

Proximal methods:

consider

$$f(x) = g(x) + h(x)$$

convex, diff,
domain is \mathbb{R}^n

askward: convex
but not necessarily
diff and domain
might be smaller

gradient descent builds a second order
model at x_n , and minimizes that.

$w(x) \leftarrow$ conv, diff

$$x_{n+1} = \underset{z}{\operatorname{argmin}} \left[w(x_n) + (\nabla w)^T (z - x_n) + \frac{1}{2t} \|z - x_n\|^2 \right]$$

$$= x_n - t \nabla w$$

Strategy:

approx g , ignore h

$$x_{n+1} = \underset{z}{\operatorname{argmin}} \left[g(x_n) + (\nabla g)^T(z - x_n) + \frac{1}{2t} \|z - x_n\|_2^2 + h(z) \right]$$

$$= \underset{z}{\operatorname{argmin}} \left[\frac{1}{2t} \|z - (x_n - t \nabla g)\|_2^2 + h(z) \right]$$

[you should check this - expand terms!]

- so z
- should be close to g.d. pred
 - make $h(z)$ smaller.

Define:

$$\text{prox}_{h,t}(x) = \underset{z}{\text{argmin}} \frac{1}{2t} \|x - z\|_2^2 + h(z)$$

proximal gradient descent:

$$x_{n+1} = \text{prox}_{h,t_n}(x_n - t_n \nabla g(x_n))$$

at its most useful when h is such
that prox_{h,t_n} is easy

(Quite common — examples follow)

ISTA :

the lasso problem.

minimize β $\frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$

↑ g -easy

↑ h -awkward.

we solved this by splitting and ADMM.

$$\text{prox}_t(\beta) = \underset{z}{\text{argmin}} \frac{1}{2t} \|\beta - z\|_2^2 + \lambda \|z\|_1$$

(which should look familiar)

notice $\text{prox}_t(\beta) = \underset{z}{\text{argmin}} \frac{1}{2t} \sum_u (\beta_u - z_u)^2 + \lambda \sum_u \text{abs}(z_u)$

so it splits over the components of z, β .

consider one term

$$\underset{z_u}{\text{argmin}} \frac{1}{2t} (\beta_u - z_u)^2 + \lambda \text{abs}(z_u)$$

solu occurs when

$$0 \ni \partial_{z_u}$$

$$\text{so } 0 \ni -\frac{1}{t} (\beta_u - z_u) + \lambda \partial_{z_u} \text{abs}(z_u).$$

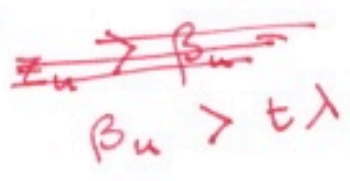
Cases:

$$\frac{z_u > 0}{\text{then}}$$

$$-\frac{1}{t} (\beta_u - z_u) + \lambda = 0$$

$$\text{so } z_u = \beta_u - t\lambda.$$

and so



$$\frac{z_u < 0}{\text{then}}$$

$$z_u = \beta_u + t\lambda$$

$$\frac{z_u = 0}{\text{then}}$$

$$0 \ni -\frac{1}{t} (\beta_u - \frac{z_u}{0}) + \lambda [-1; 1]$$

↑ a whole interval!

$$\text{so } 0 \ni \beta_u \bar{\in} \lambda t [-1; 1]$$

$$\text{so } \lambda \leq \beta_u \leq \lambda t$$

This should all look fairly familiar (6)

Notation

$$S_{\lambda t}(\beta)$$

← Shrinkage operator!

$$[S_{\lambda t}(\beta)]_u = \begin{cases} \beta_u - \lambda t & , \beta_u > \lambda t \\ 0 & , -\lambda t \leq \beta_u \leq \lambda t \\ \beta_u + \lambda t & , \beta_u < -\lambda t. \end{cases}$$

