Acceleration

- we know that past gradients can be useful

- work with two sequences.

  x_k ← the thing we want

  y_k ← auxiliary sequence ↙ looks like GD.

$$x_k = y_{k-1} - s \nabla f(y_{k-1})$$

$$y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$$

This improves the convergence rate, and yields

$$f(x_k) - f^* \leq \mathcal{O}\left(\frac{\|x_0 - x^*\|^2}{s k^2}\right)$$

for fixed step size $< \frac{1}{L}$ ← Lipschitz const.

although the rate improves, one doesn't always benefit.

- seems not to be widely used in practice

BUT some cases are very helpful

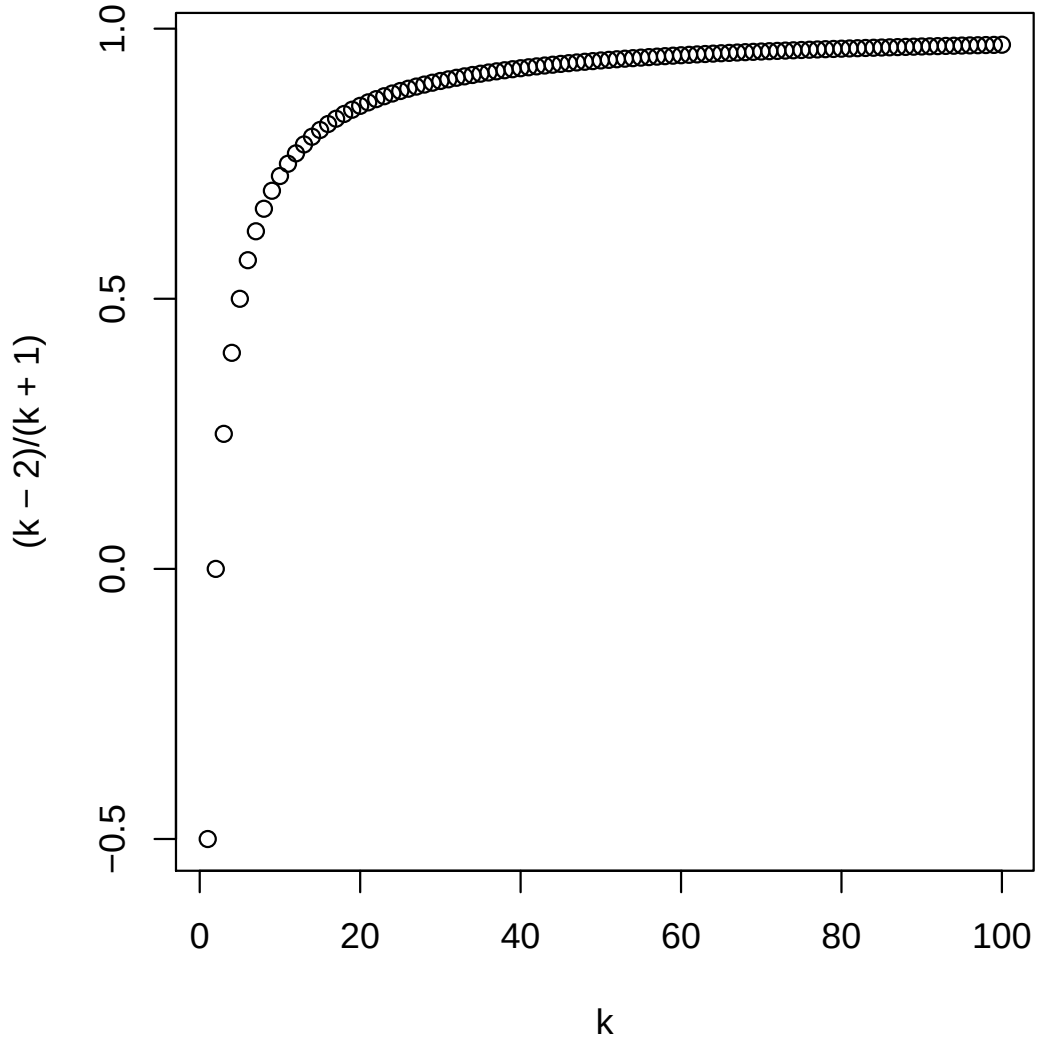
In proximal methods, this acceleration becomes

$$v = x_{k-1} + \left(\frac{k-2}{k+1}\right)\left(x_{k-1} - x_{k-2}\right)$$

$$x_k = \text{prox}_{t_k}\left(v - t_k \nabla g(v)\right)$$

- $k=1$ — proximal gradient update
- v is affected by x_{k-2} ("momentum")
- $k=0$ → accelerated gradient

