Applications of Generalized Gradient Descent

Anthony Rios

https://github.com/AnthonyMRios/proximal_methods_notes

Outline

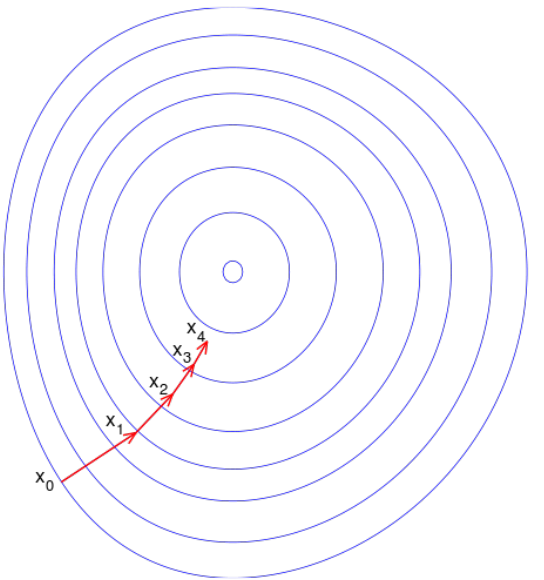
Background

Gradient Descent

Algorithm 1 Pseudocode for gradient descent

1: **for** $i = 1$ to N **do**
2: $\quad \theta = \theta - \alpha \nabla_\theta J(\theta)$
3: **end for**

$$J(\theta) = f(\theta) + h(\theta)$$



Decomposable functions

$$J(\theta) = f(\theta) + h(\theta)$$

- f is convex, differentiable
- h is convex, not necessarily differentiable

If J were differentiable, we could use gradient descent!

Generalized gradient descent: update rule defined as

$$\theta = prox_{\alpha}(\theta - \alpha \nabla_{\theta} f(\theta))$$

where

$$prox_{\alpha}(x) = argmin_{z \in \mathbb{R}^n} \frac{1}{2\alpha} \|x - z\|^2 + h(z)$$

Generalized Gradient Descent

$$J(\theta) = f(\theta) + h(\theta)$$

Algorithm 2 Pseudocode for generalized gradient descent

1: **for** $i = 1$ to N **do**
2: $\quad \theta = prox_{\alpha}(\theta - \alpha \nabla_{\theta} f(\theta))$
3: **end for**

Note: Can be **accelerated** with **momentum** (accelerated generalized gradient descent).

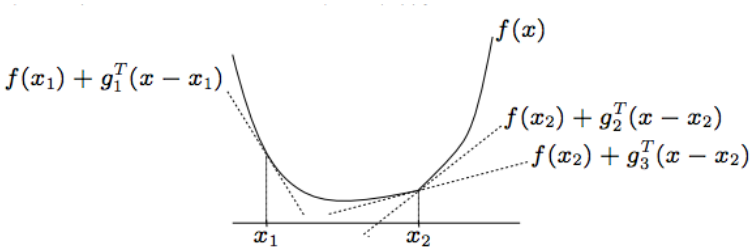
Subgradient Descent

A **subgradient** of convex $f : \mathbb{R}^n \to \mathbb{R}$ at x is any $g \in \mathbb{R}^n$ such that

$$f(x) \geq f(x_2) + g^T(x - x_2), \text{for all } x, x_2$$

Algorithm 3 Pseudocode for subgradient descent

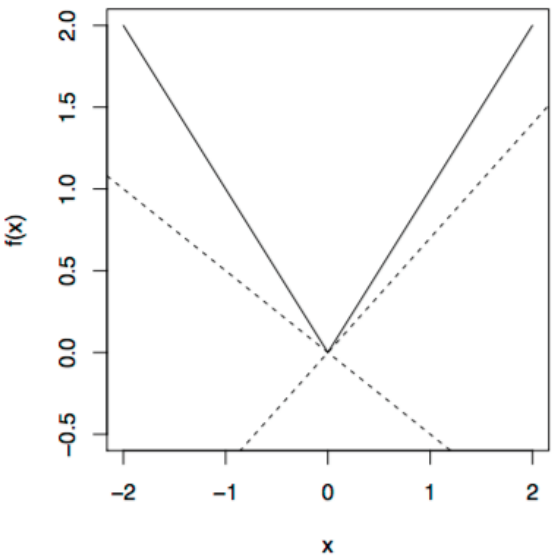
1: **for** $i = 1$ to N **do**
2: $\quad \theta = \theta - \alpha \partial_\theta J(\theta)$
3: **end for**



