Coordinate and stochastic variations early termination, rates, acceleration

Pontus Giselsson

Learning goals

- Understand scaling in coordinate gradient method
- Know adaptive scaling variations of SGD (Adagrad, Adam)
- Know Polyak-Ruppert averaging
- Know about implicit regularization in GD and SGD
- Know that early termination can be regularization
- Know about different rates and what algorithms achieve
- Know about the accelerated proximal gradient method

Coordinate Descent

Scaling

Scaled coordinate descent

- In coordinate descent, used the following model of the smooth f:

$$\hat{f}_{x^k}(y) = f(x^k) + \nabla f(x^k)^T (y - x^k) + \frac{1}{2\gamma_k} \|y - x^k\|_2^2$$

- We can instead use scaled model with fixed norm $\|\cdot\|_H$:

$$\hat{f}_{x^k}(y) = f(x^k) + \nabla f(x^k)^T (y - x^k) + \frac{1}{2\gamma_k} \|y - x^k\|_H^2$$

- We still assume $g(x) = \sum_{i=1}^n g_i(x_i)$ is separable
- Convergence analysis goes through identically in this setting

The algorithm – Single coordinate case

- Recall the coordinate-selection set

$$C_j^x = \{y \in \mathbb{R}^n : y_l = x_l \text{ for all } l \neq j\}$$

  i.e., $y_l = x_l$ for all coordinates $l \neq j$, only y_j is free
- The coordinate update with new model is, select j at random and:

$$\begin{aligned}
x^{k+1} &= \operatorname*{argmin}_y (f(x^k) + \nabla f(x^k)^T (y - x^k) + \tfrac{1}{2\gamma_k} \|y - x^k\|_H^2 + g(y) + \iota_{C_j^{x^k}}(y)) \\
&= \operatorname*{argmin}_{y_j, y_l = x_l^k} (f(x^k) + \nabla f(x^k)^T (y - x^k) + \tfrac{1}{2\gamma_k} \|y - x^k\|_H^2 + g(y)) \\
&= \operatorname*{argmin}_{y_j, y_l = x_l^k} (\nabla f(x^k)_j^T (y_j - x_j^k) + \tfrac{1}{2\gamma_k} H_{jj} (y_j - x_j^k)^2 + g_j(y_j)) \\
&= \operatorname*{argmin}_{y_j, y_l = x_l^k} (g_j(y_j) + \tfrac{H_{jj}}{2\gamma_k} \|y_j - (x_j^k - \tfrac{\gamma_k}{H_{jj}} \nabla f(x_k)_j)\|_2^2) \\
&= \begin{cases} \operatorname{prox}_{\frac{\gamma_k}{H_{jj}} g_j}(x_j^k - \frac{\gamma_k}{H_{jj}} \nabla f(x^k)_j) & \text{for coordinate } j \\ x_l^k & \text{for all coordinates } l \neq j \end{cases}
\end{aligned}$$

  where H_{jj} is j:th diagonal element of H
- Only difference to nominal method: divide step γ_k by H_{jj}
- Same iteration cost as nominal method (unlike gradient method)

Example – A quadratic

- Assume $f(x) = \frac{1}{2} x^T H x$ with H positive definite (can add $h^T x$)
- Let in addition $\gamma_k = 1$, then (as we have seen)

$$\hat{f}_{x^k}(y) = f(x^k) + \nabla f(x^k)^T (y - x^k) + \tfrac{1}{2} \|y - x^k\|_H^2 = f(y)$$

- Coordinate descent with this model and quadratic f therefore is

$$x^{k+1} = \operatorname*{argmin}_y (f(y) + g(y) + \iota_{C_j^{x^k}}(y)) = \operatorname*{argmin}_{y_j, y_l = x_l^k} (f(y) + g(y))$$

  that is, optimize problem itself w.r.t. randomly chosen variable!
- Implement as

$$x^{k+1} = \begin{cases} \operatorname{prox}_{\frac{\gamma_k}{H_{jj}} g_j(y)}(x_j^k - \frac{\gamma_k}{H_{jj}} \nabla f(x^k)_j) & \text{for coordinate } j \\ x_l^k & \text{for all coordinates } l \neq j \end{cases}$$

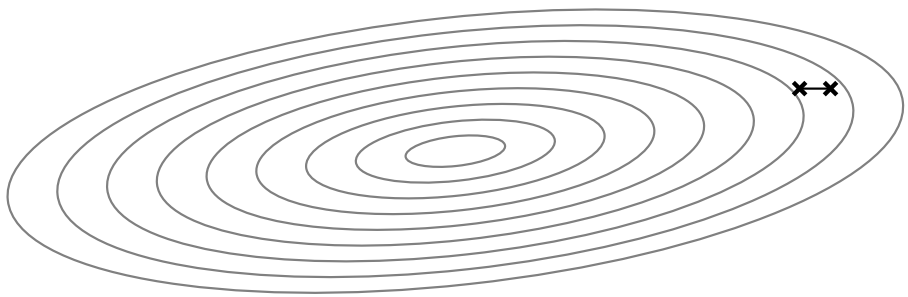
- Very efficient algorithm in this setting, e.g., dual SVM
- Can be used in subroutine in Newton proximal gradient method

Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

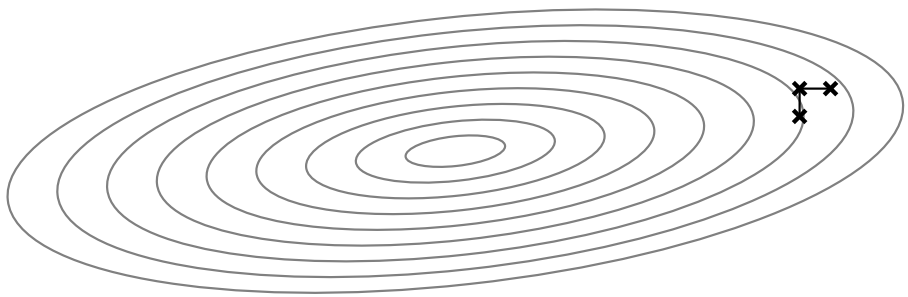


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

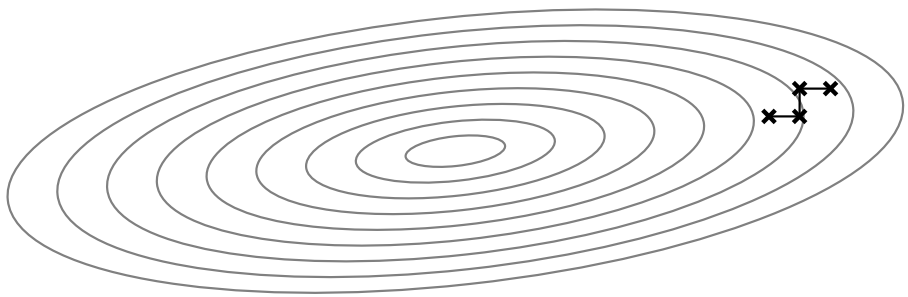


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

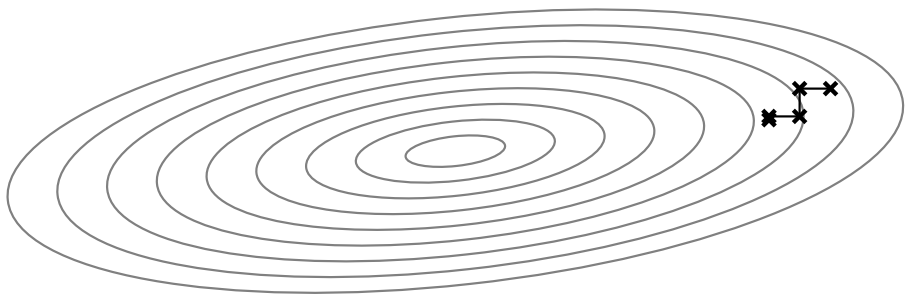


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

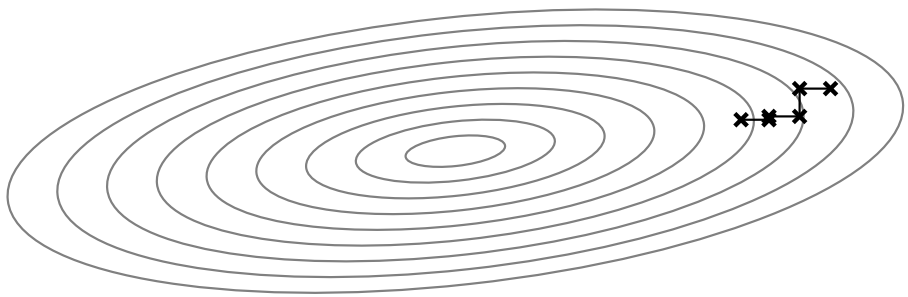


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

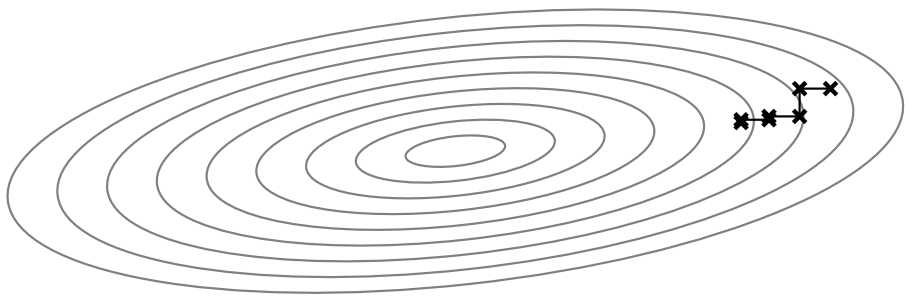


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

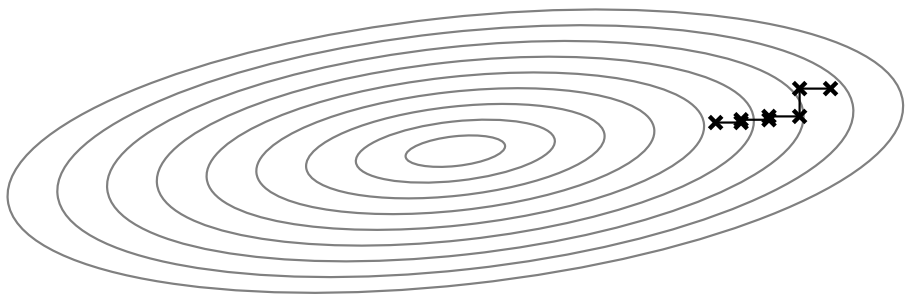


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

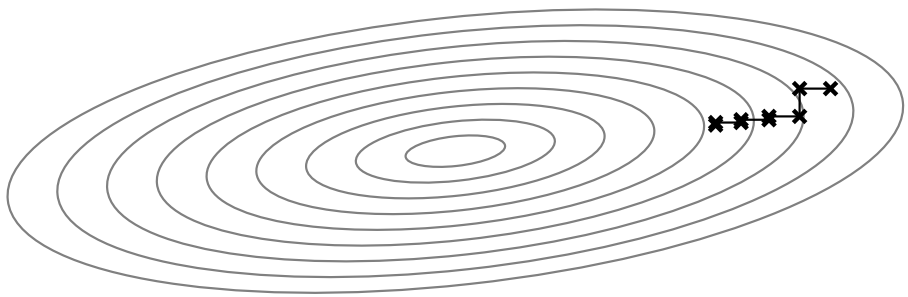


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

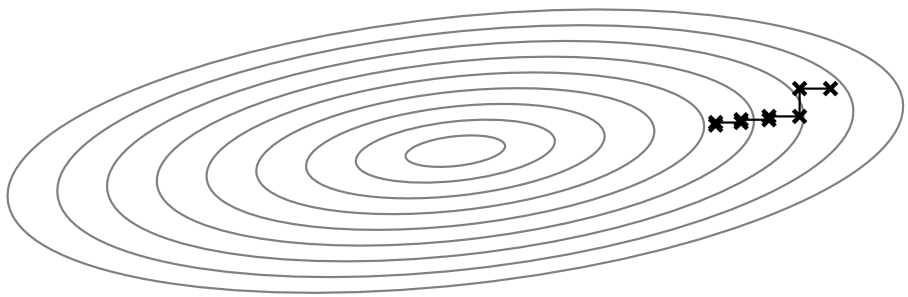


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = \frac{1}{\beta}$ and norm $\|\cdot\|_2$ in model