So proximal gradient descent is

$$\beta_{n+1} = S_{\lambda t} \left[\beta_n + t X^T (y - X\beta) \right]$$

(iterative soft-thresholding algorithm — ISTA)

Convergence :

(7)

assume : g convex, diff, $\text{dom}(g) = \mathbb{R}^n$,
 ∇g is Lipschitz, const L .

h convex, prox can be evaluated

Proximal gradient descent with fixed
step size $t \leq \frac{1}{L}$ satisfies

$$f(x_n) - f^* \leq \frac{\|x_0 - x^*\|_2^2}{2tn}$$

notice: as before, this bounds function
values, rather than arguments
matches gradient descent...

Matrix completion:

(8)

we have $Y \in \mathbb{R}^{(m \times n)}$

But observe only some entries.

we want to fill in missing entries

- general case isn't possible

- but assume Y is known to have

small nuclear norm

(Ky-Fan norm)

$$\|B\|_{KF} = \sum_i \sigma_i(B)$$

where σ_i are singular values
of B AND $\sigma_i \geq 0$

General idea: this norm favors
matrices with low rank

Why (applications).

③

- ~~abstract~~
This is a generalization of the case
where $Y = AB$ and entries are missing

(in this case, rank of Y is known)

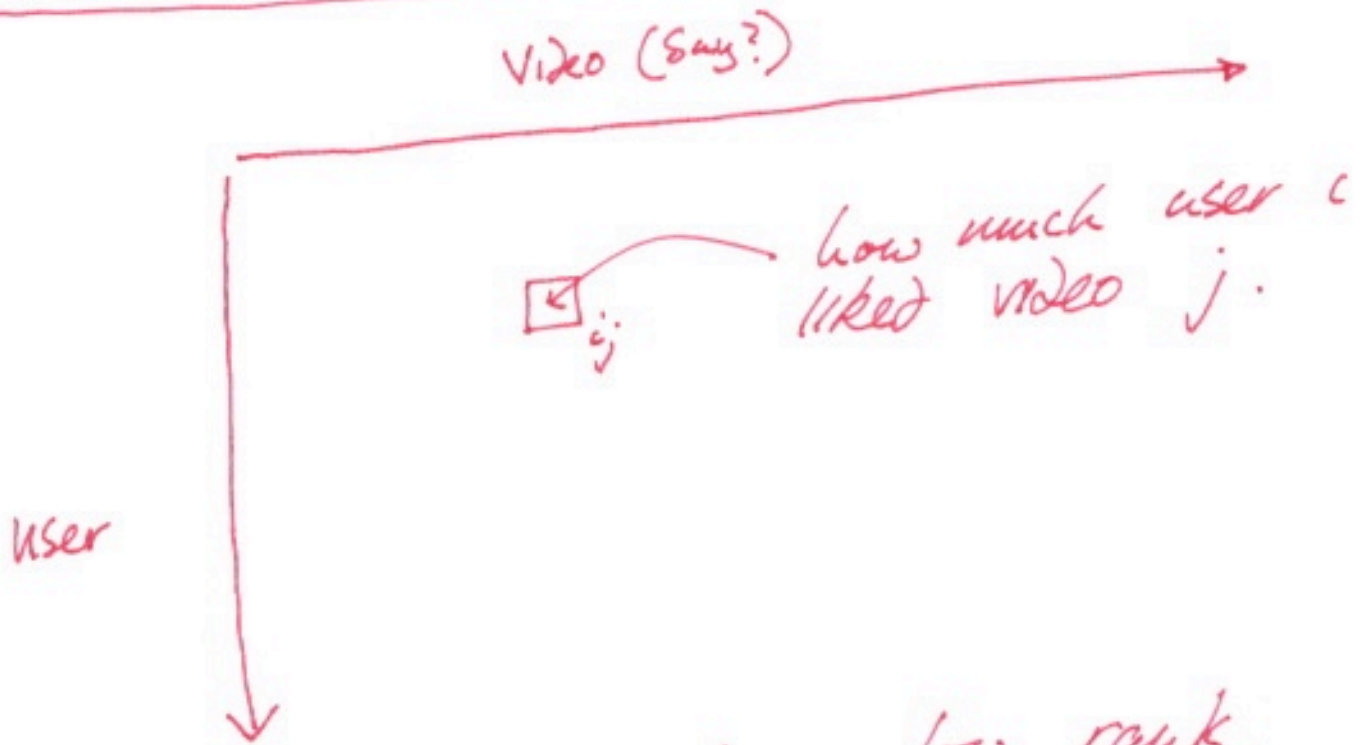
- Classic example of $Y = AB$
- affine structure from motion with missing observations