Subgradient descent is equivalent to gradient descent, but replaces the gradient with the subgradient

Subgradient Example

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



- For $x \neq 0$, unique subgradient $g = sign(x)$
- For $x = 0$, subgradient g is any element of $[-1, 1]$

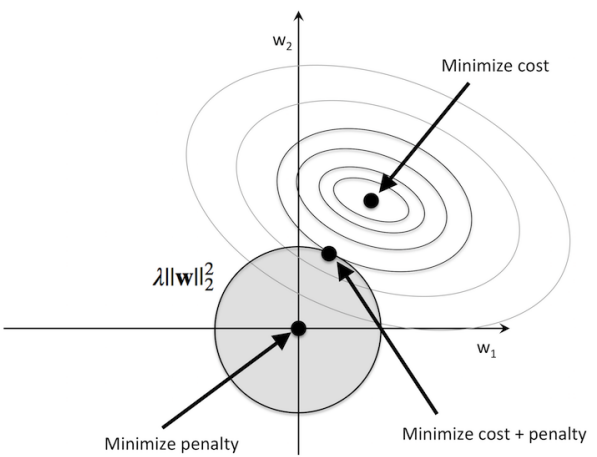
LASSO

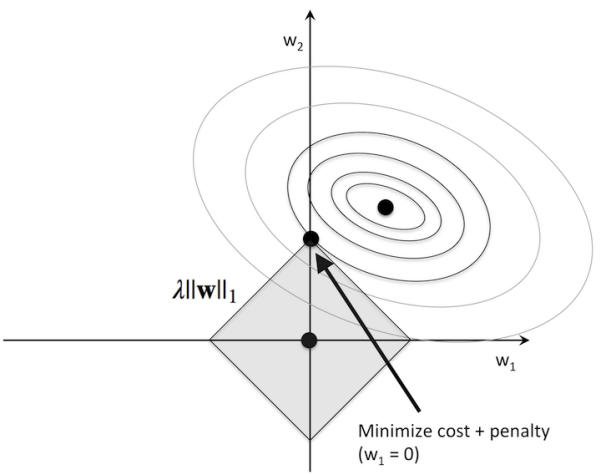
LASSO: Linear regression with l1 regularization

$$J(\theta) = \|X\theta - y\|_F^2 + \lambda\|\theta\|_1$$

Why not use l2 regularization (ridge regression)?

- LASSO results in **sparse** solutions
- Results in more **interpretable** models
- Reduces size of model (uses less memory)





LASSO

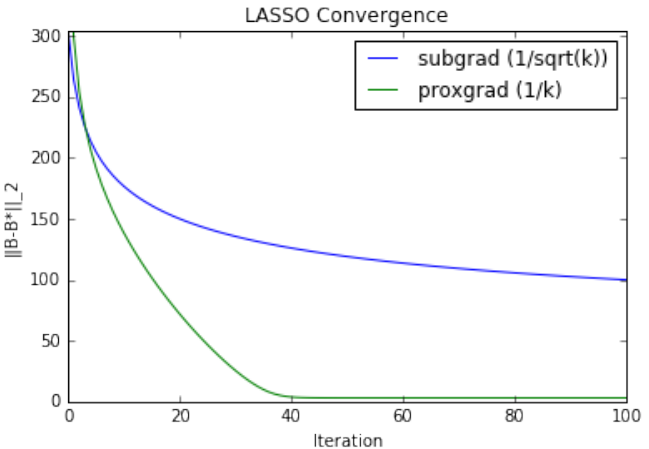
$$J(\theta) = \|X\theta - y\|_F^2 + \lambda\|\theta\|_1$$

The proximal operator for the l1 norm is called soft-thresholding

$$prox_{\alpha,\lambda}(x) = \begin{cases} z - \lambda, & \text{if } z > \lambda \\ 0, & \text{if } |z| \leq \lambda \\ z + \lambda, & \text{if } z < -\lambda \end{cases}$$