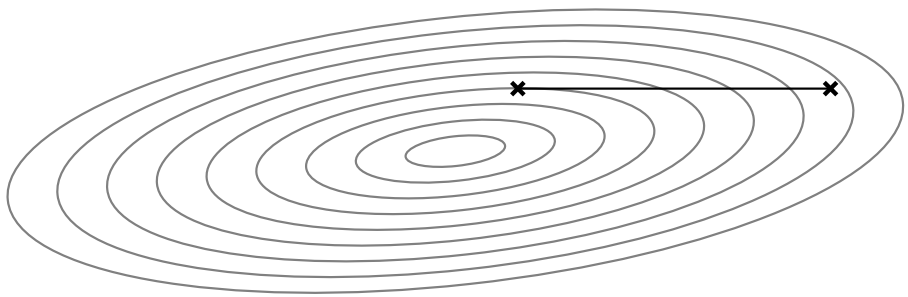


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model

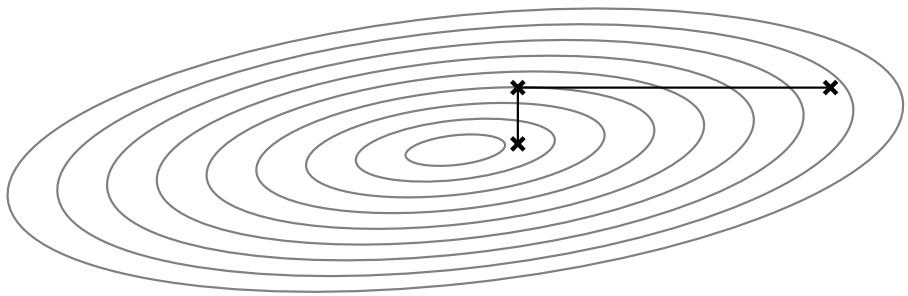


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model

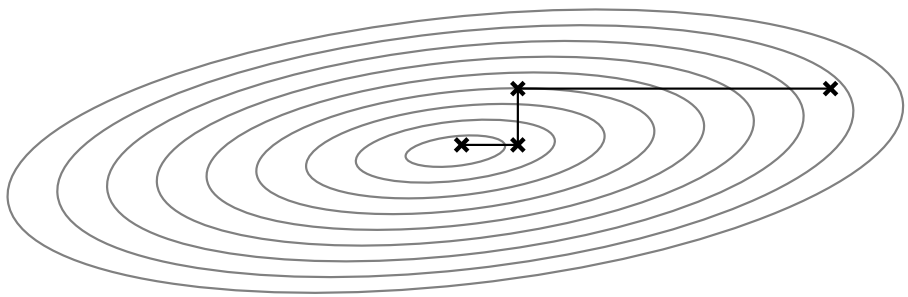


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model

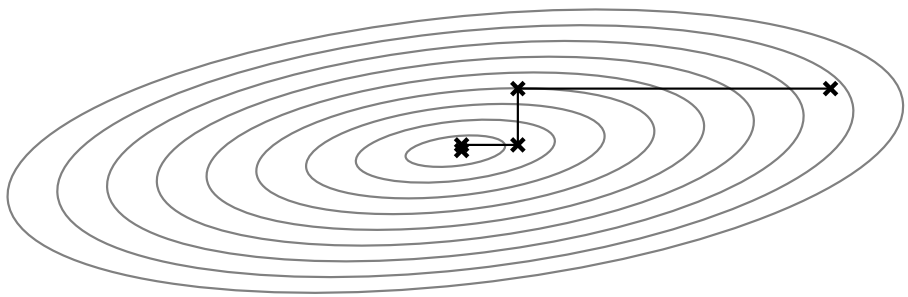


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model

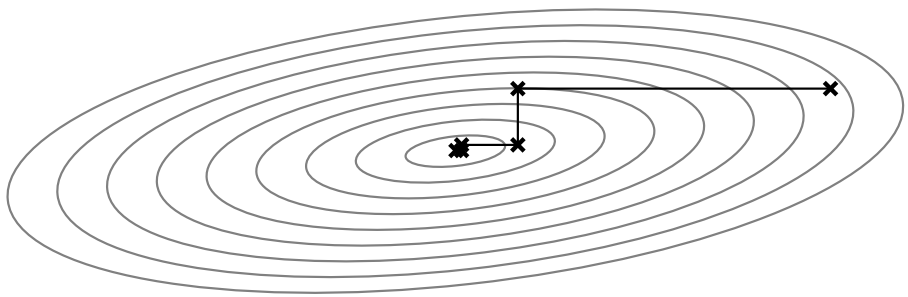


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model

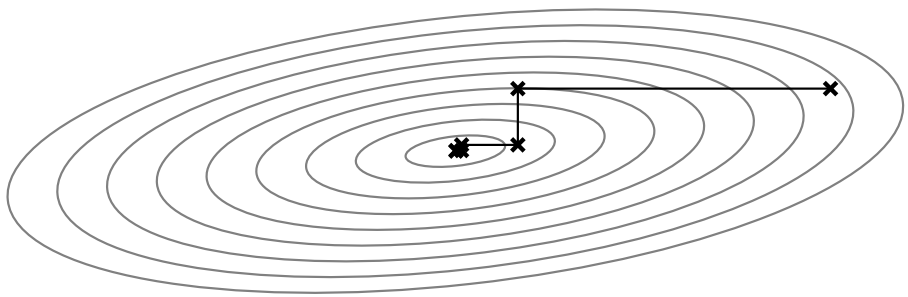


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model

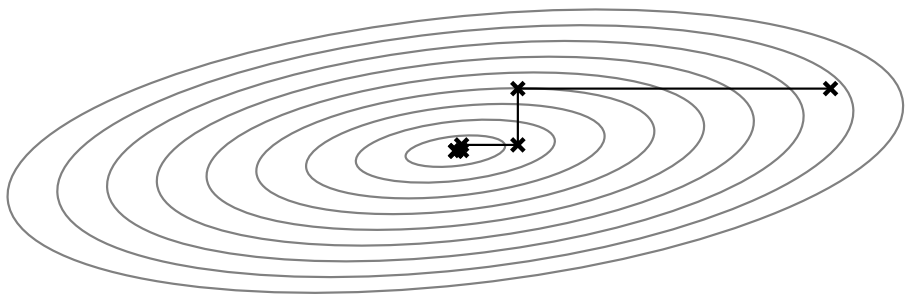


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model

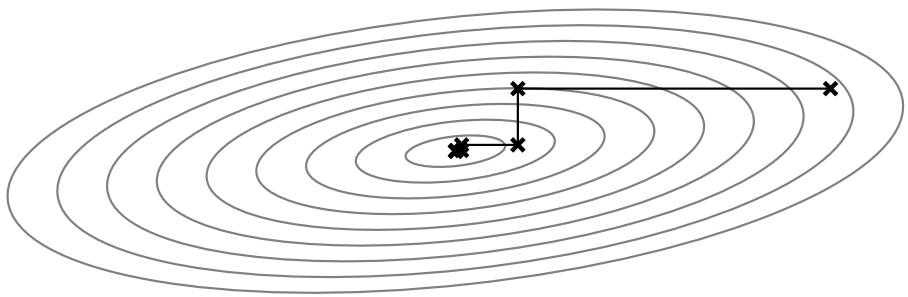


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model

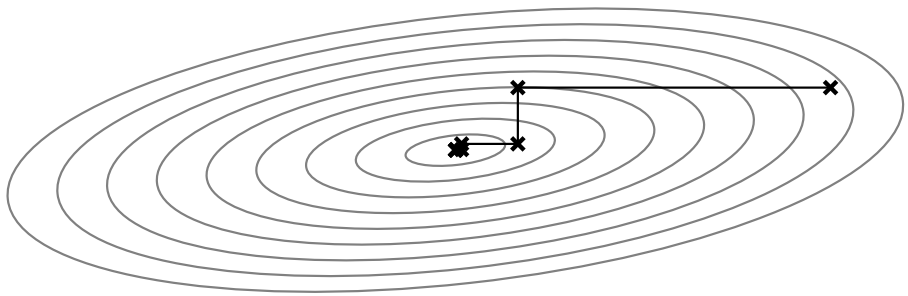


Better model – Coordinate minimization

- Coordinate descent on β-smooth quadratic problem

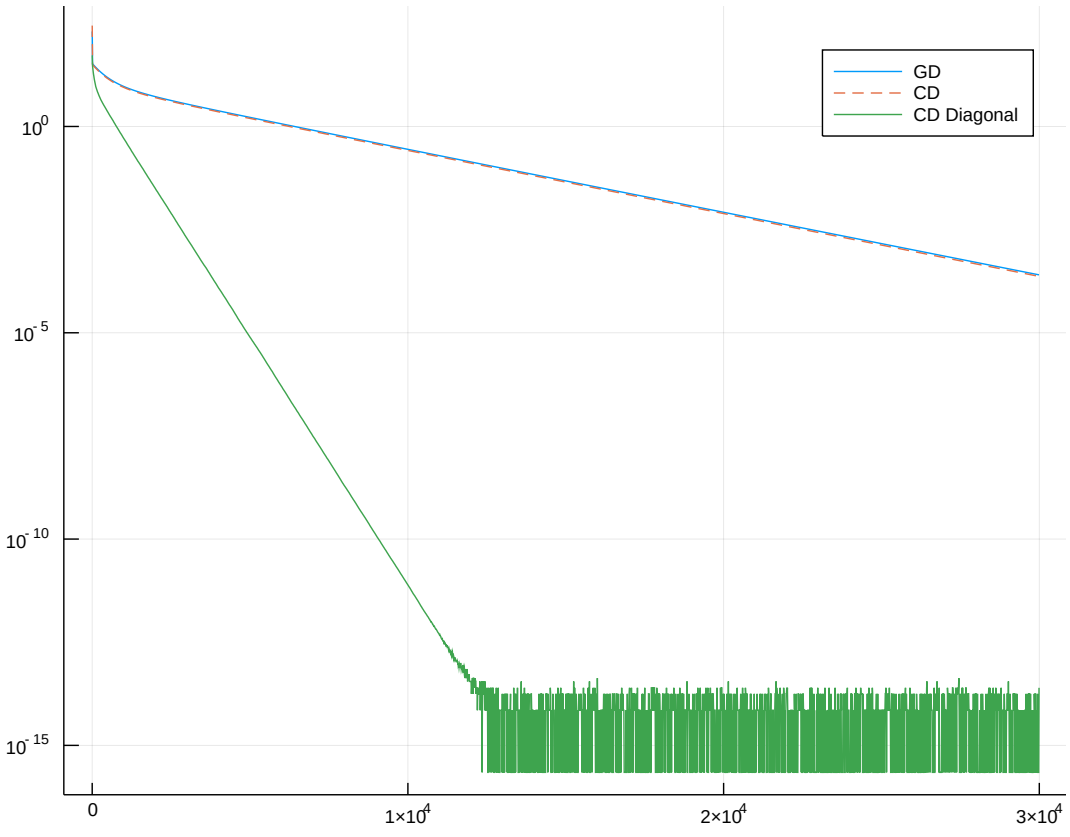
$$\underset{x}{\text{minimize}} \ \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size $\gamma = 1$ and norm $\|\cdot\|_H$ in model



Numerical example

- Least squares example from previous lecture
- Compares: gradient, coordinate, and scaled coordinate descent
- x axis normalized for fair comparison, y-axis is function value



Stochastic Gradient Descent

Scaling and Implicit Regularization

Adaptive diagonal scaling

- Diagonal scaling gives one step-size (learning rate) per variable
- Can be seen as SGD with diagonal metric H_k
- A few methods exists that *adaptively* select individual step sizes
  - Adagrad
  - RMSProp
  - Adam
  - Adamax
  - Adadelta
- Among these, Adagrad was first but Adam most popular
- Sometimes improve convergence compared to SGD
- Currently believed to generalize worse than SGD
- Will motivate Adagrad and show how Adam differs

Adagrad

- Adaptive methods solve problems

$$\underset{x}{\text{minimize}}\, f(x)$$

- Adagrad: Select step-size $\gamma > 0$ and iterate:
  1. g^k is subgradient or stochastic (sub)gradient of f at x_k
  2. Choose metric H_k
     - set $s_k = \sum_{l=1}^{k} (g^l)^2$
     - set $h_k = \epsilon \mathbf{1} + \sqrt{s_k}$
     - set $H_k = \gamma^{-1}\, \mathbf{diag}(h_k)$
  3. $x_{k+1} = x_k - H_k^{-1} g_k = x_k - \gamma g_k ./ (\epsilon \mathbf{1} + \sqrt{s_k})$

  where $\epsilon > 0$ is for numerical stability
- Adaptive and individual step size for each coordinate: $\frac{\gamma}{\epsilon + \sqrt{s_k}}$

Motivation for Adagrad

- Objective is adaptive diagonal scaling for improved convergence
- Adagrad: different analyses in different settings (all convex):
  - convergence rate in stochastic gradient descent
  - regret bound in online learning setting (in paper)
  - convergence rate in deterministic subgradient setting
- Convergence rate in subgradient setting with diagonal $H_k \succ 0$ is:

$$\min_{l=1,\ldots,k} (f(x_k) - f^\star) \leq \frac{1}{2k} \left(\sum_{l=1}^{k} \|g^l\|^2_{H_l{}^{-1}} + R^2 \, \mathbf{tr}(H_k) \right)$$

  where
  - $g^l \in \partial f(x_l)$ are subgradients (not nonsmooth functions!)
  - R is radius of ball containing iterates
  - $f^\star$ is optimal value

Motivation for Adagrad

- Let $g_j^{1:k} = (g_j^1, \ldots, g_j^k)$ be vector of historical j:th coordinates g_j^l
- Idea: Choose $H_k = \mathbf{diag}(h_k)$ to minimize bound in hindsight:
  - Fix $\mathbf{tr}(H_k) = \|h_k\|_1 = c_k$ to control second term in bound
  - Optimize first term with $H_l = H$ for all l and set $H_k = H$:

$$H_k = \operatorname*{argmin}_H \left(\sum_{k=1}^{k} \|g^l\|_{H^{-1}}^2 + \iota_{\{c_k\}}(\mathbf{tr}(H)) \right)$$
$$= \frac{c_k}{\|(\|g_1^{1:k}\|_2, \ldots, \|g_n^{1:k}\|_2)\|_2} \mathbf{diag}(\|g_1^{1:k}\|_2, \ldots, \|g_n^{1:k}\|_2)$$

