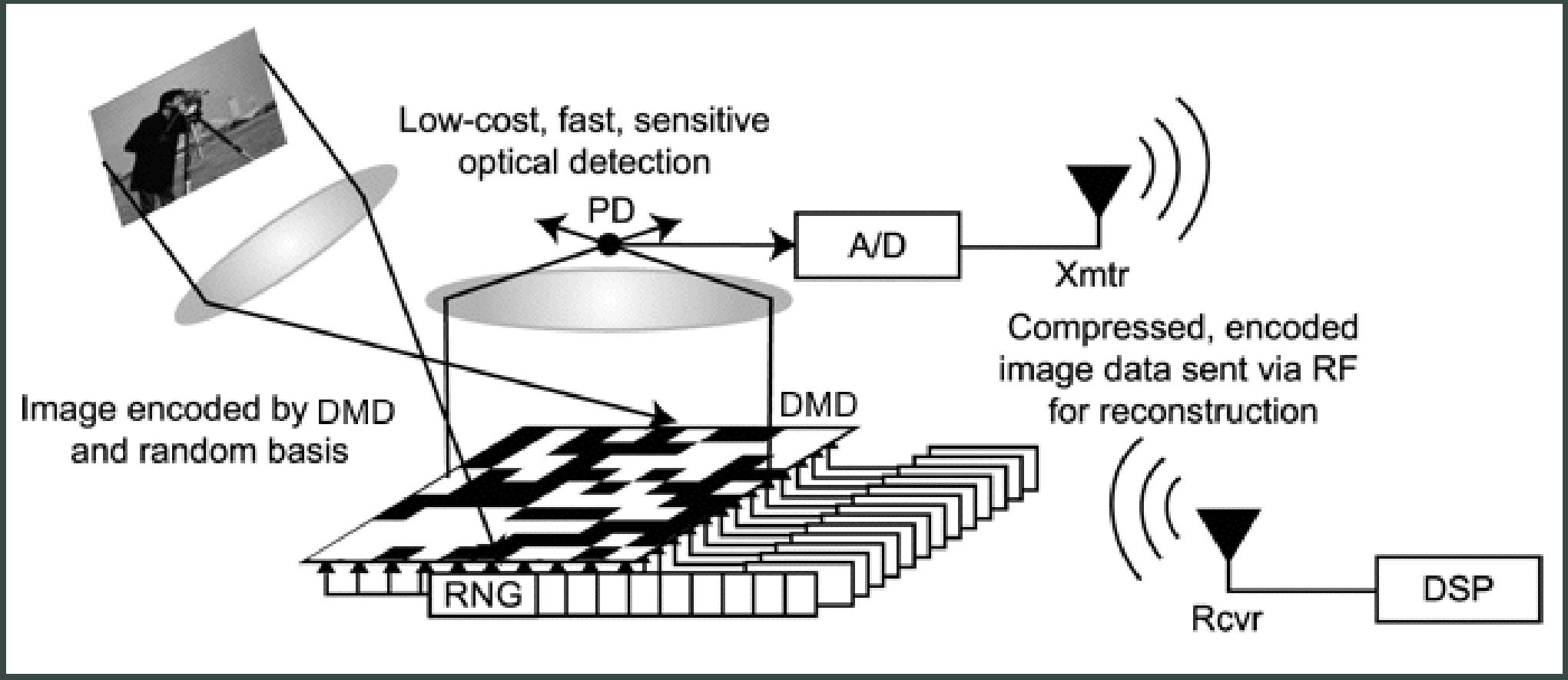
Daniel Hsu
UCSD

Sham M. Kakade
UPenn

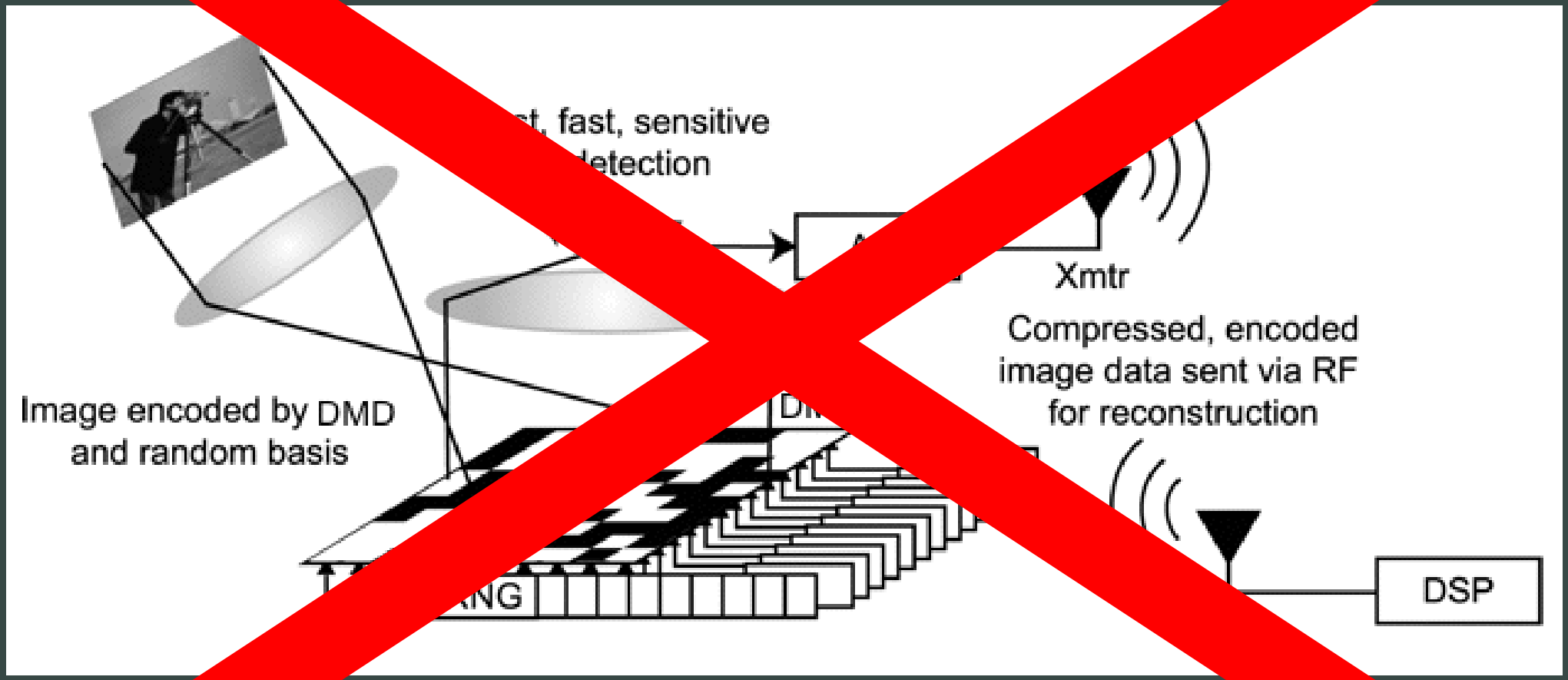
John Langford
Yahoo!

Tong Zhang
Rutgers

Single-pixel camera



Single-pixel camera



Outline

1. Multi-label prediction (introduction)
2. The reduction
3. Compression & reconstruction
4. Results

Part 1
Multi-label prediction (introduction)

Multi-label prediction

Supervised learning in which more than one class may be correct.

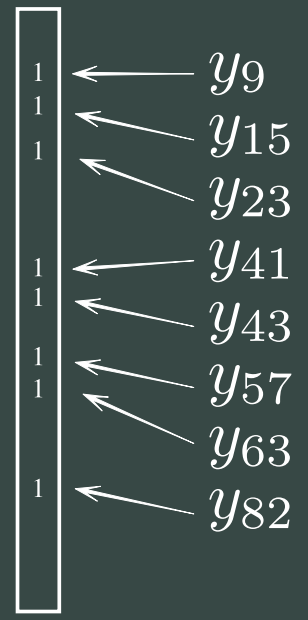
$$F : \mathcal{X} \to \{0,1\}^d \qquad (d = |\mathcal{Y}|)$$

e.g.

$\mapsto$ { Alyssa, David, James, Jim, Joe, Mark, Rahm, Robert }

$x \in \mathcal{X}$

$y \in \{0,1\}^d$



General abstract problem

Supervised learning of d real-value prediction tasks.

$$F : \mathcal{X} \to \mathbb{R}^d$$

Challenge: output dimension d is typically very large (*e.g.* several thousands).