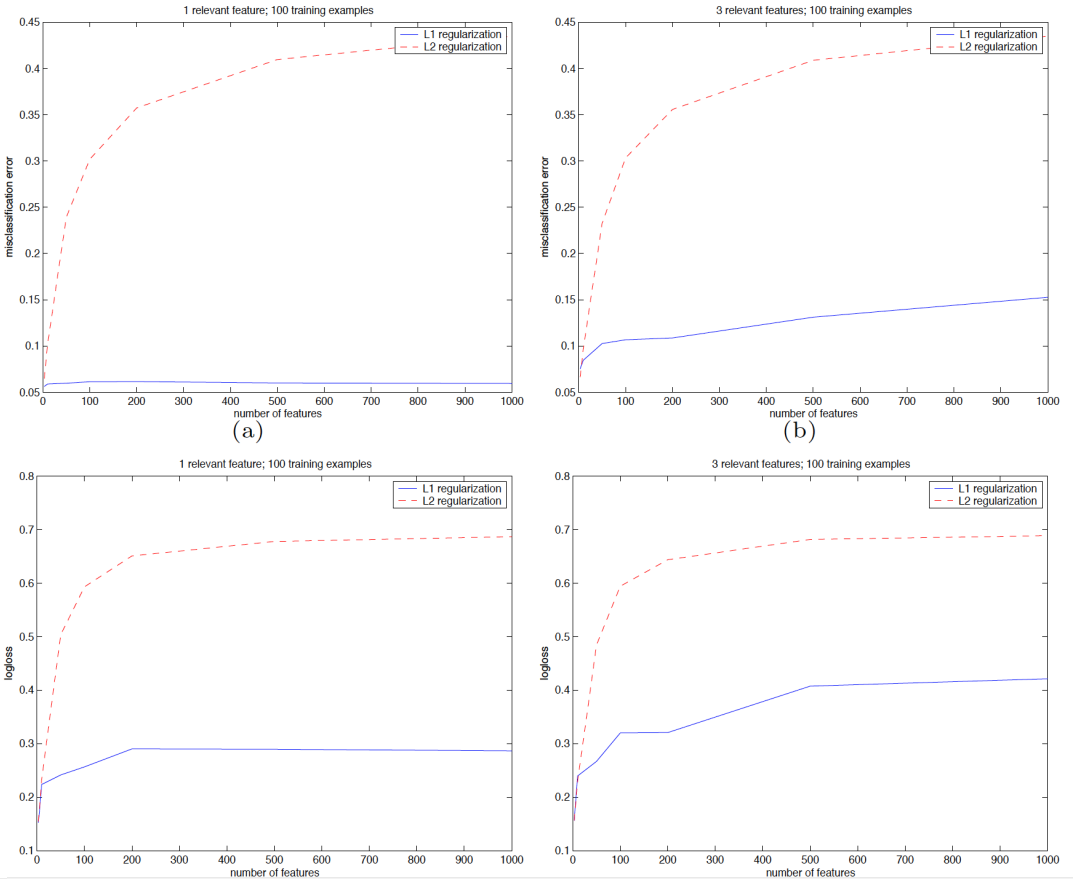
We can train using generalized gradient descent!

Experiment using a synthetic dataset



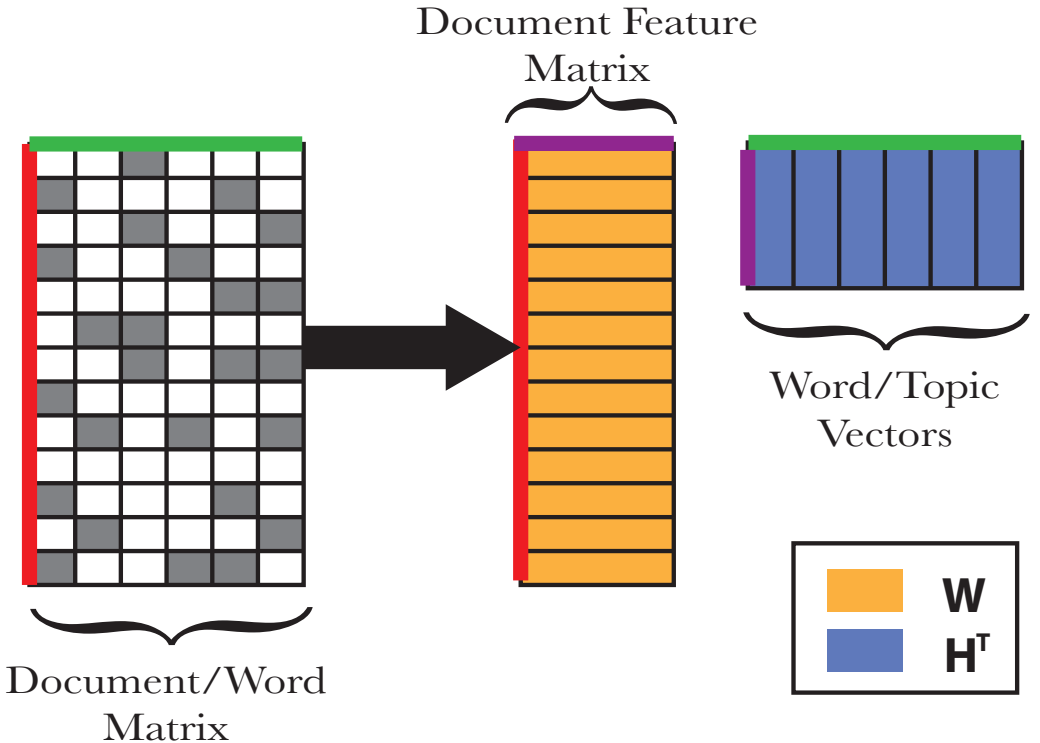
- 5001 examples
- 5000 features
- True parameter vector has 96 non-zero entries.
- The proximal method has 94/5000 non-zero coefs after 100 iterations.
- Subgradient method has 3010/5000.