- Select $c_k = \gamma^{-1} \|(\|g_1^{1:k}\|_2, \ldots, \|g_n^{1:k}\|_2)\|_2$ for some $\gamma > 0$ to have

$$H_k = \gamma^{-1} \mathbf{diag}(\|g_1^{1:k}\|_2, \ldots, \|g_n^{1:k}\|_2)$$

- Intuition:
  - Reduce step size for coordinates with many steep (large) gradients
  - Increase step size for coordinates with many flat (small) gradients

Adagrad – Convergence

- It can be shown that $\sum_{l=1}^{k} \|g^l\|^2_{H_l^{-1}} \leq 2\gamma\, \mathbf{tr}(H_k)$
- If $\|g\|_\infty \leq G$, it can be shown that $\mathbf{tr}(H_k) \leq nG\sqrt{k}$
- The bound becomes

$$\begin{aligned}\min_{l=1,\ldots,k}(f(x_k) - f^\star) &\leq \tfrac{1}{2k}\left(\sum_{l=1}^{k} \|g^l\|^2_{H_l{}^{-1}} + R^2\, \mathbf{tr}(H_k)\right)\\ &\leq \tfrac{1}{2k}(2\gamma + R^2)\, \mathbf{tr}(H_k)\\ &\leq \tfrac{nG}{\sqrt{k}}(2\gamma + R^2)\end{aligned}$$

  and $f(x_k) - f^\star \to$ if R bounded
- In practice,

$$H_k = \gamma^{-1}(\epsilon I + \mathbf{diag}(\|g_1^{1:k}\|_2, \ldots, \|g_n^{1:k}\|_2))$$

  with some $\epsilon > 0$ is used for numerical stability

Variations – RMSprop and Adam

- Adagrad H_k often grows too fast $\Rightarrow$ step sizes decay too fast
- One approach: Let c_k grow slower than in Adagrad
- Another approach: Don't *sum* gradient square, estimate variance:

$$\hat{v}_k = b_v \hat{v}_{k-1} + (1 - b_v)(g^k)^2$$

  where $\hat{v}_0 = 0$, $b_v \in (0, 1)$, and g^k are *stochastic* gradients
- H_k is chosen (approximately) as standard deviation:
  - RMSprop: biased estimate $H_k = \mathbf{diag}(\sqrt{\hat{v}_k} + \epsilon)$
  - Adam: unbiased estimate $H_k = \mathbf{diag}(\sqrt{\frac{\hat{v}_k}{1-b_v^k}} + \epsilon)$

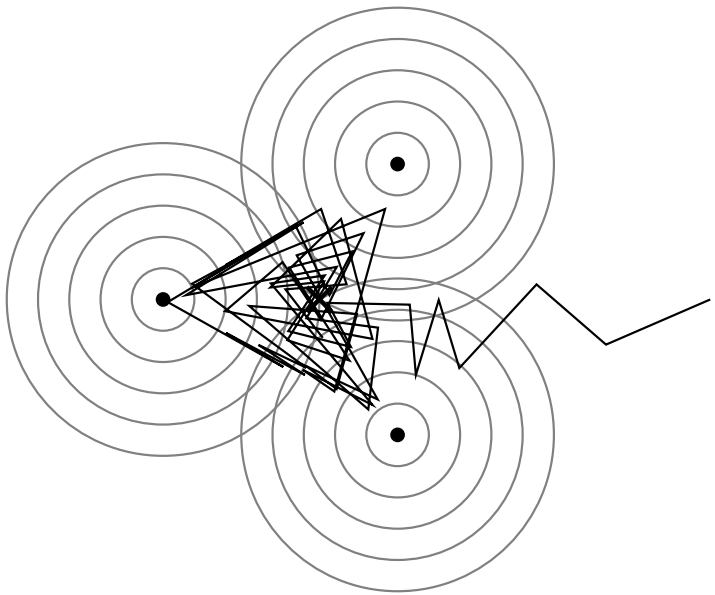
  which is much smaller H_k than in Adagrad $\Rightarrow$ longer steps
- Intuition:
  - Reduce step size for high variance coordinates
  - Increase step size for low variance coordinates
- Adam also filters stochastic gradients for smoother updates

Filtered stochastic gradients

- Let $m_0 = 0$ and $b_m \in (0, 1)$, and update

$$\hat{m}_k = b_m \hat{m}_{k-1} + (1 - b_m) g_k$$

- Adam uses unbiased estimate: $\frac{\hat{m}_k}{1 - b_m^k}$
- Does not improve convergence properties, but slower changes
- Problem from before, fixed-stepsize, without filtered gradient



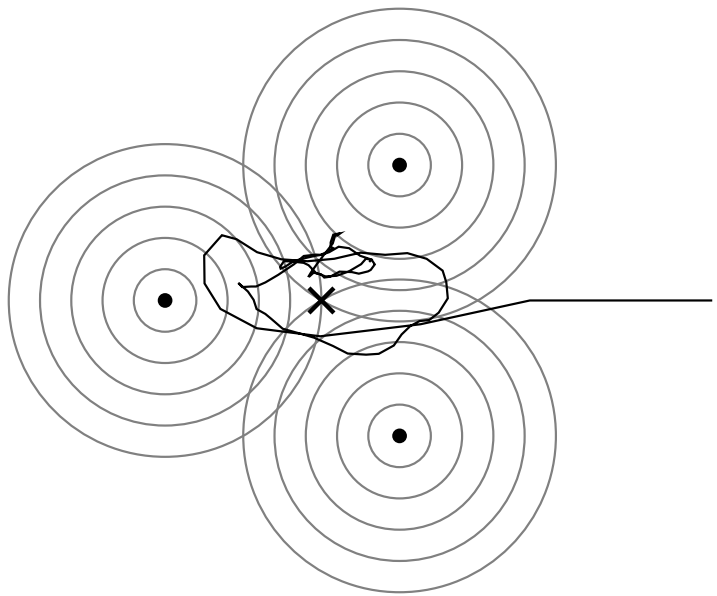
Levelsets of summands

Filtered stochastic gradients

- Let $m_0 = 0$ and $b_m \in (0, 1)$, and update

$$\hat{m}_k = b_m \hat{m}_{k-1} + (1 - b_m) g_k$$

- Adam uses unbiased estimate: $\frac{\hat{m}_k}{1 - b_m^k}$
- Does not improve convergence properties, but slower changes
- Problem from before, fixed-stepsize, with filtered gradient