Momentum weights:



Acceleration for Lasso

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta_1\|.$$

ISTA:

$$\beta_k = \text{Shrink}_{\lambda t_k}\left[\beta_{k-1} + t_k X^T(y - X\beta_{k-1})\right]$$

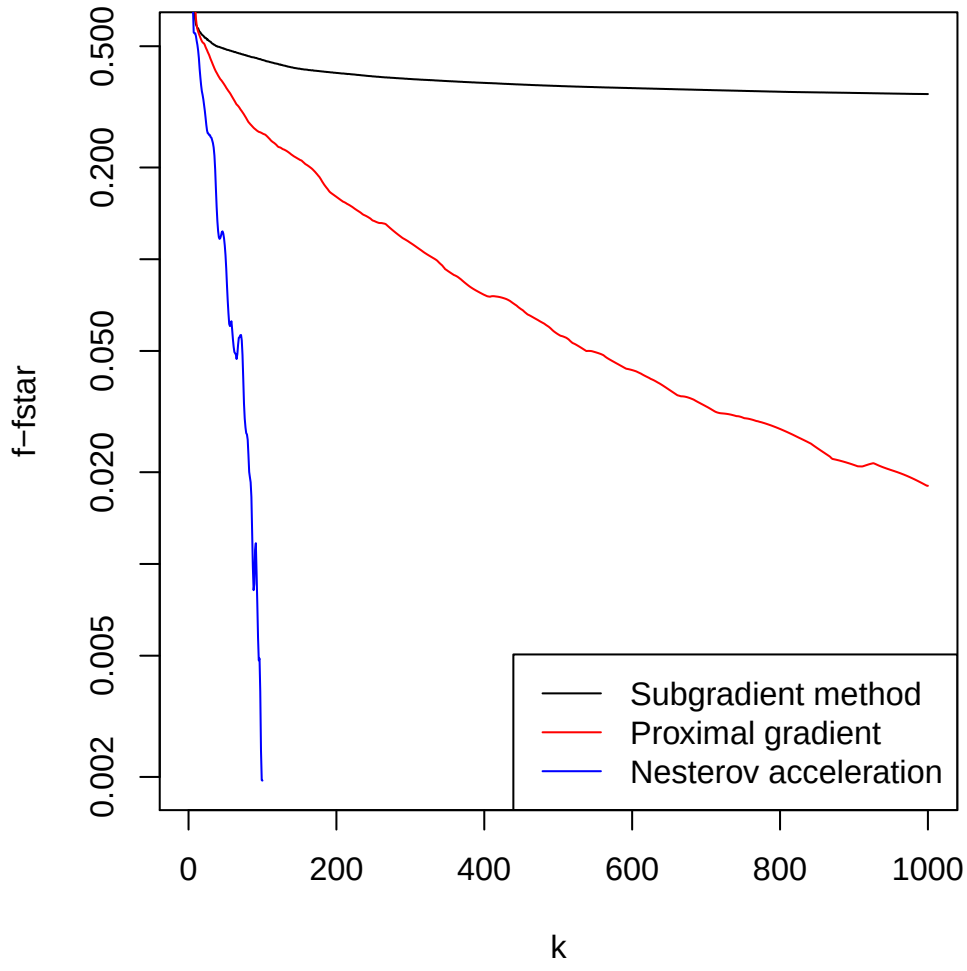
FISTA:

