Convex Methods for Representation Learning

Dale Schuurmans
University of Alberta

ACML 2012 Singapore
Part 0: Representation Learning
Representation Learning

Suppress “irrelevant” details
- invariants

Express “important” aspects
- underlying causes
- semantic features
- important abstract categories
Representation Learning
Representation Learning

What should representation be like?

experienced features
↓ ↓
representation features
↓ ↓

... guy ...
... bottle ...
... fizzing ...
duck!

360 × 480 × 3 = 172800 × 3 = 518400 experienced features
Representation Learning

How can meaningful features be identified?
- reduce dimensionality?
- expand dimensionality? (over-complete, sparse?)
- independent feature separation?
- lossy?
- not lossy?
The Brain: A Representation Machine
The Brain: A Representation Machine

Brain
- \( \sim 10^9 \) (1B) sensors
- \( \sim 10^7 \) (10M) nerves
- \( \sim 10^{11} \) (100B) neurons
- \( \sim 10^{14} \) (100T) synapses

Vision
- \( \sim 10^8 \) (100M) sensors
- \( \sim 10^6 \) (1M) nerves
- \( \sim 10^9 \) (2B) neurons
- \( \sim 10^{12} \) (2T) synapses

The Brain: A Representation Machine

Experience $\mathcal{X}$ \quad Representation $\Phi$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

10M nerves \quad 100B neurons

- high dimensional (expanded 10K fold)
- nonnegative
- sparse (2-4% active)

Not a recording device

30f/sec (300M dims/sec): exceeds \# neurons in 5 mins
Lifetime $\sim 8.4M$ times longer than 5 mins
Consider the Cerebral Cortex

16B neurons
16T synapses

Essentially a 2D sheet


40K × 40K “pixels” (columns)
20 × 20 retina displays 13”
6 m × 6 m retina display
40K × 6 m = 240 km
How is Representation Organized in the Cortex?

Functional areas

Sensory processing regions

Layers

Serre et al., A theory of object recognition: computations and circuits in the feed-forward path of the ventral stream in primate visual cortex, MIT AI Memo 2005-036.

Topographic maps

Abstractions

- faces
- celebrities
  (grandma)
- mirror neurons
  (arm waving, empathy, sports-movies-stories, autism)
How is the Neural Representation Computed?

Synaptic connections: directed graph

\[
\begin{array}{c|c}
\text{nerves} & \text{neurons} \\
\hline
N & U \\
W & S \\
\end{array}
\]

\[
\begin{bmatrix}
\mathbf{x} \\
\phi
\end{bmatrix}^\top \begin{bmatrix}
N & U \\
W & S
\end{bmatrix} \mapsto \begin{bmatrix}
\text{input to } \mathbf{x} \\
\text{input to } \phi
\end{bmatrix}^\top
\]

intra-region connections: inhibitory
inter-region connections: exitatory

Connections are sparse
\[\approx 10^3 \text{ (out of } 10^{11}) \text{ connections per neuron}\]

If connections were dense:
\[.25 \times 10^{-6} m^3 \times 10^{22} \approx (5.4 m)^3\]

sparse:
\[.25 \times 10^{-6} m^3 \times 10^{14} \approx (1.2 cm)^3\]
Neural Learning

Learn synaptic connections

Development
- spatial arrangement forms early – guided by genes
- functional circuits form later – guided by experience

20K genes  100B neurons  100T synapses

Learning is unsupervised
- e.g. learn appropriate response for experience (duck!)
- $10^{14}$ synaptic connections to specify
- $10^9$ seconds alive
- $10^5$ connections/second have to be specified
How? No such training signal
"General" purpose

Brain must not be targeting specific concepts

- "General" unsupervised learning principles used to recover representations from "normal" experience?
Unsupervised Machine Learning
Unsupervised Machine Learning

Basic forms
Clustering
Dimensionality reduction
  – multi-label clustering
  – ICA

Sparse coding
  – restricted Boltzmann machines
  – nonnegative matrix factorization
  – topic modeling

Extensions

Multi-view
Temporal
(Causal)
Deep
Relations

Experience $X$ $\Phi$
General Learning Principles

Generative model

\[
\begin{align*}
\min_{\Phi, W \in \mathcal{W}} & \quad L(\Phi W, X) + R(\Phi) \\
\end{align*}
\]

backward projection

Note: scale invariant: have to constrain both $W$ and $\Phi$

Recognition model

\[
\begin{align*}
\min_{\Phi, U \in \mathcal{U}} & \quad L(XU, \Phi) + R(\Phi) \\
\end{align*}
\]

forward projection

Note: vacuous unless $\Phi$ constrained
General Learning Principles

Combined model

\[
\min_{\Phi, W \in W, U \in U} L(\Phi W, X) + L(XU, \Phi) + R(\Phi)
\]

Note: these models subsume likelihood, MAP, mutual information
General Learning Principles

Principles
- Generative model
- Recognition model
- Combined model

Problem
None of these formulations are jointly convex

Chicken and egg problem
  infer $\Phi$ given $U$ and-or $W$
  optimize $U$ and-or $W$ given $\Phi$

Common strategy: alternate and declare victory
  But no guarantees: unreliable results, hides nature of solution
Two Constraints on Learning Formulations

1. Biological plausibility
   ● implementable in brain via strictly local computations
   ● don’t worry about local optima

2. Computational tractability
   ● guarantee global solution in polynomial time
   ● don’t worry about biological plausibility
Convex Formulations of Representation Learning

Why?
- Modular: separates specification from implementation
- Exportable: can be commoditized
- Expository: reveals true nature of optima

Why not?
- Impossible?
- Hopeless?
- Not biologically plausible?
Many Recent Positive Results

- Can: achieve convex formulations for seemingly hard problems
- Do: observe improved results over local methods
- Do: reveal nature of global optima not revealed locally

Two general tricks
1. Induced matrix norms
2. Output kernels
Part 1: Induced Matrix Norms
Discovering a feature representation from data

Consider generative model

Examples
- dimensionality reduction (PCA, exponential family PCA)
- sparse coding
- independent component analysis