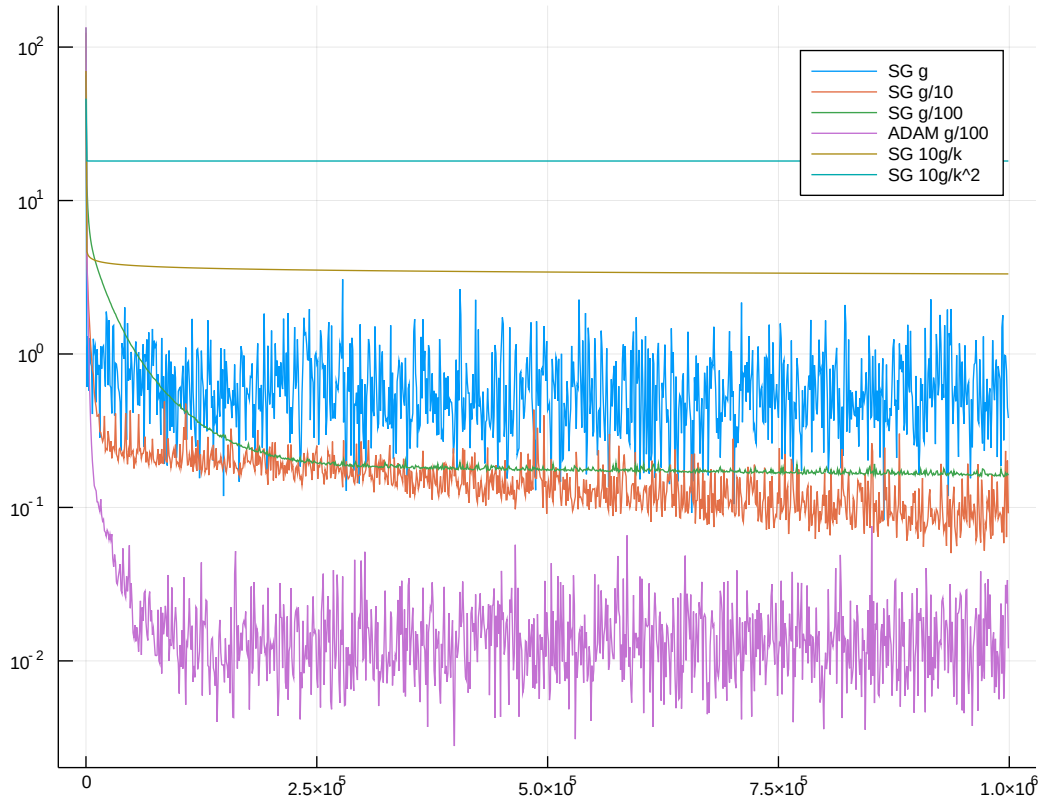
Levelsets of summands

Adam – Summary

- Initialize $\hat{m}_0 = \hat{v}_0 = 0$, $b_m, b_v \in (0,1)$, and select $\gamma > 0$
  1. $g_k = \widetilde{\nabla} f(x_k)$ (stochastic gradient)
  2. $\hat{m}_k = b_m \hat{m}_{k-1} + (1 - b_m) g_k$
  3. $\hat{v}_k = b_v \hat{v}_{k-1} + (1 - b_v) g_k^2$
  4. $m_k = \hat{m}_k / (1 - b_m^k)$
  5. $v_k = \hat{v}_k / (1 - b_v^k)$
  6. $x_{k+1} = x_k - \gamma m_k ./ (\sqrt{v_k} + \epsilon \mathbf{1})$
- Suggested choices $b_m = 0.9$ and $b_v = 0.999$
- Similar to Adagrad, but $\sqrt{v_k} \ll \sqrt{s_k} \Rightarrow$ longer steps
- May not work in deterministic setting (unlike Adagrad):
  - If method converges $\nabla f(x_k) \to 0$
  - Then $v_k \to 0$ and steps become very large
  - Needs noise and stochastic gradients to work well

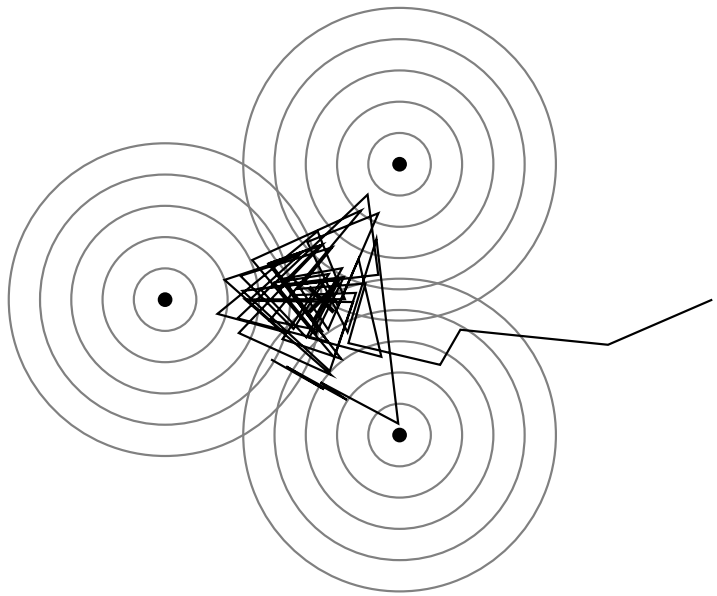
Adam – Example

- This is variation of problem from stochastic gradient lecture
- Had to engineer problem a bit to have Adam faster than SGD



Polyak-Ruppert averaging

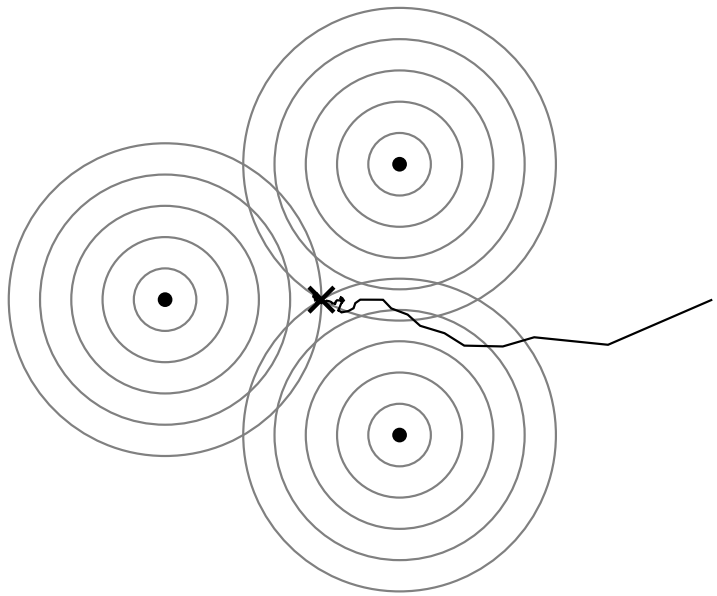
- Fixed-step SGD converges to noise ball (quite fast)
- Diminishing-step SGD converges (in some sense) to solution
- Polyak-Ruppert averaging:
  - Run SGD with (small enough) fixed step size
  - Output average of iterations instead of last iteration
- Example: SGD with constant stepsize



Levelsets of summands

Polyak-Ruppert averaging

- Fixed-step SGD converges to noise ball (quite fast)
- Diminishing-step SGD converges (in some sense) to solution
- Polyak-Ruppert averaging:
  - Run SGD with (small enough) fixed step size
  - Output average of iterations instead of last iteration
- Example: Average of SGD with constant stepsize



Levelsets of summands

Generalization and implicit regularization

- What is implicit regularization?
  - Assume infinitely many solutions on large manifolds exist
    - overparameterized neural networks
    - least squares with fewer examples than features
  - Algorithm selects solution with small(est) desired norm
- Good implicit regularization gives model with better generalization
- SGD is believed to have good implicit regularization
- Adaptive scaling methods believed to have worse

Implicit regularization – Least squares

- Consider *convex* least squares problem of the form

$$\underset{x}{\text{minimize}}\, \tfrac{1}{2}\|Ax - b\|_2^2$$

  where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $m < n$
- Solution set $X = \{x : Ax = b\} = \{x : \bar{x} + v\}$ where:
  - $\bar{x} \in X$ is such that $A\bar{x} = b$ (which exists since $m < n$)
  - v any vector such that $Av = 0$, i.e. in nullspace of A, $\mathcal{N}(A)$

  why? cost satisfies for all x: $\frac{1}{2}\|Ax - b\|_2^2 \geq 0$ and

$$\tfrac{1}{2}\|A(\bar{x} + v) - b\|_2^2 = \|A\bar{x} + Av - b\|_2^2 = \|b - b\|_2^2 = 0$$

- If $v \in \mathcal{N}(A)$ so is tv $\big(A(tv) = tAv = 0\big)$ for all $t \in \mathbb{R}$
- Since $m < n$, $\mathcal{N}(A)$ is (at least) $n - m$-dimensional subspace

Implicit regularization – Gradient method

- The gradient method satisfies for $\gamma \in (0, \frac{2}{\beta})$ *for all* solutions $x^\star$

$$\begin{aligned}\|x_{k+1} - x^\star\|_H^2 &= \|x_k - \gamma H^{-1}\nabla f(x_k) - x^\star\|_H^2 \\ &= \|x_k - x^\star\|_H^2 - 2\gamma\nabla f(x_k)^T(x_k - x^\star) + \gamma^2\|\nabla f(x_k)\|_{H^{-1}}^2 \\ &\leq \|x_k - x^\star\|_H^2 - (\tfrac{2\gamma}{\beta} - \gamma^2)\|\nabla f(x_k)\|_{H^{-1}}^2 \\ &\leq \|x_k - x^\star\|_H^2\end{aligned}$$

  where we use $\nabla f(x^\star) = 0$ and $\frac{1}{\beta}$-cocoercivity[1] of ∇f w.r.t. $\|\cdot\|_H$

- Comments:
  - Adagrad motivated by making residual $\|\nabla f(x_k)\|_{H^{-1}}^2$ small fast
  - Gradient method will converge to $\Pi_X^H(x_0)$ (next slide)
  - If $x_0 = 0$, will converge to minimum $\|\cdot\|_H$ element in X
  - Give implicit Tikhonov-type regularization if $H = I$
  - Can give not desired point if H very skewed

[1] Convexity is needed for ∇f to be cocoercive. That distance to solution nonincreasing holds only in convex setting.