L1 Regularized Logistic Regression (Ng 2004)



Non-negative Matrix Factorization

Non-negative Matrix Factorization



$$\underset{W,H}{\text{minimize}} \quad \|X - WH^T\|_F^2$$

$$\text{subject to} \quad W_{i,a} \geq 0, H_{j,b} \geq 0, \ \forall i, j, a, b$$

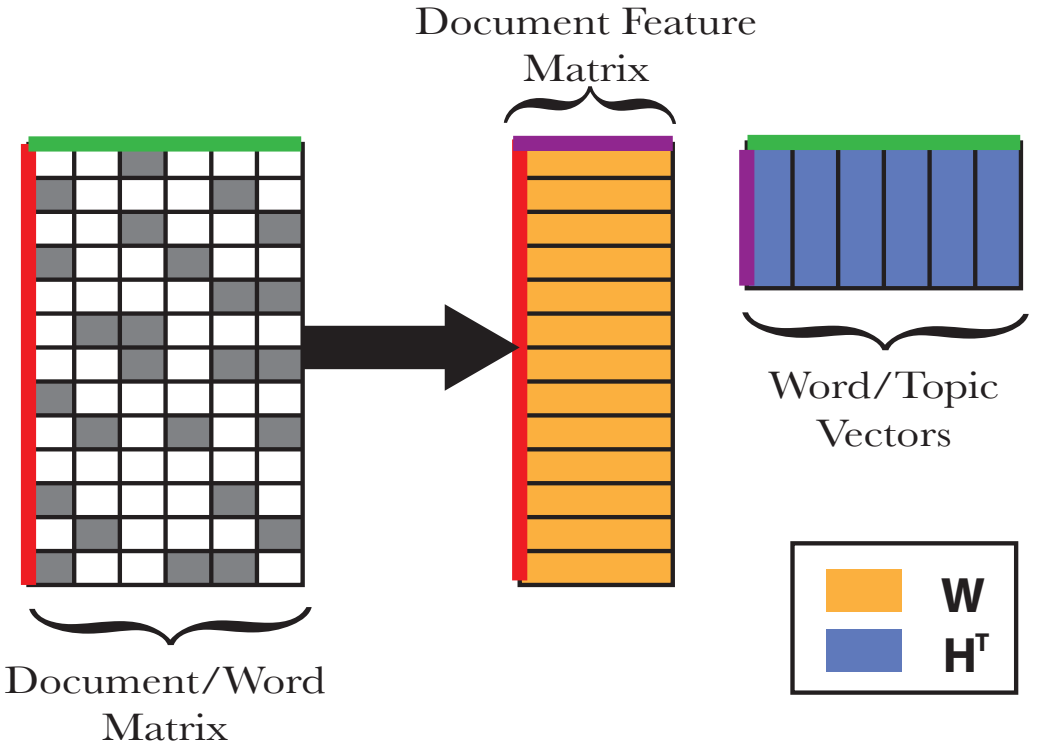
Applications of NMF

Why use non-negative if SVD would give achieve a better fit to the least squares loss?