Usually involves learning both
a latent representation for data and a data reconstruction model
Challenge

Optimal feature discovery appears to be generally intractable
Have to jointly train
• latent representation
• data reconstruction model

Usually resort to alternating minimization
(sole exception: PCA)
• Many feature discovery problems can be solved **globally**

• Requires a transformation of the joint training problem

• Can easily extend to multi-view and semi-supervised scenarios
Unsupervised feature discovery
Unsupervised feature discovery

Single layer case = matrix factorization

Choose $W$ and $\Phi$ to minimize data reconstruction loss

$$L(\Phi W, X) = \sum_{i=1}^{t} L(\Phi_i W, X_i)$$

Seek desired structure in latent feature representation

- $\Phi$ low rank: dimensionality reduction
- $\Phi$ sparse: sparse coding
- $\Phi$ cols independent: independent component analysis
Generalized matrix factorization

Assume reconstruction loss $L(\hat{x}, x)$ is convex in first argument

Bregman divergence

$$L(\hat{x}, x) = D_F(\hat{x} \parallel x) = D_{F^*}(f(x) \parallel f(\hat{x}))$$

($F$ strictly convex potential with transfer $f = \nabla F$)

Tries to make $\hat{x} \approx x$

Matching loss

$$L(\hat{x}, x) = D_F(\hat{x} \parallel f^{-1}(x)) = D_{F^*}(x \parallel f(\hat{x}))$$

Tries to make $f(\hat{x}) \approx x$

(A nonlinear predictor, but loss still convex in $\hat{x}$)

Regular exponential family

$$L(\hat{x}, x) = -\log p_B(x \mid \phi) = D_F(\hat{x} \parallel f^{-1}(x)) - F^*(x) - \text{const}$$
Training problem

\[
\min_{W \in \mathbb{R}^{m \times n}} \min_{\Phi \in \mathbb{R}^{t \times m}} L(\Phi W, X)
\]

How to impose desired structure on $\Phi$?
Training problem

\[
\min_{W \in \mathbb{R}^{m \times n}} \min_{\Phi \in \mathbb{R}^{t \times m}} L(\Phi W, X)
\]

How to impose desired structure on \( \Phi \)?

Dimensionality reduction

Fix \# features \( m < \min(n, t) \)

- But only known to be tractable if \( L(\hat{X}, X) = \|\hat{X} - X\|^2_F \) (PCA)
- No known efficient algorithm for other standard losses

Problem

\( \text{rank}(\Phi) = m \) constraint is too hard
Training problem

\[
\min_{W \in \mathcal{W}_2^m} \min_{\Phi \in \mathbb{R}^{t \times m}} L(\Phi W, X) + \alpha \|\Phi'\|_{2,1}
\]

How to impose desired structure on \( \Phi \)?

Relaxed dimensionality reduction (subspace learning)

Add rank reducing regularizer

\[
\|\Phi'\|_{2,1} = \sum_{j=1}^{m} \|\Phi_{j}^{\cdot}\|_2
\]

Favors null columns in \( \Phi \)

But need to add constraint to \( W \)

\[
W_{j}^{\cdot} \in \mathcal{W}_2 = \{w : \|w\|_2 \leq 1\}
\]

(Otherwise can make \( \Phi \) small just by making \( W \) large)
The $\| \cdot \|_{2,1}$ norm

\[
\begin{bmatrix}
1 & 1 \\
0 & 0 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
1 & 0 \\
0 & 1
\end{bmatrix}
\]

$\sqrt{2} \quad 2 \quad 2$

favors sparse rows
Training problem

\[
\min_{W \in \mathcal{W}_q^m} \min_{\Phi \in \mathbb{R}^{t \times m}} L(\Phi W, X) + \alpha \|\Phi\|_{1,1}
\]

How to impose desired structure on \( \Phi \)?

**Sparse coding**

Use sparsity inducing regularizer

\[
\|\Phi\|_{1,1} = \sum_{j=1}^m \sum_{i=1}^t |\Phi_{ij}|
\]

Favors sparse entries in \( \Phi \)

Need to add constraint to \( W \)

\[
W_j : \in \mathcal{B}_q = \{w : \|w\|_q \leq 1\}
\]

(Otherwise can make \( \Phi \) small just by making \( W \) large)
Training problem

\[
\min_{W \in \mathbb{R}^{m \times n}} \min_{\Phi \in \mathbb{R}^{t \times m}} L(\Phi W, X) + \alpha D(\Phi)
\]

How to impose desired structure on \( \Phi \)?

Independent components analysis

Usually enforces \( \Phi W = X \) as a constraint
- but interpolation is generally a bad idea
- Instead just minimize reconstruction loss
  plus a dependence measure \( D(\Phi) \) as a regularizer

Reduction

Classical ICA can be reduced to

\[
\min_{W \in \mathbb{R}^{m \times n}} \min_{\Phi \in \mathbb{R}^{t \times m}} \frac{1}{t} \| X - \Phi W \|_F^2 + \alpha \| \Phi \|_{1,1}
\]

for centered and normalized \( X \): sparse coding!
Consider subspace learning and sparse coding

\[
\min_{W \in \mathcal{W}^m} \min_{\Phi \in \mathbb{R}^{t \times m}} L(\Phi W, X) + \alpha \| \Phi \|
\]

Choice of \( \| \Phi \| \) and \( \mathcal{W} \) determines type of representation recovered.
Training problem

Consider subspace learning and sparse coding

\[
\min_{W \in \mathcal{W}^\infty} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi\|
\]

Choice of \(\|\Phi\|\) and \(\mathcal{W}\) determines type of representation recovered

Problem
Still have rank constraint imposed by \# new features \(m\)

Idea
Just relax \(m \rightarrow \infty\)
• Rely on sparsity inducing norm \(\|\Phi\|\) to select features
Training problem

Consider subspace learning and sparse coding

\[
\min_{W \in \mathcal{W}^\infty} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi\|
\]

Still have a problem

Optimization problem is not jointly convex in \( W \) and \( \Phi \)
Training problem

Consider subspace learning and sparse coding

\[
\min_{W \in \mathcal{W}^\infty} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi\|
\]

Still have a problem
Optimization problem is not jointly convex in \(W\) and \(\Phi\)

Idea 1: Alternate!
- convex in \(W\) given \(\Phi\)
- convex in \(\Phi\) given \(W\)

Could use any other form of local training
Training problem

Consider subspace learning and sparse coding

$$\min_{W \in \mathcal{W}^\infty} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi\|$$

Still have a problem

Optimization problem is not jointly convex in $W$ and $\Phi$

Idea 2: Boost!