Gradient method converges to $\Pi_X^H(x_0)$

For all $v \in \mathcal{N}(A)$ (and $\bar{x}$ defined before) it holds that

$$\|x_{k+1} - \bar{x}\|_H \leq \|x_k - \bar{x}\|_H \tag{1}$$

$$\text{and} \quad \|x_{k+1} - \bar{x} + v\|_H \leq \|x_k - \bar{x} + v\|_H \tag{2}$$

Condition (2) is (after squarring and expanding) equivalent to

$$\|x_{k+1} - \bar{x}\|_H^2 \leq \|x_k - \bar{x}\|_H^2 + 2(H(x_k - x_{k+1}))^T v$$

To not violate (1), $(H(x_k - x_{k+1}))^T v = 0$ ($-v$ and v are in $\mathcal{N}(A)$)

Therefore $(Hx_k)_{k\in\mathbb{N}}$ lives in affine subspace ($\lambda \in \mathbb{R}^m$ is arbitrary)

$$Hx_0 + (\mathcal{N}(A))^\perp = Hx_0 + A^T\lambda$$

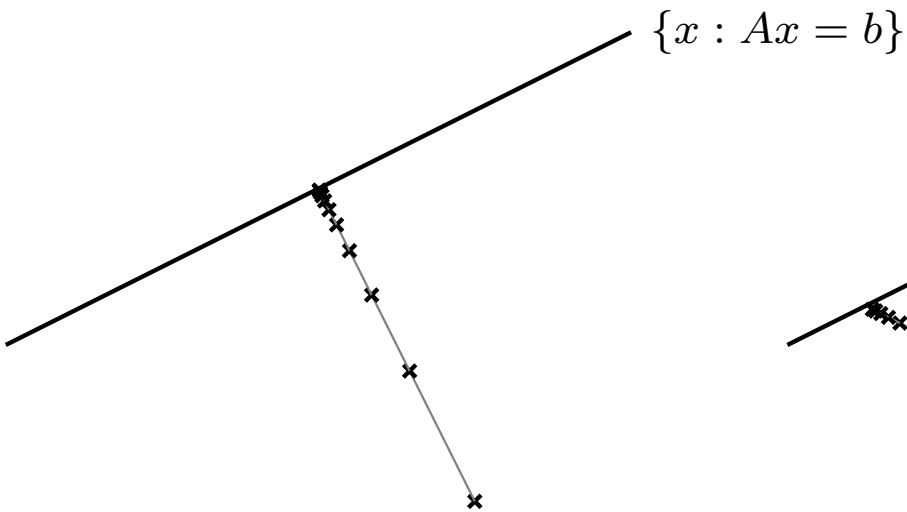
and converges to point in $X = \{x : Ax = b\}$, but

$$\{x : Ax = b\} \cap \{x : H(x - x_0) - A^T\lambda = 0\} = \Pi_{Ax=b}^H(x_0)$$

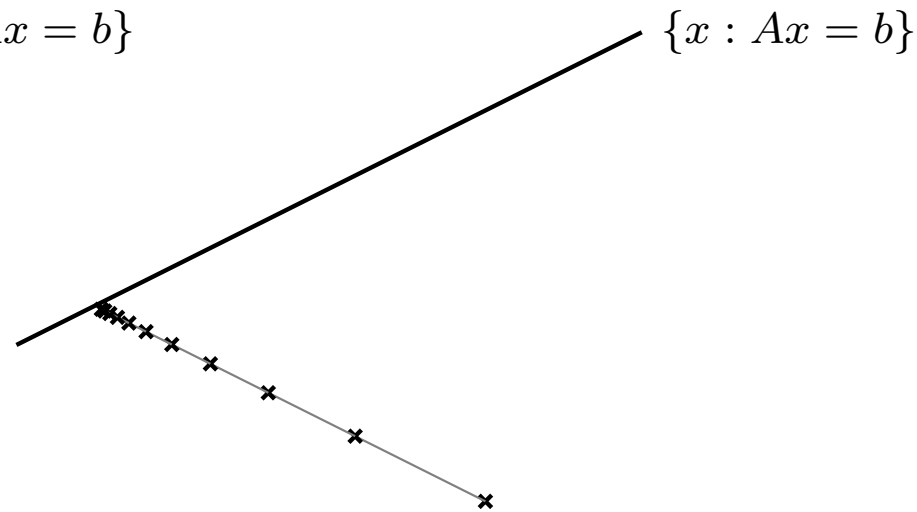
since intersection is optimatity condition for unique projection point

Graphical interpretation

- Iterates move closer (in $\|\cdot\|_H$) to all points in $X = \{x : Ax = b\}$
- X extends infinitely, therefore
  - sequence must be perpendicular to X (in scalar product $(Hx)^T y$)
  - algorithm converges to projection point $\Pi_X^H(x_0)$



Gradient method with $\|\cdot\|_2$



Gradient method with scaled $\|\cdot\|_H$

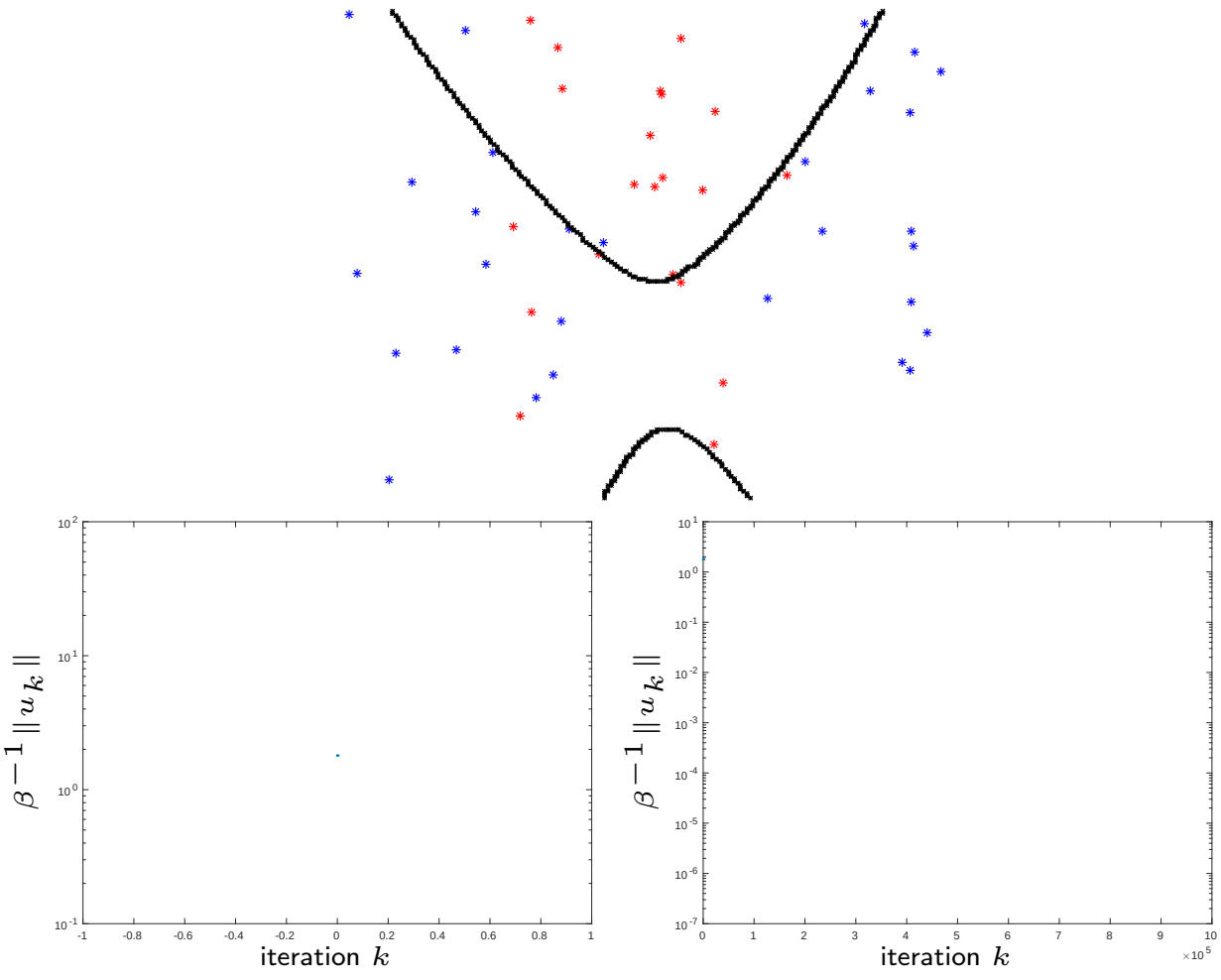
Implicit regularization

- Often good regularization when $\|\cdot\|_2$ is small
- For underdetermined convex least squares:
  - GD with $\|\cdot\|_H$ metric converges to minimum $\|\cdot\|_H$-norm solution
  - Same conclusion in expectation for SGD
- Implicit regularization for least squares:
  - Gradient method with $\|\cdot\|_2$ has implicit desired regularization
  - Adagrad-type methods have different (not desired?) implicit regularization
- Emperical evidence that same thing holds in nonconvex settings
- So, why not explicitly regularize?
  - Convex settings: of course!
  - Nonconvex: may change loss landscape in not desired ways