Better interpretation for non-negative data

- Text mining
  - Topic Modeling/Word Embeddings (Ding et al., 2008)
  - (Soft) Document Clustering (Shahnaz et al., 2006)
  - Relationship Extraction (Riedel et al., 2013)
  - Multi-label classification (Yu et al., 2014)
- Image analysis and computer vision
  - Feature representation/sparse coding (Li et al, 2001)
- Social Network
  - Recommendation Systems (Zhang et al., 2006)
- Healthcare
  - Clinical Phenotyping (Joshi et al., 2016)

Non-negative Matrix Factorization



$$\underset{W,H}{\text{minimize}} \quad \|X - WH^T\|_F^2$$

$$\text{subject to} \quad W_{i,a} \geq 0, H_{j,b} \geq 0, \ \forall i, j, a, b$$

Non-negative Matrix Factorization (Lin, 2007)

Projected gradient method form non-negative matrix factorization is formulated as

$$min_H \|X - WH^T\|_F^2 + I_{\mathbb{R}^+}(H)$$

where

$$I_{\mathbb{R}^+}(x) = \begin{cases} 0, & \text{if } x \in \mathbb{R}^+ \\ \infty, & \text{otherwise.} \end{cases}$$

The proximal operator is defined as

$$prox_\alpha(x) = max(0, x)$$

The W update if formulated in the same way. Optimization is done using alternating least squares (**block coordinate descent**)

Example: Text Mining

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
scsi	ca	israel	cmu	god
drive	team	israeli	andrew	jesus
ide	game	armenian	mellon	bible
controller	hockey	turkish	carnegie	christian
drives	players	armenians	pittsburgh	christians

Table: Topics generated using the 20-newsgroups dataset

Example: Text Mining

DOES GOD LOVE YOU?

Q. What kind of question is that? Anyone who can read sees signs, tracts, books, and bumper stickers that say, "God Loves You." Isn't that true?

A. It is true that God offers His love to the whole world, as we read in one of the most quoted verses in the Bible:

> For God so loved the world, that he gave his only begotten Son, that whosoever believeth in him should not perish, but have everlasting life. John 3:16

However, God's love is qualified. The Bible says:

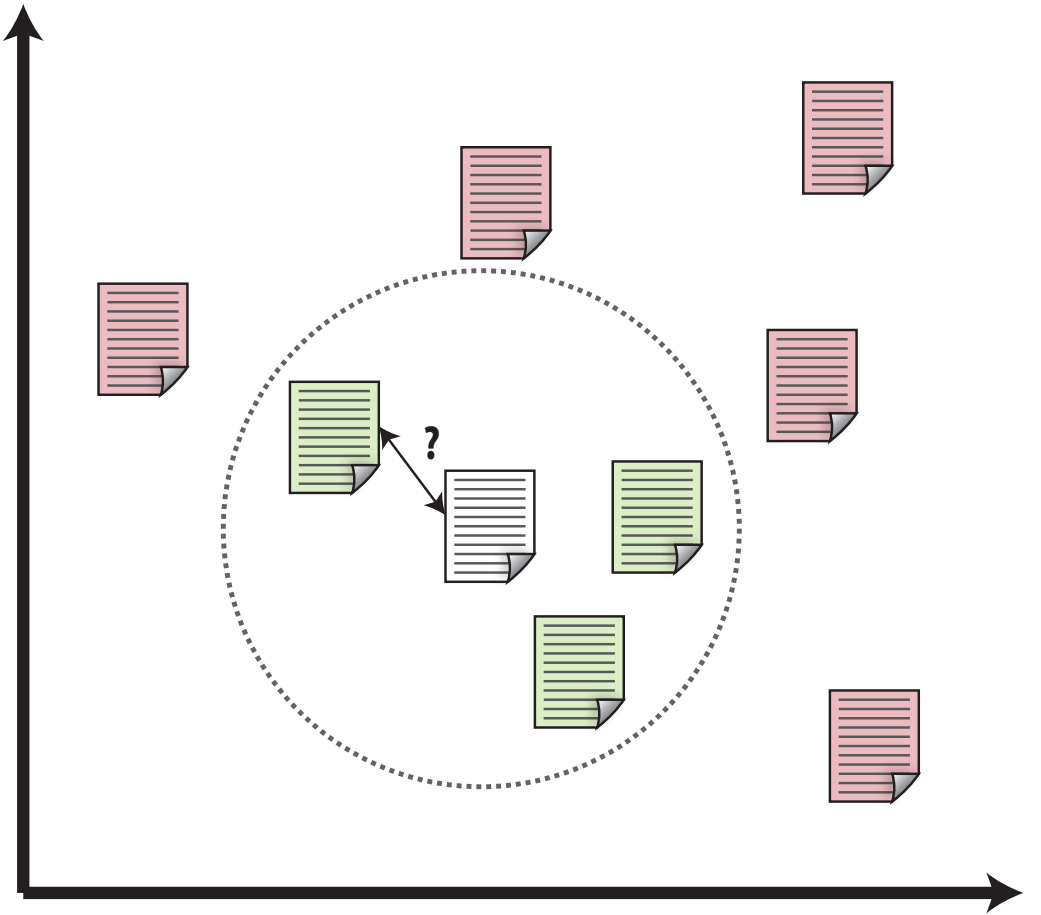
> The way of the wicked is an abomination unto the LORD: but he loveth him that followeth after righteousness. Proverbs 15:9