- Implicitly fix $W$ to universal dictionary
- Keep column-wise sparse $\Phi$
- Incrementally select row in $W$ ("weak learning problem")
- Update sparse $\Phi$

Can prove convergence under broad conditions
Training problem

Consider subspace learning and sparse coding

$$\min_{W \in \mathcal{W}} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi\|$$

Still have a problem?
Optimization problem is not jointly convex in $W$ and $\Phi$

Idea 3: Solve!
- Can easily solve for globally optimal joint $W$ and $\Phi$
- **But** requires a significant reformulation
Key Observation
Equivalent reformulation

Theorem

\[
\min_{W \in \mathcal{W}^\infty} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi'\|_{p,1} \\
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \|\hat{X}\|
\]

- \(\|\cdot\|\) is an induced matrix norm on \(\hat{X}\)
determined by \(\mathcal{W}\) and \(\|\cdot\|_{p,1}\)

Important fact

Norms are always convex

Computational strategy

1. Solve for optimal response matrix \(\hat{X}\) first (convex minimization)
2. Then recover optimal \(W\) and \(\Phi\) from \(\hat{X}\)
Example: subspace learning

\[
\min_{W \in \mathcal{W}_2^n} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi'\|_{2,1}
\]

\[
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \|\hat{X}\|_{tr}
\]

Recovery

- Let \( U\Sigma V' = \text{svd}(\hat{X}) \)
- Set \( W = V' \) and \( \Phi = U\Sigma \)

Preserves optimality

- \( \| W_j \|_2 = 1 \) hence \( W \in \mathcal{W}_2^n \)
- \( \|\Phi'\|_{2,1} = \|\Sigma U'\|_{2,1} = \sum_j \sigma_j \|U_j\|_2 = \sum_j \sigma_j = \|\hat{X}\|_{tr} \)

Thus

\[
L(\hat{X}, X) + \alpha \|\hat{X}\|_{tr} = L(\Phi W, X) + \alpha \|\Phi'\|_{2,1}
\]
Example: sparse coding

\[
\min_{W \in \mathcal{W}^\infty_q} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi\|_{1,1} = \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \|\hat{X}\|_{q,1}
\]

Recovery

\[
W = \left[ \frac{1}{\|\hat{X}_1\|_q}, \ldots, \frac{1}{\|\hat{X}_t\|_q} \right]' \quad \text{(rescaled rows)}
\]

\[
\Phi = \begin{bmatrix}
\|\hat{X}_1\|_q & 0 \\
0 & \ddots \\
0 & 0 & \|\hat{X}_t\|_q 
\end{bmatrix} \quad \text{(diagonal matrix)}
\]

Preserves optimality

- \(\|W_j\|_q = 1\) hence \(W \in \mathcal{W}_q^t\)
- \(\|\Phi\|_{1,1} = \sum_j \|\hat{X}_j\|_q = \|\hat{X}\|_{q,1}\)

Thus

\[
L(\hat{X}, X) + \alpha \|\hat{X}\|_{q,1} = L(\Phi W, X) + \alpha \|\Phi\|_{1,1}
\]
Example: sparse coding

Outcome
Sparse coding with $\| \cdot \|_{1,1}$ regularization = vector quantization
• drops some examples
• memorizes remaining examples

Optimal solution is not overcomplete
Could not make these observations using local solvers
Explaining the main result

Theorem

$$\min_{W \in \mathcal{W}} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi'\|_{p,1}$$

$$= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \|\hat{X}\|$$

for an induced matrix norm $\|\hat{X}\| = \|\hat{X}\|_{(\mathcal{W}, p^*)}$
Explaining the main result

**Theorem**

\[
\min_{W \in \mathcal{W}^\infty} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi'\|_{p,1} = \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \|\hat{X}\|
\]

for an induced matrix norm \(\|\hat{X}\| = \|\hat{X}\|^{*}_{(\mathcal{W},p^*)}\)

A dual norm

\[
\|\hat{X}\|^{*}_{(\mathcal{W},p^*)} = \max_{\|\Lambda\|_{(\mathcal{W},p^*)} \leq 1} \text{tr}(\Lambda' \hat{X})
\]

(standard definition of a dual norm)
Explaining the main result

Theorem

$$\min_{W \in \mathcal{W}} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi'\|_{p,1}$$

$$= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \|\hat{X}\|$$

for an induced matrix norm $\|\hat{X}\| = \|\hat{X}\|_{(\mathcal{W}, p^*)}^*$

A dual norm

$$\|\hat{X}\|_{(\mathcal{W}, p^*)} = \max_{\|\Lambda\|_{(\mathcal{W}, p^*)} \leq 1} \text{tr}(\Lambda' \hat{X})$$

(standard definition of a dual norm)

of a vector-norm induced matrix norm

$$\|\Lambda\|_{(\mathcal{W}, p^*)} = \max_{w \in \mathcal{W}} \|\Lambda w\|_{p^*}$$