Early termination

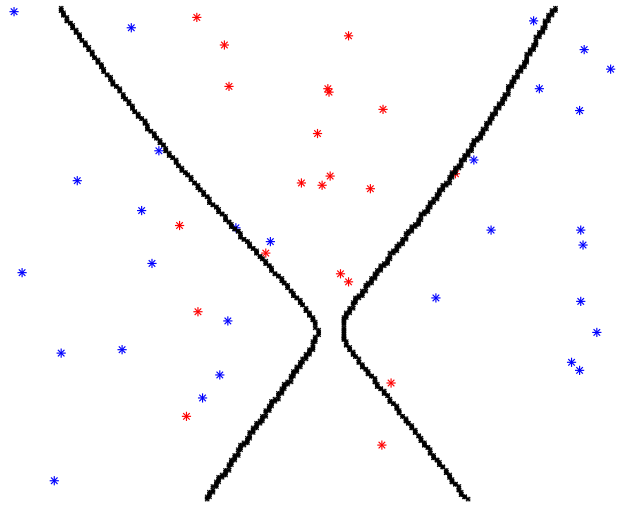
Early termination

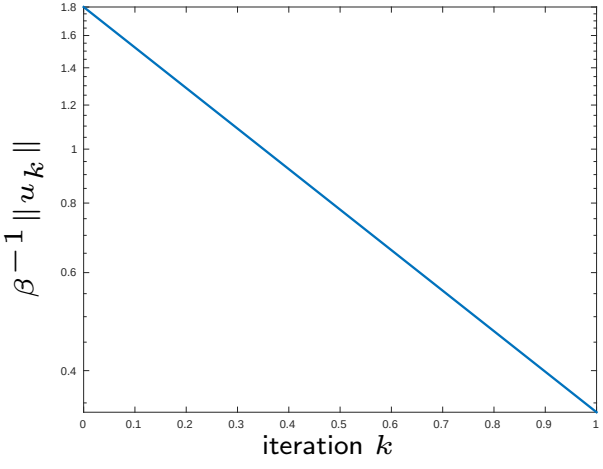
- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 1 Residual norm: $\beta^{-1}\|u_k\|_2 = 6.6e^{-1}$

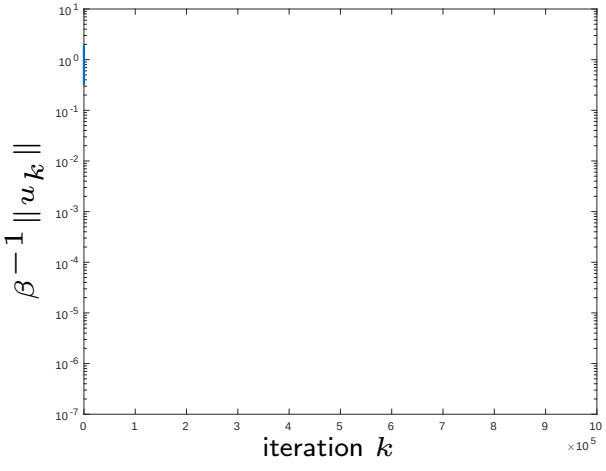


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 2 Residual norm: $\beta^{-1}\|u_k\|_2 = 4.7e^{-1}$

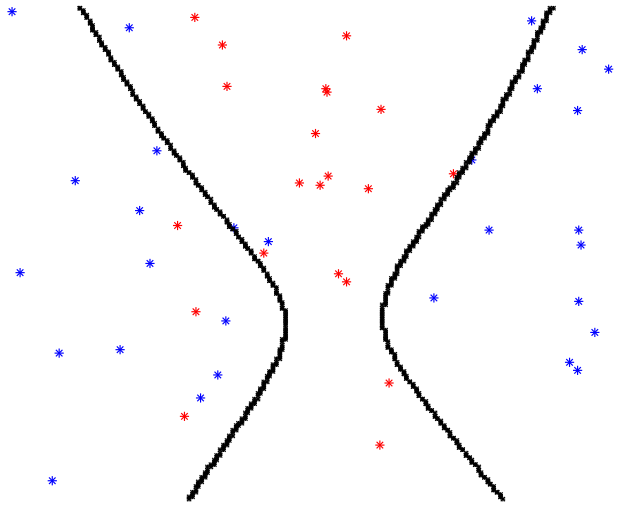


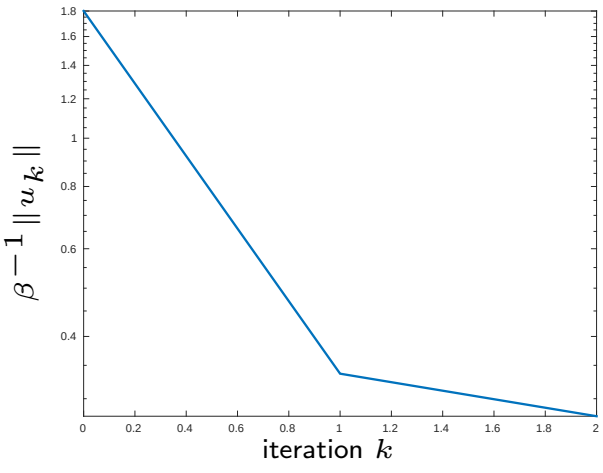


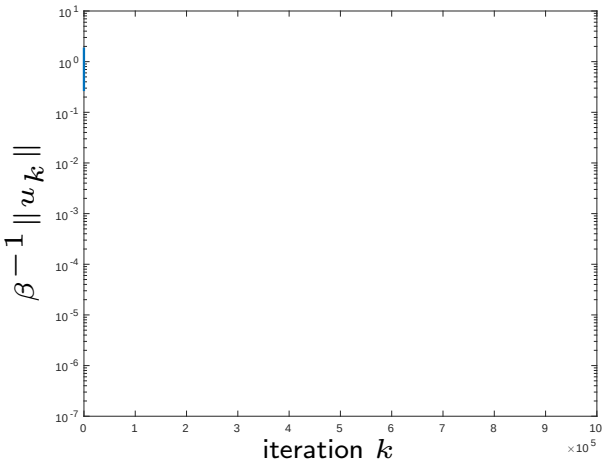


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 3 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.5e^{-1}$