General abstract problem

Supervised learning of d real-value prediction tasks.

$$F : \mathcal{X} \to \mathbb{R}^d$$

Challenge: output dimension d is typically very large (*e.g.* several thousands).

However, in many cases, the output vector is *sparse* (*i.e.* only a few entries are non-zero).

General abstract problem

Supervised learning of d real-value prediction tasks.

$$F : \mathcal{X} \to \mathbb{R}^d$$

Challenge: output dimension d is typically very large (*e.g.* several thousands).

However, in many cases, the output vector is *sparse* (*i.e.* only a few entries are non-zero).

> *e.g.* although the total number of people is large, only a few people are depicted in each image.

Previous work and context

One-Against-All: the simplest general approach
to multi-label prediction:

- learn d binary predictors, one for each label (class)
- highly inefficient when d is large

Previous work and context

One-Against-All: the simplest general approach
to multi-label prediction:

- learn d binary predictors, one for each label (class)
- highly inefficient when d is large

Many domain-specific solutions for coping
with large output spaces:

- leverage dependency structure (*e.g.* CRF)
- model relationships (*e.g.* hierarchy) among classes

Previous work and context

One-Against-All: the simplest general approach
to multi-label prediction:

- learn d binary predictors, one for each label (class)
- highly inefficient when d is large

Many domain-specific solutions for coping
with large output spaces:

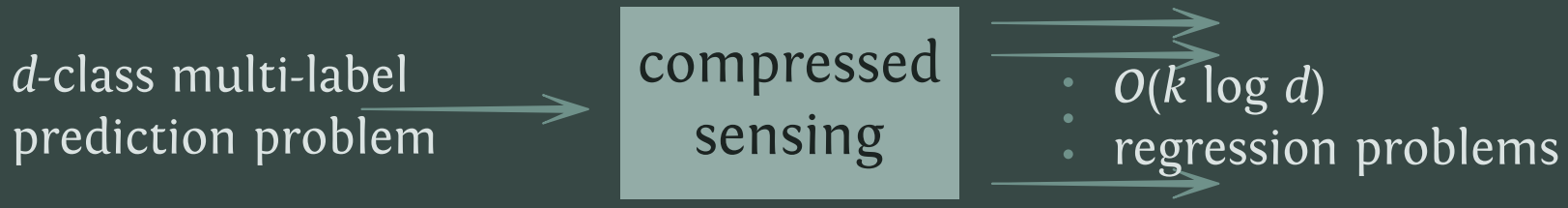
- leverage dependency structure (*e.g.* CRF)
- model relationships (*e.g.* hierarchy) among classes

Is there a general method for exploiting *output sparsity*?

Can we get away with $\ll d$ predictors, *e.g.* $O(\log d)$?

Our work

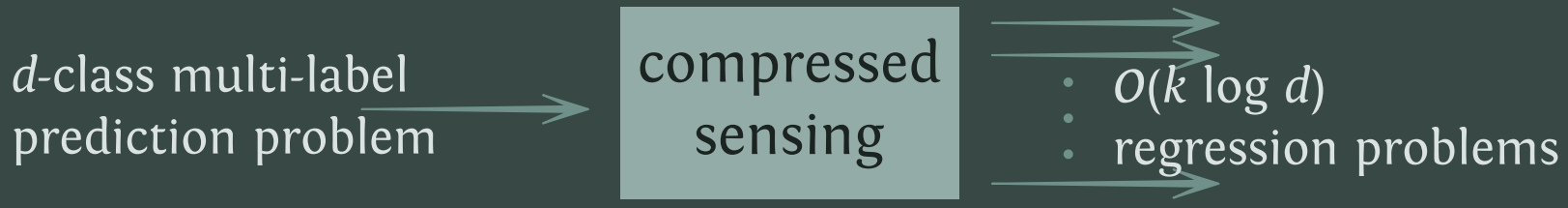
Develop a *learning reduction* method for
multi-label prediction that exploits output sparsity.



k is the sparsity level of the output.

Our work

Develop a *learning reduction* method for
multi-label prediction that exploits output sparsity.



k is the sparsity level of the output.