(easy to prove this yields a norm on matrices)
Proof outline

\[
\min_{W \in \mathcal{W}} \phi^\infty \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi'\|_{p,1}
\]

\[
= \min_{\hat{X} \in \mathbb{R}^{\hat{t} \times n}} \min_{W \in \mathcal{W}^\infty} \min_{\Phi : \Phi W = \hat{X}} L(\hat{X}, X) + \alpha \|\Phi'\|_{p,1}
\]

\[
= \min_{\hat{X} \in \mathbb{R}^{\hat{t} \times n}} L(\hat{X}, X) + \alpha \min_{W \in \mathcal{W}^\infty} \min_{\Phi : \Phi W = \hat{X}} \|\Phi'\|_{p,1}
\]
Proof outline

\[
\min_{W \in \mathcal{W}} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi^\prime\|_{p,1}
\]

\[
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} \min_{W \in \mathcal{W}} \min_{\Phi: \Phi W = \hat{X}} L(\hat{X}, X) + \alpha \|\Phi^\prime\|_{p,1}
\]

\[
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \min_{W \in \mathcal{W}} \min_{\Phi: \Phi W = \hat{X}} \|\Phi^\prime\|_{p,1}
\]

For any \( W \in \mathcal{W}^{\infty} \) that spans the rows of \( \hat{X} \)

\[
\min_{\Phi: \Phi W = \hat{X}} \|\Phi^\prime\|_{p,1} = \min \max \max \operatorname{tr}(V^\prime \Phi) + \operatorname{tr}(\Lambda^\prime (\hat{X} - \Phi W))
\]

\[
\leq \max \max \min \operatorname{tr}(\Lambda^\prime \hat{X}) + \operatorname{tr}(\Phi^\prime (V - \Lambda W^\prime))
\]

\[
= \max \max \operatorname{tr}(\Lambda^\prime \hat{X})
\]

\[
\operatorname{tr}(\Lambda^\prime \hat{X})
\]
Proof outline

\[
\min_{W \in \mathcal{W}} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi'\|_{p,1}
\]

\[
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} \min_{W \in \mathcal{W}} \min_{\Phi : \Phi W = \hat{X}} L(\hat{X}, X) + \alpha \|\Phi'\|_{p,1}
\]

\[
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \min_{W \in \mathcal{W}} \max_{\Lambda : \|\Lambda W'\|_{p^*,\infty} \leq 1} \text{tr}(\Lambda' \hat{X})
\]
Proof outline

\[
\begin{align*}
\min_{W \in \mathcal{W}^\infty} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \| \Phi' \|_{p,1} \\
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} \min_{W \in \mathcal{W}^\infty} \min_{\Phi : \Phi W = \hat{X}} L(\hat{X}, X) + \alpha \| \Phi' \|_{p,1} \\
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \min_{W \in \mathcal{W}^\infty} \max_{\Lambda : \| W \Lambda \|_{p^*, \infty} \leq 1} \text{tr}(\Lambda' \hat{X}) \\
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \max_{\Lambda : \| W \Lambda \|_{p^*, \infty} \leq 1, \forall W \in \mathcal{W}^\infty} \text{tr}(\Lambda' \hat{X}) \\
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \max_{\Lambda : \| w' \Lambda \|_{p^*} \leq 1, \forall w \in \mathcal{W}} \text{tr}(\Lambda' \hat{X}) \\
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \max_{\Lambda : \| \Lambda \|_{(\mathcal{W}, p^*)} \leq 1} \text{tr}(\Lambda' \hat{X}) \\
= \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \| \hat{X} \|_{(\mathcal{W}, p^*)}^* \\
\text{done}
\end{align*}
\]
Closed form induced norms

Theorem

\[
\min_{W \in \mathcal{W}} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L(\Phi W, X) + \alpha \|\Phi'\|_{p,1} = \min_{\hat{X} \in \mathbb{R}^{t \times n}} L(\hat{X}, X) + \alpha \|\hat{X}\|^{*}(\mathcal{W}, p^*)
\]

Special cases

- \(\mathcal{W}_2, \|\Phi'\|_{2,1} \Rightarrow \|\hat{X}\|^{*}(\mathcal{W}_2, 2) = \|\hat{X}\|_{tr}\) (subspace learning)
- \(\mathcal{W}_q, \|\Phi'\|_{1,1} \Rightarrow \|\hat{X}\|^{*}(\mathcal{W}_q, \infty) = \|\hat{X}\|_{q,1}\) (sparse coding)
- \(\mathcal{W}_1, \|\Phi'\|_{p,1} \Rightarrow \|\hat{X}\|^{*}(\mathcal{W}_1, p^*) = \|\hat{X}'\|_{p,1}\)
Simple Experiments
**Experimental results**

Alternate : repeatedly optimize over $W, \Phi$ successively

Global : recover global joint minimizer over $W, \Phi$
## Experimental results: Sparse coding

### Objective value achieved

<table>
<thead>
<tr>
<th></th>
<th>COIL</th>
<th>WBC</th>
<th>BCI</th>
<th>Ionos</th>
<th>G241N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate</td>
<td>1.314</td>
<td>4.918</td>
<td>0.898</td>
<td>1.612</td>
<td>1.312</td>
</tr>
<tr>
<td>Global</td>
<td>0.207</td>
<td>0.659</td>
<td>0.306</td>
<td>0.330</td>
<td>0.207</td>
</tr>
</tbody>
</table>

\( \times 10^{-2} \)

(squared loss, \( q = 2 \), \( \alpha = 10^{-5} \))
Experimental results: Sparse coding

Run time (seconds)

<table>
<thead>
<tr>
<th></th>
<th>COIL</th>
<th>WBC</th>
<th>BCI</th>
<th>Ionos</th>
<th>G241N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate</td>
<td>1.95</td>
<td>10.54</td>
<td>0.88</td>
<td>1.71</td>
<td>2.37</td>
</tr>
<tr>
<td>Global</td>
<td>0.06</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.09</td>
</tr>
</tbody>
</table>

(squared loss, $q = 2$, $\alpha = 10^{-5}$)
# Experimental results: Subspace learning

## Objective value achieved

<table>
<thead>
<tr>
<th></th>
<th>COIL</th>
<th>WBC</th>
<th>BCI</th>
<th>Ionos</th>
<th>G241N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate</td>
<td>1.314</td>
<td>4.957</td>
<td>0.903</td>
<td>1.632</td>
<td>1.313</td>
</tr>
<tr>
<td>Global</td>
<td>0.072</td>
<td>0.072</td>
<td>0.092</td>
<td>0.079</td>
<td>0.205</td>
</tr>
</tbody>
</table>

\[ \times 10^{-2} \]