as +

$$v = \beta_{k-1} + \left(\frac{k-2}{k+1}\right)(\beta_{k-1} - \beta_{k-2})$$

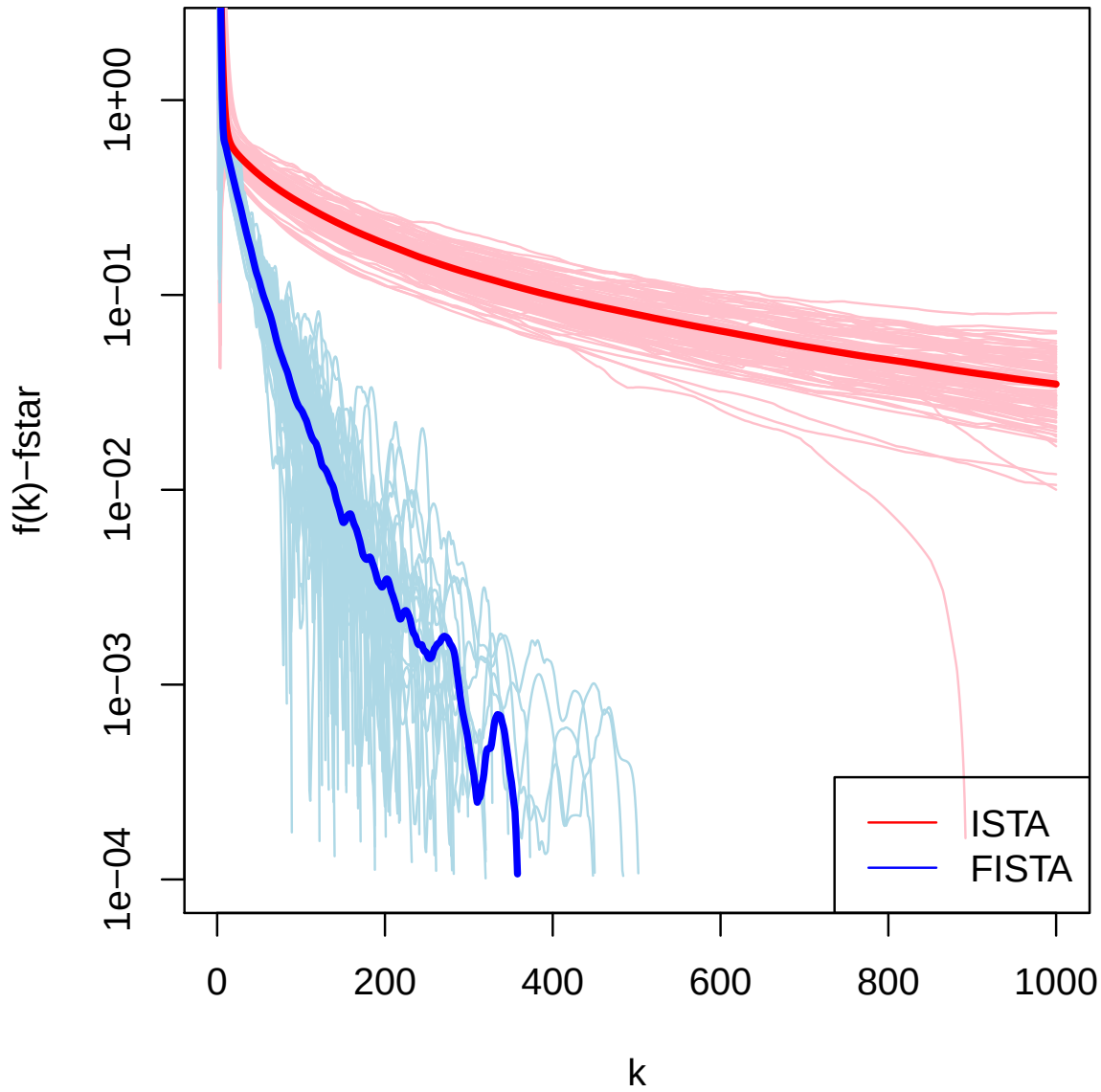
$$\beta_k = \text{Shrink}_{\lambda t_k}\left[v + t_k X^T(y - Xv)\right]$$

Back to lasso example: acceleration can really help!