- Focus is on high-dimensional outputs
- Exploiting output sparsity gives huge savings:

$$O(k \log d) \quad \text{vs} \quad O(d)$$

Part 2
The reduction

Problem setup

Goal: learn a predictor $F : \mathcal{X} \to \mathbb{R}^d$
from labeled training examples $\{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathbb{R}^d$
drawn iid from a fixed distribution.

We hope that $\mathbb{E}[y|x]$ is close to being *k*-sparse
(*i.e.* have at most *k* non-zero entries, for $k \ll d$),
but we want to be *agnostic* w.r.t. this condition.

Performance measure: mean squared error

$$\mathbb{E}_x \|F(x) - \mathbb{E}[y|x]\|_2^2$$

Learning reduction

Two components:
Linear compression function A, Reconstruction algorithm R

Learning reduction

Two components:
Linear compression function A, Reconstruction algorithm R

Main idea:
Instead of predicting the label y,
predict the compressed label Ay.

Learning reduction

Two components:
Linear compression function A, Reconstruction algorithm R

Main idea:
Instead of predicting the label y,
predict the compressed label Ay.

1. Training:
   Learn to predict compressed label Ay from x.
2. Prediction:
   Reconstruct y from prediction of Ay.

Learning reduction

Two components:
Linear compression function A, Reconstruction algorithm R

Main idea:
Instead of predicting the label y,
predict the compressed label Ay.

1. Training:
   Learn to predict compressed label Ay from x.
2. Prediction:
   Reconstruct y from prediction of Ay.

Similar to ECOC (Dietterich & Bakiri, 1995),
but we exploit output sparsity.

Reduction - training

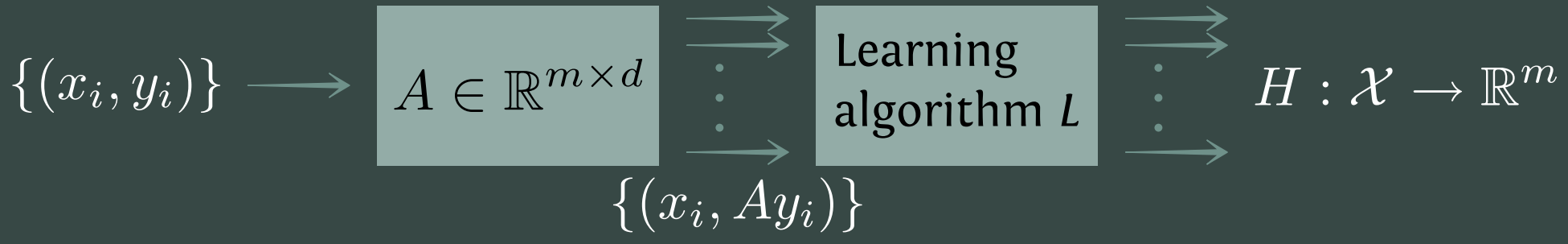
Learn to predict compressed labels $Ay \in \mathbb{R}^m, \quad m \ll d.$

1. Compress the training labels with m x d matrix A:

$$\{(x_i, y_i)\} \;\mapsto\; \{(x_i, Ay_i)\}$$

2. Learn m predictors using the compressed labels:

$$\{(x_i, Ay_i)\} \;\mapsto\; H : \mathcal{X} \to \mathbb{R}^m$$



Learning should try to minimize $\mathbb{E}_x \|H(x) - \mathbb{E}[Ay|x]\|_2^2$

Reduction - prediction

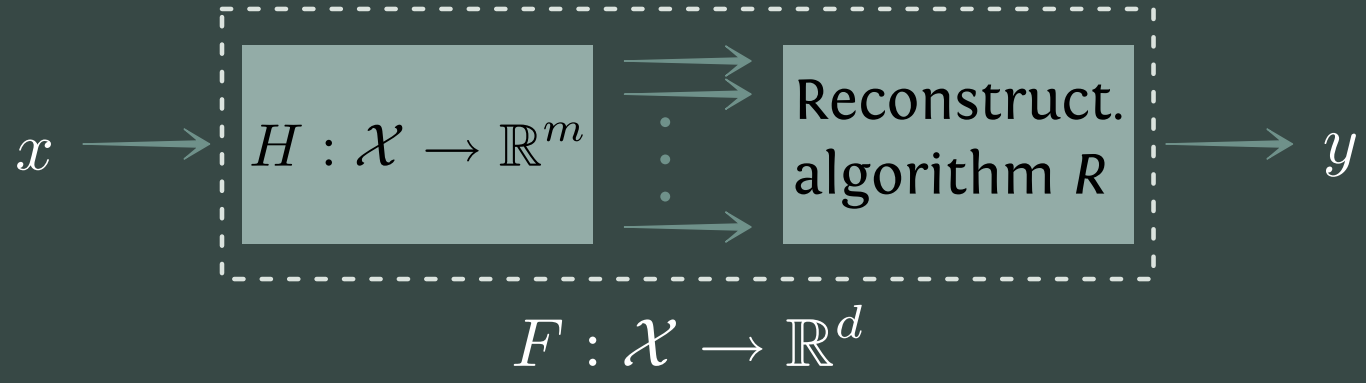
Decode predictions of compressed labels Ay.