(squared loss, \( \alpha = 10^{-5} \))
## Experimental results: Subspace learning

### Run time (seconds)

<table>
<thead>
<tr>
<th></th>
<th>COIL</th>
<th>WBC</th>
<th>BCI</th>
<th>Ionos</th>
<th>G241N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate</td>
<td>2.40</td>
<td>9.31</td>
<td>1.12</td>
<td>0.47</td>
<td>2.43</td>
</tr>
<tr>
<td>Global</td>
<td>2.18</td>
<td>0.06</td>
<td>0.19</td>
<td>0.06</td>
<td>2.11</td>
</tr>
</tbody>
</table>

(squared loss, $\alpha = 10^{-5}$)
Every norm is convex

**But** not every induced matrix norm is **tractable**

\[
\|X\|_2 = \sigma_{\text{max}}(X)
\]

\[
\|X\|_1 = \max \sum_j |X_{ij}|
\]

\[
\|X\|_{\infty} = \max \sum_i |X_{ij}|
\]

\[
\|X\|_p \quad \text{NP-hard to approximate for } p \neq 1, 2, \infty
\]

**Question**

Any other useful induced matrix norms that are tractable?

**Yes!**
Multi-view Feature Discovery
Multi-view feature discovery

Learn

\( \Phi = [\Phi_l', \Phi_u']' \) data representation

\( W = \) input reconstruction model \( f(\Phi W) \approx X \)

\( B = \) output reconstruction model \( h(\Phi_l B) \approx Y_l \)
Multi-view feature discovery

Let

\[
Z = \begin{bmatrix}
X_l & Y_l \\
X_u & \emptyset
\end{bmatrix} \quad U = [W \ B] \quad U = [B \ W]
\]

Formulation

\[
\min_{W \in \mathcal{W}^\infty} \min_{B \in \mathcal{B}^\infty} \min_{\Phi \in \mathbb{R}^{t \times \infty}} L_u(\Phi W, X) + \beta L_s(\Phi B, Y_l) + \alpha \|\Phi'\|_{p,1}
\]

\[
= \min_{\hat{Z} \in \mathbb{R}^{(n+k) \times t}} \tilde{L}(\hat{Z}, Z) + \alpha \|\hat{Z}\|^*_{(U,p^*)}
\]

Note

Imposing separate constraints on \( W \) and \( B \)

Questions

- Is the induced norm \( \|\hat{Z}\|^*_{(U,p^*)} \) efficiently computable?
- Can optimal \( B, W, \Phi \) be recovered from optimal \( \hat{Z} \)?
Example: sparse coding formulation

Regularizer: \( \| \Phi \|_{1,1} \)

Constraints:
\[
\mathcal{W}_{q_1} = \{ \mathbf{w} : \| \mathbf{w} \|_{q_1} \leq 1 \}
\]
\[
\mathcal{W}_{q_2} = \{ \mathbf{w} : \| \mathbf{w} \|_{q_2} \leq \gamma \}
\]
\[
\mathcal{U}_{q_1} = \mathcal{W} \times \mathcal{B}
\]

Theorem

\[
\| \hat{Z} \|_{(\mathcal{U}_{q_2}, \infty)}^{*} = \sum_j \max \left( \| \hat{Z}_j^X \|_{q_1}, \frac{1}{\gamma} \| \hat{Z}_j^Y \|_{q_2} \right)
\]
efficiently computable

Recovery

\[
\Phi_{jj} = \max \left( \| \hat{Z}_j^X \|_{q_1}, \frac{1}{\gamma} \| \hat{Z}_j^Y \|_{q_2} \right) \text{ (diagonal matrix)}
\]

\[
U = \Phi^{-1} \hat{Z}
\]

Preserves optimality
But still reduces to a form of vector quantization
Example: subspace learning formulation

Regularizer: $\|\Phi'\|_{2,1}$

Constraints:

$\mathcal{W}_2 = \{ w : \|w\|_2 \leq 1 \}$

$\mathcal{B}_2 = \{ b : \|b\|_2 \leq \gamma \}$

$\mathcal{U}_2 = \mathcal{W} \times \mathcal{B}$

Theorem

$$\| \hat{Z} \|_{(\mathcal{U}_2^2, \infty)}^* = \max_{\rho \geq 0} \| \hat{Z} D^{-1}_\rho \|_{tr} \quad \text{where} \quad D_\rho = \begin{bmatrix} \sqrt{1 + \gamma \rho} & l \\ 0 & \sqrt{\frac{1 + \gamma \rho}{\rho}} l \end{bmatrix}$$

efficiently computable: quasi-concave in $\rho$
Example: subspace learning formulation

Lemma: dual norm

\[ \| \Lambda \|_{(U^2, 2)}^2 = \max_{h: \| h^X \|_2 = 1, \| h^Y \|_2 = \gamma} h' \Lambda' \Lambda h \]

\[ = \max_{H: H \succeq 0, \text{tr}(HI^X) = 1, \text{tr}(HI^Y) = \gamma} \text{tr}(H \Lambda' \Lambda) \]

\[ = \min_{\lambda \geq 0, \nu \geq 0} \min_{\Lambda: \Lambda' \Lambda \preceq \lambda I^X + \nu I^Y} \lambda + \gamma \nu \]

\[ = \min_{\lambda \geq 0, \nu \geq 0} \min_{\Lambda: \| \Lambda D_{\nu/\lambda} \|_{sp}^2 \leq \lambda + \gamma \nu} \lambda + \gamma \nu \]

\[ = \min_{\lambda \geq 0, \nu \geq 0} \| \Lambda D_{\nu/\lambda} \|_{sp}^2 \]

\[ = \min_{\rho \geq 0} \| \Lambda D_{\rho} \|_{sp}^2 \]
Example: subspace learning formulation