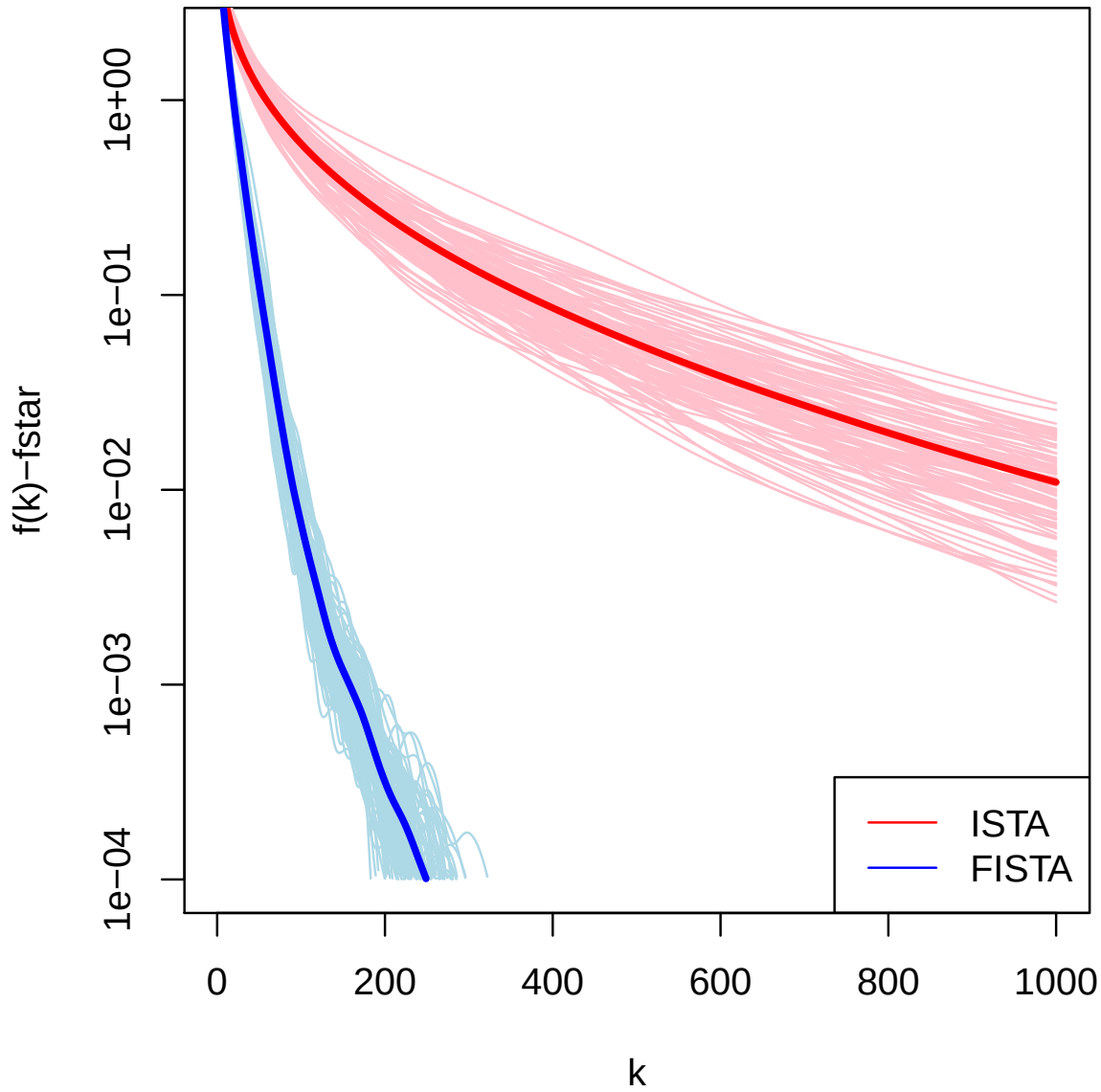


Note: accelerated proximal gradient is not a descent method

Lasso regression: 100 instances (with $n = 100$, $p = 500$):



Lasso logistic regression: 100 instances ($n = 100$, $p = 500$):



Notice, for big k

$$v \approx 2\beta_{k-1} - \beta_{k-2}$$

$$= \underbrace{\beta_{k-2} + 2(\beta_{k-1} - \beta_{k-2})}$$

see this as a form of overprediction; $\beta_{k-1} = \beta_{k-2} + \Delta$ and $v = \beta_{k-2} + 2\Delta$

This *really* helps in practice.

For Lasso, practice shows that acceleration is very helpful with "cold start" not so much with "warm start"

e.g. lasso with some λ, unknown $\beta \longrightarrow$ FISTA

- we've done λ_i, ($\beta(\lambda_i)$) and now want $\lambda_{i+1} \approx \lambda_i$; $\longrightarrow$ ISTA, starting at $\beta(\lambda_i)$

Acceleration affects computation

- matrix completion, $t=1$

$$B_{k+1} = \mathcal{S}_\lambda\left[P_\Omega(Y) + P_\Omega^\perp(B)\right]$$

↳ computing this requires an SVD.

Notice:

- expect $P_\Omega(Y)$ is sparse (cause we observed few values)
- expect $P_\Omega^\perp(B)$ is low rank (otherwise we couldn't complete)

$\Rightarrow$ easy SVD

But (accelerated)

$$V_k = B_{k-1} + \frac{k-2}{k+1}\left(B_{k-1} + B_{k-2}\right)$$

$$B_k = S_\lambda\left[P_\Omega(Y) + P_\Omega^{\perp}(V)\right]$$

This may not have low rank — whatever we gain in acceleration, we could lose in SVD.