1. Upon input x, predict compressed label $H(x)$:

$$x \mapsto H(x) \in \mathbb{R}^m$$

2. Reconstruct $O(k)$-sparse label y from $H(x)$:

$$H(x) \mapsto \mathsf{R}(H(x)) = y \in \mathbb{R}^d$$



Reduction - prediction

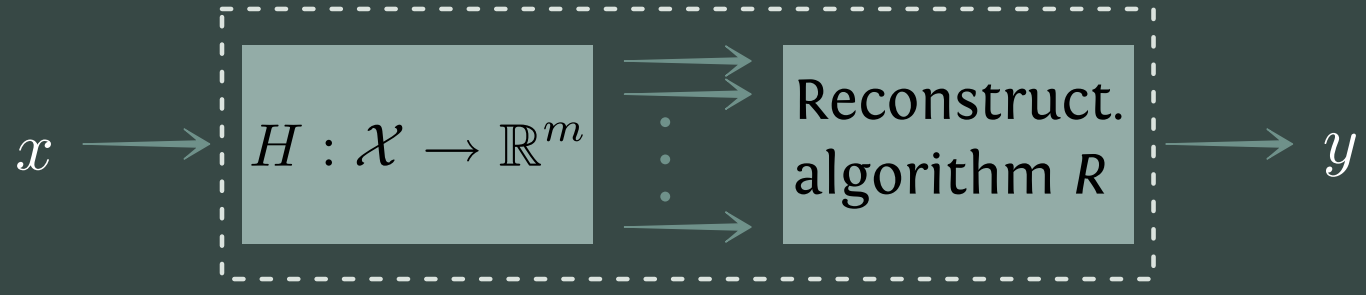
Decode predictions of compressed labels Ay.

1. Upon input x, predict compressed label $H(x)$:

$$x \mapsto H(x) \in \mathbb{R}^m$$

2. Reconstruct $O(k)$-sparse label y from $H(x)$:

$$H(x) \mapsto \mathsf{R}(H(x)) = y \in \mathbb{R}^d$$



Reconstruction finds sparse y s.t. Ay approximates $H(x)$.

Part 3
Compression & reconstruction

How to choose the components?

How to choose compression function A
& reconstruction algorithm R?

How to choose the components?

How to choose compression function A
& reconstruction algorithm R?

Compressed sensing 101:

For all k-sparse y, we can compress to $m = O(k \log d)$ dimensions so that perfect reconstruction of y is easy.

Component requirements

The compression fn. A & reconstruction alg. R should have the following property:

> Upon input c (the predicted compressed label), if there exists k-sparse y s.t. Ay is close to c, then $R(c)$ is $O(k)$-sparse and close to y:

$$\|y - \mathsf{R}(c)\|_2^2 \leq C_1 \cdot \|Ay - c\|_2^2.$$

(approx. error)

*Also handle case when y is approximately k-sparse.

Compression and reconstruction

Examples of valid components:

- Compression function: random m x d matrix!
  (Mendelson *et al*, 2008; Rudelson & Vershynin, 2006)

  e.g. all entries iid Gaussians N(0,1/m) with $m = O(k \log d)$.

- Reconstruction algorithm: many greedy/iterative sparse recovery algorithms.

  e.g. Compressive Sampling Matching Pursuit (Needell & Tropp, 2007)
  Forward-Backward Greedy (Zhang, 2008),
  Orthogonal Matching Pursuit (Mallat & Zhang, 1993) [almost].