Can easily derive target norm from dual norm

\[ \| \hat{Z} \|_{(U_2^2, 2)}^* = \max_{\| \Lambda \|_{(U_2^2, 2)} \leq 1} \text{tr}(\Lambda' \hat{Z}) \]

\[ = \max_{\rho \geq 0} \max_{\Lambda : \| \Lambda D_\rho \|_{sp} \leq 1} \text{tr}(\Lambda' \hat{Z}) \]

\[ = \max_{\rho \geq 0} \max_{\tilde{\Lambda} : \| \tilde{\Lambda} \|_{sp} \leq 1} \text{tr}(D_\rho^{-1} \tilde{\Lambda}' \hat{Z}) \]

\[ = \max_{\rho \geq 0} \| \hat{Z} D_\rho^{-1} \| \text{tr} \]

(proves theorem)
Example: subspace learning formulation

Recovery
Given optimal \( \hat{Z} \), recover \( U \) and \( \Phi \) iteratively by repeating:

\( \bullet \) \( (\Phi^{(\ell)}, \Lambda^{(\ell)}) \in \arg \min_{\Phi} \max_{\Lambda} \|\Phi'\|_{2,1} + \text{tr}(\Lambda'(\hat{Z} - \Phi U^{(\ell)})) \)

\( \bullet \) \( u^{(\ell+1)} \in \arg \max_{u \in U_2^2} \|\Lambda^{(\ell)} u\|_2 \)

\( \bullet \) \( U^{(\ell+1)} = [U^{(\ell)}; u^{(\ell+1)'}] \)

Converges to optimal \( U \) and \( \Phi \)

\( \bullet \) \( \Phi^{(\ell)} U^{(\ell)} = \hat{Z} \) for all \( \ell \)

\( \bullet \) \( \|\Phi^{(\ell)}'\|_{2,1} \to \|\hat{Z}\|^*_{(U_2^2,2)} \)
Experiments
Experimental results: Subspace learning

Staged: first locally optimize $B$, $\Phi$, then optimize $W$
Alternate: repeatedly optimize over $B$, $W$, $\Phi$ successively
Global: recover joint global minimizer over $B$, $\Phi$, $W$
Experimental results: Subspace learning

### Objective value achieved

<table>
<thead>
<tr>
<th></th>
<th>COIL</th>
<th>WBC</th>
<th>BCI</th>
<th>Ionos</th>
<th>G241N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staged</td>
<td>1.384</td>
<td>1.321</td>
<td>0.799</td>
<td>0.769</td>
<td>1.381</td>
</tr>
<tr>
<td>Alternate</td>
<td>0.076</td>
<td>0.122</td>
<td>0.609</td>
<td>0.081</td>
<td>0.076</td>
</tr>
<tr>
<td>Global</td>
<td>0.070</td>
<td>0.113</td>
<td>0.069</td>
<td>0.078</td>
<td>0.070</td>
</tr>
</tbody>
</table>

(1/3 labeled, 2/3 unlabeled, squared loss, $\alpha^* = 10$, $\beta = 0.1$)
Experimental results: Subspace learning

Run time (seconds)

<table>
<thead>
<tr>
<th></th>
<th>COIL</th>
<th>WBC</th>
<th>BCI</th>
<th>Ionos</th>
<th>G241N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staged</td>
<td>272</td>
<td>73</td>
<td>45</td>
<td>28</td>
<td>290</td>
</tr>
<tr>
<td>Alternate</td>
<td>2352</td>
<td>324</td>
<td>227</td>
<td>112</td>
<td>2648</td>
</tr>
<tr>
<td>Global</td>
<td>106</td>
<td>8</td>
<td>25</td>
<td>61</td>
<td>94</td>
</tr>
</tbody>
</table>

(1/3 labeled, 2/3 unlabeled, squared loss, $\alpha^* = 10$, $\beta = 0.1$)
## Experimental results: Subspace learning

### Transductive generalization error

<table>
<thead>
<tr>
<th>Method</th>
<th>COIL</th>
<th>WBC</th>
<th>BCI</th>
<th>Ionos</th>
<th>G241N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staged</td>
<td>0.476</td>
<td>0.200</td>
<td>0.452</td>
<td>0.335</td>
<td>0.484</td>
</tr>
<tr>
<td>Alternate</td>
<td>0.464</td>
<td>0.388</td>
<td>0.440</td>
<td>0.457</td>
<td>0.478</td>
</tr>
<tr>
<td>Global</td>
<td>0.388</td>
<td>0.134</td>
<td>0.380</td>
<td>0.243</td>
<td>0.380</td>
</tr>
<tr>
<td>(Lee et al. 2009)</td>
<td>0.414</td>
<td>0.168</td>
<td>0.436</td>
<td>0.350</td>
<td>0.452</td>
</tr>
<tr>
<td>(Goldberg et al. 2010)</td>
<td>0.484</td>
<td>0.288</td>
<td>0.540</td>
<td>0.338</td>
<td>0.524</td>
</tr>
</tbody>
</table>

(1/3 labeled, 2/3 unlabeled, squared loss, $\alpha^* = 10$, $\beta = 0.1$)
Experimental results: Image denoising

Train on two views
- different lighting conditions
- learn $\Phi$, $W$ and $B$

Test
- Given $X$-view, recover $\phi'$, recover $Y$-view
Part 2: Output Kernels
Recognition model

Clustering

- Add constraints

\[ \Phi \in \{0, 1\}^{t \times k}, \quad \Phi 1 = 1 \]

- Training

\[
\min_{\Phi \in \mathcal{C}, \mathcal{U} \in \mathcal{U}} L(XU, \Phi)
\]

- Prediction

given \( x' \), \( \hat{\phi}' = \text{indmax}(x'U) \)
Clustering with recognition models

Use output kernels

• Many supervised learning methods only require output kernel i.e. equivalence relation instead of target labels
Two-class SVM