> For the LORD knoweth the way of the righteous: but the way of the ungodly shall perish. Psalm 1:6

Demo

Distance Metric Learning

Metric Learning (Huang, 2009)



Metric Learning

Metric Learning is the task of learning a distance function over objects that obeys **four axioms**:

- non-negativity: $dist(x_1, x_2) \geq 0$
- symmetry: $dist(x_1, x_2) = dist(x_2, x_1)$
- identity: $dist(x_1, x_1) = 0$
- triangle inequality: $dist(x_1, x_3) \leq dist(x_1, x_2) + dist(x_2, x_3)$

Specifically, we will learn a distance metric

$$D_W(x_1, x_2)^2 = (x_1 - x_2)^T W(x_1 - x_2)$$

where $W \in S_+^d$. Looks like Mahalanobis distance. If W is the **identity matrix**, then the distance becomes **euclidean distance**.

Intuition: Push the distance between similar vectors to zero and make the distance between dissimilar vectors large.

$$min_W \sum_{i,j,k \in T} max(0, 1 + x_{ij}^T W x_{ij} - x_{ik}^T W x_{ik}) + I_{S_+^d}(W)$$

where

$$x_{ij} = (x_i - x_j).$$

Nonzero when $dist(x_i, x_j) + 1 > dist(x_i, x_k)$.

$$prox_\alpha(x) = P^T \Lambda_+ P$$

The *prox* operator sets the eigenvalues of W to be non-negative (Henrion and Malick, 2012).

Metric Learning (Huang, 2009)

Data Set	GSML	SML	Xing	LMNN	Euclidean
Wine (%)	$\mathbf{96.45 \pm 1.97}$	95.00 ± 1.46	91.06 ± 3.12	94.74 ± 1.95	88.76 ± 2.85
Breast (%)	$\mathbf{90.89 \pm 2.71}$	88.12 ± 2.97	84.16 ± 5.01	87.37 ± 1.16	83.82 ± 5.21
Pima (%)	64.25 ± 3.06	61.31 ± 3.26	56.59 ± 1.84	$\mathbf{64.38 \pm 3.90}$	59.67 ± 2.75
Iris (%)	$\mathbf{97.53 \pm 2.31}$	96.22 ± 1.94	95.09 ± 3.76	96.73 ± 3.22	93.87 ± 3.34
Ionosphere (%)	$\mathbf{75.21 \pm 3.50}$	70.05 ± 2.95	59.95 ± 3.17	74.95 ± 4.57	60.82 ± 3.77
Balance (%)	84.21 ± 3.09	77.51 ± 3.21	84.01 ± 3.02	$\mathbf{84.80 \pm 3.71}$	66.21 ± 3.02

Figure: Triplet Test Accuracy Comparison

Conclusion

Generalized gradient descent is a great tool for optimization.

Other interesting applications (proximal operators) include:

- (Overlapping) Group Lasso
  - Sentence/Topic Regularization (Yogatama and Smith, 2014)
  - Finding Representive Exemplars (Elhamifar et al., 2013)
- Robust Regression
- Total Variation Denoising
  - Image denoising (Chambolle et al., 2009)
- Generalized Lasso
  - Network LASSO (Hallac et al., 2015)

Finally: Operators can be combined. For example, it is common to combine the **non-negative** constraint with **l1 regularization** for matrix factorization.

Thank you!

https://github.com/AnthonyMRios/proximal_methods_notes