- Have to analyze the approximation error,
  because typically don't have Ay exactly, due to prediction error.

Part 4
Results

Regret transform bound

Theorem (regret transform):

Let A and R be a valid compression func. / reconstruct. alg. pair.

Let $F(\cdot) = R(H(\cdot))$ (*i.e.* composition of R and H). Then:

$$\mathbb{E}_x \|F(x) - \mathbb{E}[y|x]\|_2^2 \leq C_1 \cdot \underbrace{\mathbb{E}_x \|H(x) - \mathbb{E}[Ay|x]\|_2^2}$$

↑

average MSE from $m = O(k \log d)$ subproblems

Regret transform bound

Theorem (regret transform):
Let A and R be a valid compression func. / reconstruct. alg. pair.

Let $F(\cdot) = R(H(\cdot))$ (*i.e.* composition of R and H). Then:

$$\mathbb{E}_x \|F(x) - \mathbb{E}[y|x]\|_2^2 \leq C_1 \cdot \underbrace{\mathbb{E}_x \|H(x) - \mathbb{E}[Ay|x]\|_2^2}$$

average MSE from $m = O(k \log d)$ subproblems

But aren't the subproblems more difficult to solve than the original problem?

Linear prediction

Theorem (reverse regret transform):

If there is a linear predictor with MSE ε for original problem,
then there are linear predictors with MSE $(1+o(1))\varepsilon$ for subproblems.

Experimental validation

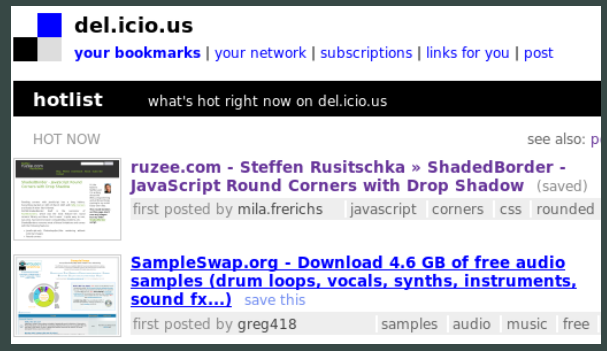
Experiment #1:
68k images from ESP game
(von Ahn & Dabbish, 2004)

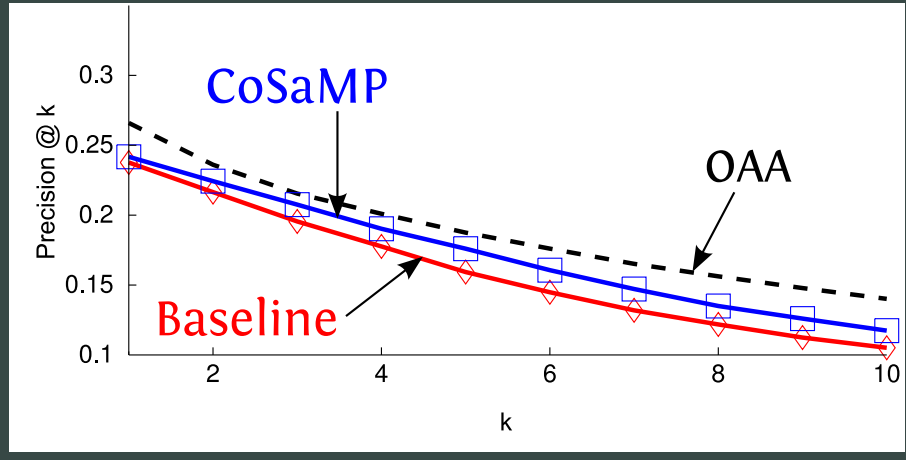
Experiment #2:
16k webpages on del.icio.us
(curated by Tsoumakas et al, 2008)

d = 1000 most common tags assigned by players/users

Compress to m = 200 subproblems

See poster for details!





Recap

- Efficient reduction for multi-label prediction
  in the presence of *output sparsity*.
  - # subproblems *logarithmic* in d (number of classes).

- Regret transforms robustly from subproblems
  to original problem (and vice versa for linear prediction).

- Empirical validation on (somewhat) large output spaces,
  outperforms baseline ECOC methods.

Thanks!

- Thanks to Andy Cotter for help with ESP game data.
- Thank you for listening!