Primal and Lagrange dual

\[
\min_{u, \xi} \frac{\alpha}{2} \|u\|_2^2 + 1^t \xi \quad \text{s.t.} \quad \xi \geq 1 - \Delta(\phi)Xu
\]

\[
= \max_{0 \leq \lambda \leq 1} \lambda^t 1 - \frac{\alpha}{2} \lambda^t \Delta(\phi)XX^t \Delta(\phi)\lambda
\]

\[
= \max_{0 \leq \lambda \leq 1} \lambda^t 1 - \frac{\alpha}{2} \lambda^t (N \circ K) \lambda \quad \text{where } N = \phi \phi^t
\]

convex in \( N = \phi \phi^t \)
Multi-class SVM

Primal and Lagrange dual

\[ \min_{U, \xi} \frac{\alpha}{2} \|U\|_F^2 + 1^T \xi \quad \text{s.t.} \quad \xi 1' \geq 11' - \Phi + XU - \delta(XU\Phi')1' \]

\[ = \max_{\Lambda \succeq 0, \Lambda 1 = 1} t - \text{tr}(\Lambda' \Phi) - \frac{1}{2\alpha} \text{tr}((\Phi - \Lambda)'XX'(\Phi - \Lambda)) \]

Let \( \Omega \) be s.t. \( \Omega \Phi = \Lambda \)

\[ = \max_{\Omega \succeq 0, \Omega 1 = 1} t - \text{tr}(\Omega' \Phi' \Phi') - \frac{1}{2\alpha} \text{tr}((I - \Omega)'XX'(I - \Omega)\Phi\Phi') \]

\[ = \max_{\Omega \succeq 0, \Omega 1 = 1} t - \text{tr}(\Omega' N) - \frac{1}{2\alpha} \text{tr}((I - \Omega)'K(I - \Omega)N) \]

\[ \text{convex in } N = \Phi\Phi' \]

The equality holds provided \( \Phi \succeq 0, \Phi 1 = 1 \)
Multi-class logistic regression

Primal and Fenchel dual

\[
\min_U \frac{\alpha}{2} \|U\|_F^2 + A(XU) - \text{tr}(XU\Phi')
\]

where \( A(XU) = \sum_i \log(\sum_j X_i:U_j) \)

\[
= \max_{\Lambda \geq 0, \Lambda 1 = 1} -\text{tr}(\Lambda \log \Lambda') - \frac{1}{2\alpha} \text{tr}((\Phi - \Lambda)'XX' (\Phi - \Lambda))
\]

Let \( \Omega \) be s.t. \( \Omega\Phi = \Lambda \)

\[
= \max_{\Omega \geq 0, \Omega 1 = 1} -\text{tr}(\Omega \log \Omega') - 1'\Omega \log(N1) - \frac{1}{2\alpha} \text{tr}((I - \Omega)'K(I - \Omega)N)
\]

convex in \( N = \Phi\Phi' \)

The equality holds provided \( \Phi \in \{0, 1\}^{t \times k}, \Phi 1 = 1 \)
Multivariate least squares regression

Primal and Fenchel dual

$$\min_U \frac{\alpha}{2} \| U \|_F^2 + \| XU - \Phi \|_F^2$$

$$= \max_{\Lambda} -\frac{1}{2\alpha} \text{tr}(\Lambda' XX' \Lambda) - \frac{1}{2} \text{tr}(\Lambda' \Lambda) + \text{tr}(\Lambda' \Phi)$$

Let $\Omega$ be s.t. $\Omega \Phi = \Lambda$

$$= \max_{\Omega} -\frac{1}{2\alpha} \text{tr}(\Omega' K \Omega N) - \frac{1}{2} \text{tr}(\Omega' \Omega N) + \text{tr}(\Omega' N)$$

convex in $N = \Phi \Phi'$
Convex relaxation of clustering

\[
\min_{N \in \{0,1\}^{t \times k}, \delta(N) = 1, N \geq 0, \text{balanced}(N)} f(N)
\]

\[
\geq \min_{N \geq 0, \delta(N) = 1, N \geq 0, \text{balanced}(N)} f(N)
\]

Leads to
- Large margin clustering
- Viterbi-conditional EM
- Structured output extensions

Conclude
It is possible to simultaneously recover a (relaxed) cluster assignment and prediction model; jointly, globally, and tractably
Open questions

- Which supervised losses allow output kernelization?
- Approximation theory?
Tractable Robust Regression
Regression

Supervised recognition model

Given $X$ and $y$, recover linear predictor $u$ via

$$\min_u \frac{\alpha}{2} \|u\|^2_2 + L(Xu, y)$$

Normally, use a convex loss function $L$, to ensure tractability

Examples

$$L(Xu, y) = \|Xu - y\|^2_2$$

$$L(Xu, y) = \|Xu - y\|_1$$
Problem

For any convex loss

A single data point can perturb the solution arbitrarily

Demo
Robust regression

Standard approach: “M-estimation”

• Use a bounded loss function

Examples

\[ L(Xu, y) = \sum_i \frac{(X_i u - y_i)^2}{(X_i u - y_i)^2 + \sigma^2} \]

\[ L(Xu, y) = \sum_i \max(1, (X_i u - y_i)^2) \]
Problem

Minimizing any bounded (nontrivial) loss is NP-hard
Just use local minimization?

Demo
Robust regression

State of the art

Denial

Robust statistics

• propose *intractable* estimators
• prove they would work if you could have them
• bait and switch: declare satisfaction with weak heuristics
• ignore inconvenient truth: estimators will never exist

Machine learning

• assume outliers never exceed bounds
• achieve satisfaction by not realizing there’s a problem
Robust regression

All we want
An estimator that is
  tractable
    • polynomial time
  robust
    • bounded influence from any one data point
  consistent
    • converges to optimal model in limit

Doesn’t exist!
Robust regression

Fundamental dilemma

- convex loss $\Rightarrow$ tractable $\land$ not robust
- bounded loss $\Rightarrow$ robust $\land$ not tractable

Can have tractability or robustness
- but apparently not both

Really?
Reformulate as representation learning

Introduce outlier indicator $\phi$

- Represents a latent property of each data point
- Try to identify outliers via $\phi$ while optimizing model $u$

$$\min_{\phi \in \{0,1\}^{t \times 1}} \min_u \frac{\alpha}{2} \|u\|_2^2 \|\phi\|_1 + \underbrace{1'(1 - \phi)}_{\text{penalize outliers}} + \underbrace{\phi' L(Xu, y)}_{\text{loss on inliers}}$$ (1)