General case:

- recommender systems
- smoothing word counts

Recommender Systems

(10)



This matrix should have low rank.

- many pairs of users will have about the same tastes
→ many pairs of rows will be similar
- two "similar" [horror; history; etc] videos will be liked by about the same set of people
→ many pairs of cols will be similar.

~~and~~

AND:

- many cells were not observed
- knowing values of those cells is worthwhile.

Smoothing word counts:

(11)

word

Document

x_{ij} · how many times word j appears in doc i

· In principle we could know all entries of this Document-Word matrix.

· But 0 - counts are somewhat unreliable

· many words have similar meaning

"car", "motorcar", "automobile"

- if "car" appears, there should likely be a small count for "motorcar" etc.
- this means that the matrix should have low rank, too.

• We are now interested in

$$\min_B \left[\text{difference between observed } Y \text{ and CSP } B \right]^2 + \lambda \|B\|_{KF}$$

• write $P_{\Omega}(B) = \begin{cases} B_{ij} & \text{if } (i,j) \text{ was observed} \\ 0 & \text{otherwise} \end{cases}$

$$\text{prox}_t(B) = \underset{Z}{\text{argmin}} \left[\frac{1}{2t} \|B - Z\|_F^2 + \lambda \|Z\|_{KF} \right]$$

14

Claim:

$$\text{prox}_t(B) = S_{\lambda t}(B)$$

matrix soft thresholding

$$S_{\lambda t}(B) = U \Sigma_{\lambda t} V^T$$

where $B = U \Sigma V^T$ (SVD)

and $\Sigma_{\lambda t}$ is diagonal,

$$\left[\Sigma_{\lambda t} \right]_{ii} = \max(\sigma_{ii} - \lambda t, 0)$$

(proof in Tibshirani notes on web page).

so

$$B_{n+1} = S_{\lambda t} \left(B_n + t (P_{\Omega}(Y) - P_{\Omega}(B)) \right).$$

can choose $t=1$, get

$$B_{n+1} = S_{\lambda t} \left(P_{\Omega}(Y) + P_{\Omega}^{\perp}(B) \right)$$

(where $P_{\Omega}^{\perp}(B)$ projects onto unobserved set)

This is soft-impute (ref in notes).

Now assume we want

(16)

$$\min_{x \in C} g(x)$$

where $g(x)$ is diff, convex
and defined on \mathbb{R}^n
 C is convex

write:

$$\min_{x \in \mathbb{R}^n} g(x) + I_C(x)$$

$$I_C(x) = \begin{cases} 0 & x \in C \\ \infty & \text{otherwise} \end{cases}$$

now:

$$\begin{aligned} \text{prox}_t(x) &= \underset{z}{\text{argmin}} \frac{1}{2t} \|x - z\|_2^2 + I_C(x) \\ &= \underset{z \in C}{\text{argmin}} \|x - z\|_2^2 \end{aligned}$$

So that

$$\text{prox}_t(x) = P_C(x)$$

- project x onto C
- closest point in C to x .

so

$$x_{n+1} = P_C(x_n - t \nabla g(x_n))$$

projected gradient descent

useful if you can compute P_C

- box, sphere, etc.