Chicken and egg problem

- given $\phi$, can optimize $u$
- given $u$, can infer $\phi$

Problem

Not jointly convex in $u$ and $\phi$

Note:

$$\min_{0 \leq \phi_i \leq 1} 1 - \phi_i + \phi_i L(X_i; u, i)$$
$$= \max(1, L(X_i; u, y_i))$$
Reformulate as representation learning

Reformulate

$$\begin{align*}
(1) &= \min_{0 \leq \phi \leq 1} \min_a \frac{\alpha}{2} \|\phi\|_1 K a + 1'(1 - \phi) + \phi' L(Ka, y) \\
&= \min_{0 \leq \phi \leq 1} \max_{\nu} 1'(1 - \phi) - \phi'(L^*(\nu, y) - \Delta(y)\nu) \\
&\quad - \frac{1}{2\alpha} \nu^T \left( K \circ (\phi\|\phi\|_1^{-1} \phi') \right) \nu
\end{align*}$$

Relax

$$\begin{align*}
N \in \mathcal{N}_\phi &= \left\{ N : N \succeq 0, N1 = \phi, \text{rank}(N) = 1 \right\} \\
\mathcal{M}_\phi &= \left\{ M : M \succeq 0, M1 = \phi, \text{tr}(M) \leq 1 \right\}
\end{align*}$$

$$\begin{align*}
(3) \geq \min_{0 \leq \phi \leq 1} \min_{M \in \mathcal{M}_\phi} \max_{\nu} 1'(1 - \phi) - \phi'(L^*(\nu, y) - \Delta(y)\nu) \\
&\quad - \frac{1}{2\alpha} \nu^T (K \circ M) \nu
\end{align*}$$

Round

• Fix $\phi$ and re-solve for $a$ in (2), recover estimator
Theorem

Under some conditions, can prove that this estimator

- is tractable
- has bounded response to any single data point
- is loss consistent

Demo
Some experiments

RMSE (stddev)
Seeded with 5% outliers

<table>
<thead>
<tr>
<th>Methods</th>
<th>Datasets</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cal-housing</td>
<td>abalone</td>
<td>pumadyn</td>
<td>bank-8fh</td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>1185 (124.59)</td>
<td>7.93 (0.67)</td>
<td>1.24 (0.42)</td>
<td>18.21 (6.57)</td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>1303 (244.85)</td>
<td>7.30 (0.40)</td>
<td>1.29 (0.42)</td>
<td>6.54 (3.09)</td>
<td></td>
</tr>
<tr>
<td>Huber</td>
<td>1221 (119.18)</td>
<td>7.73 (0.49)</td>
<td>1.24 (0.42)</td>
<td>7.37 (3.18)</td>
<td></td>
</tr>
<tr>
<td>LTS</td>
<td>533 (398.92)</td>
<td>755.1 (126)</td>
<td>0.32 (0.41)</td>
<td>10.96 (6.67)</td>
<td></td>
</tr>
<tr>
<td>LocBndL2</td>
<td>967 (522.40)</td>
<td>8.39 (0.54)</td>
<td>0.81 (0.77)</td>
<td>7.74 (9.40)</td>
<td></td>
</tr>
<tr>
<td>CvxBndL2</td>
<td>9 (0.64)</td>
<td>7.60 (0.86)</td>
<td>0.07 (0.07)</td>
<td>0.20 (0.05)</td>
<td></td>
</tr>
</tbody>
</table>
Learning Two-layer Models
Learning two-layer models

- Combined model

- Two-layer neural network
- Auto-encoder network

Key subproblem in local methods for “deep” learning
Does deep learning have to be intractable?

No!

Idea

- Use output kernel trick

$$\min_{\Phi, U \in \mathcal{U}, W \in \mathcal{W}} \ L(XU, \Phi) + L(\Phi W, Y) + R(\Phi)$$
Example

Nested SVMs

\[ \min_{\Phi \in \mathcal{C}} \max_{\Omega \geq 0, \Omega 1 = 1} \max_{\Upsilon \geq 0, \Upsilon 1 = 1} t - \text{tr}(\Omega^T \Phi \Phi') - \frac{1}{2\alpha} \text{tr}((I - \Omega)^T XX'(I - \Omega)\Phi \Phi') + \]
\[ t - \text{tr}(\Upsilon^T XX') - \frac{1}{2\alpha} \text{tr}((I - \Upsilon)^T \Phi \Phi'(I - \Upsilon)YY') + \tilde{R}(\Phi \Phi') \]

Convex reformulation

\[ \min_{N \in \mathcal{M}} \max_{\Omega \geq 0, \Omega 1 = 1} \max_{\Upsilon \geq 0, \Upsilon 1 = 1} t - \text{tr}(\Omega^T N) - \frac{1}{2\alpha} \text{tr}((I - \Omega)^T K(I - \Omega)N) + \]
\[ t - \text{tr}(\Upsilon^T XX') - \frac{1}{2\alpha} \text{tr}((I - \Upsilon)^T N(I - \Upsilon)YY') + \tilde{R}(N) \]

Learn a kernel between input and output

**jointly** with prediction models
Proof of concept

Data sets

Test misclassification error %

<table>
<thead>
<tr>
<th></th>
<th>1 layer linear SVM</th>
<th>2 layer linear SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>51.5  29.7  40.0  50.0</td>
<td>25.5  11.6  20.0  0.0</td>
</tr>
</tbody>
</table>
Conclusion
None of these representation learning problems looked easy

All had chicken and egg character:
- if $\Phi$ known, then can optimize $W$ and-or $U$
- if $W$ and-or $U$ known, then can infer $\Phi$

Yet still could achieve convex reformulation
(perhaps with a relaxation-rounding step)
Conclusion

Benefits of a globally solvable formulation

- repeatable
- separate specification from implementation
- reveal insight into structure of solutions

Still plenty of opportunity to do more!
References
References

Induced matrix norms


References

Output kernels


Robust estimation