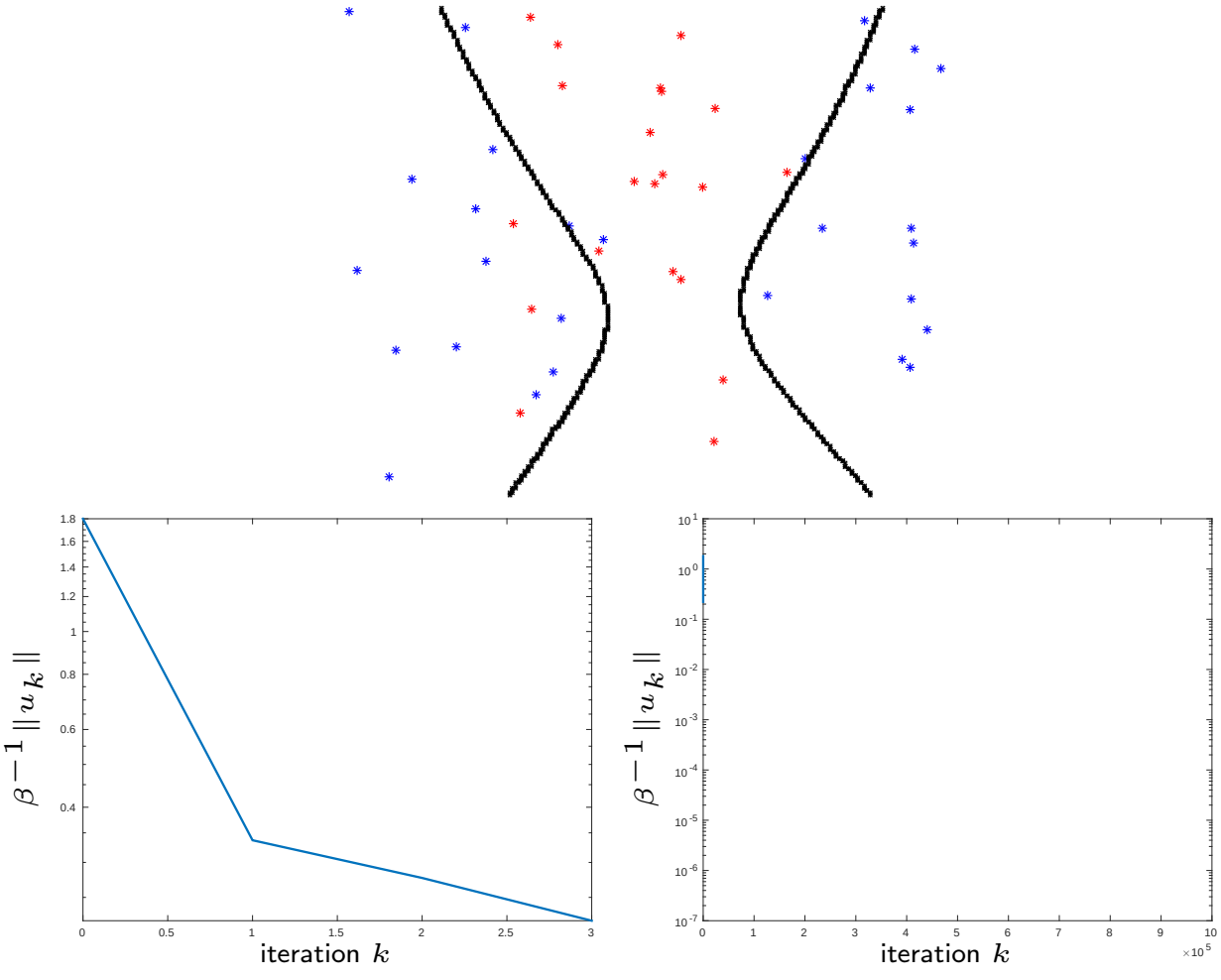






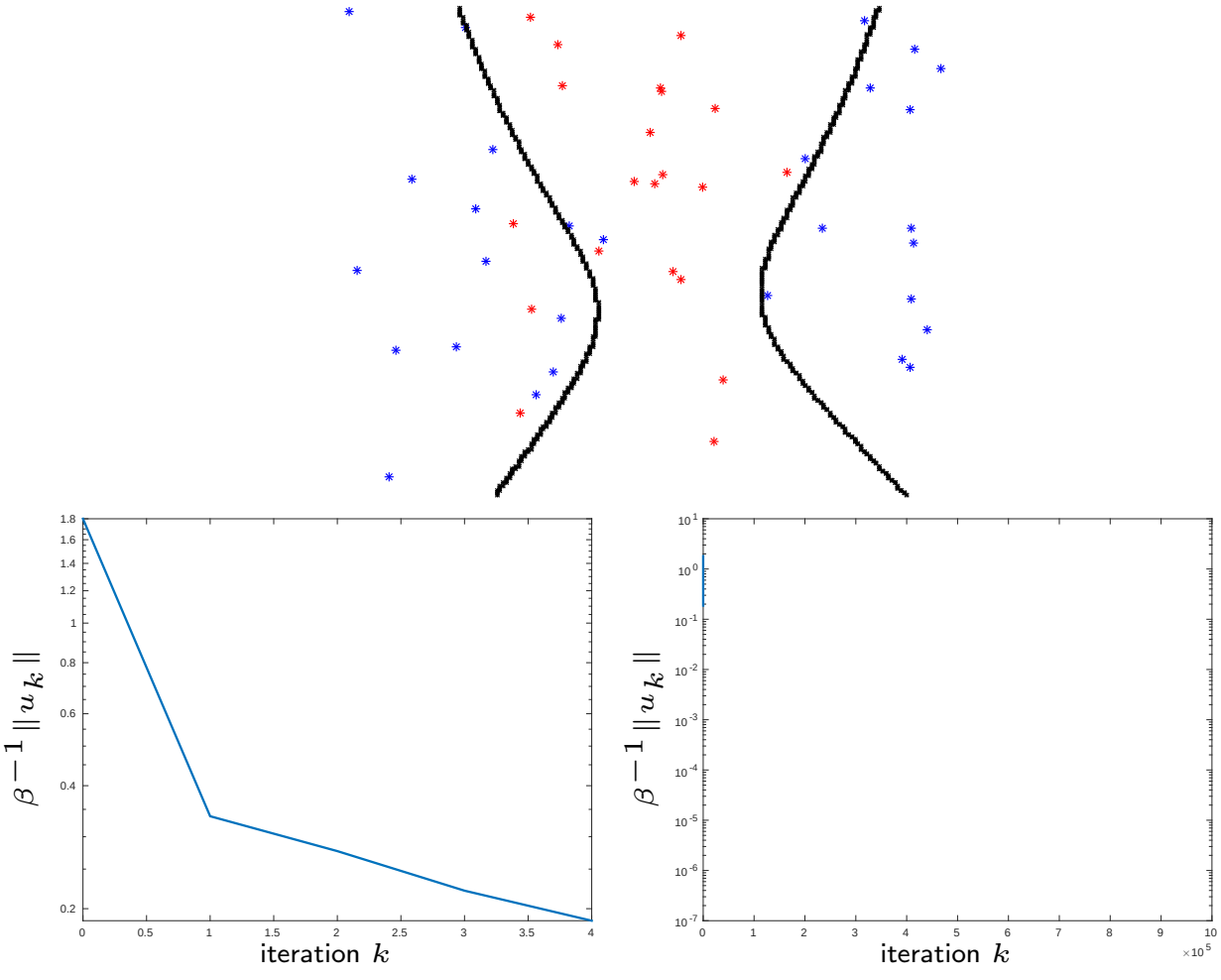
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 4 Residual norm: $\beta^{-1}\|u_k\|_2 = 2.8e^{-1}$



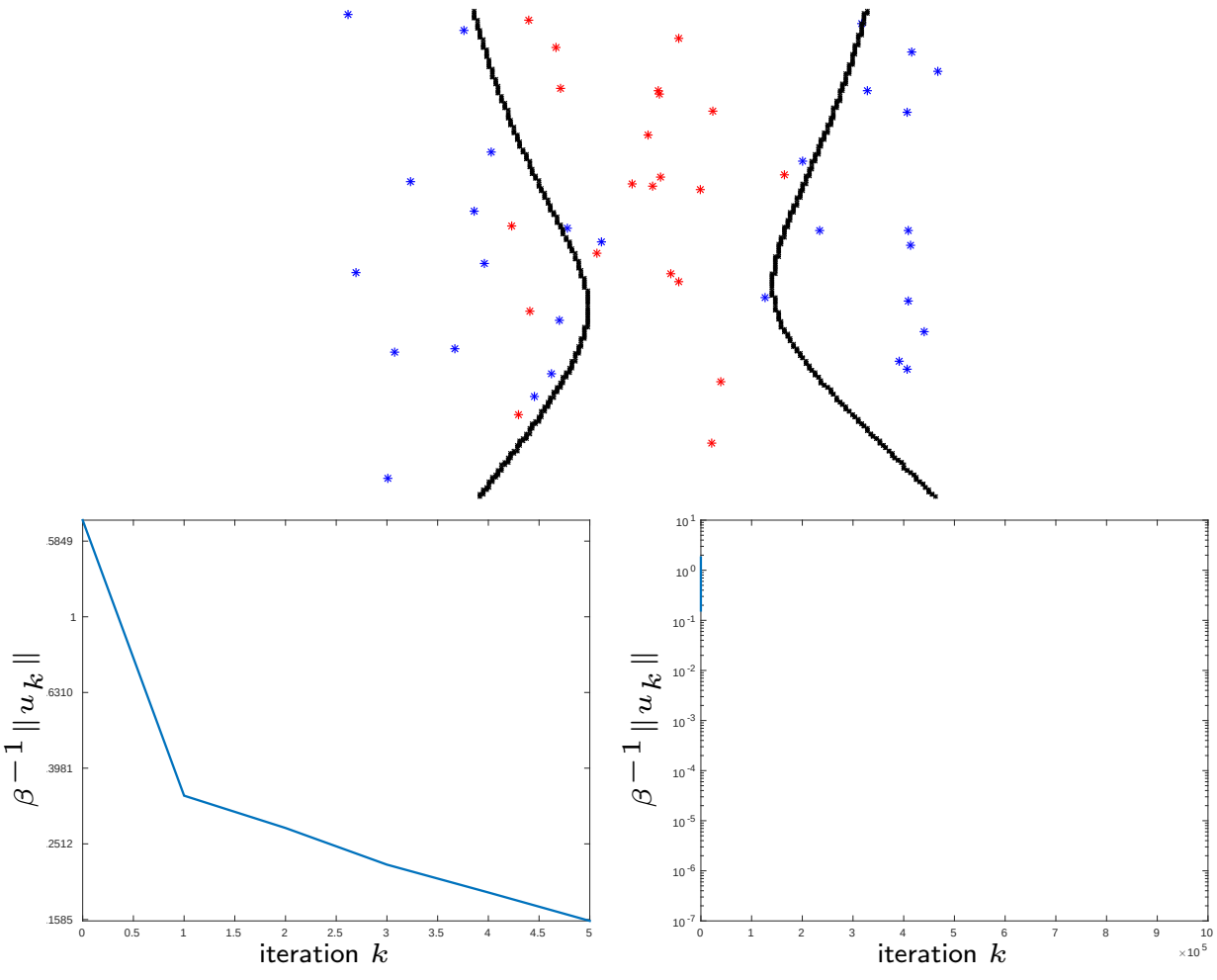
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 5 Residual norm: $\beta^{-1}\|u_k\|_2 = 2.3e^{-1}$



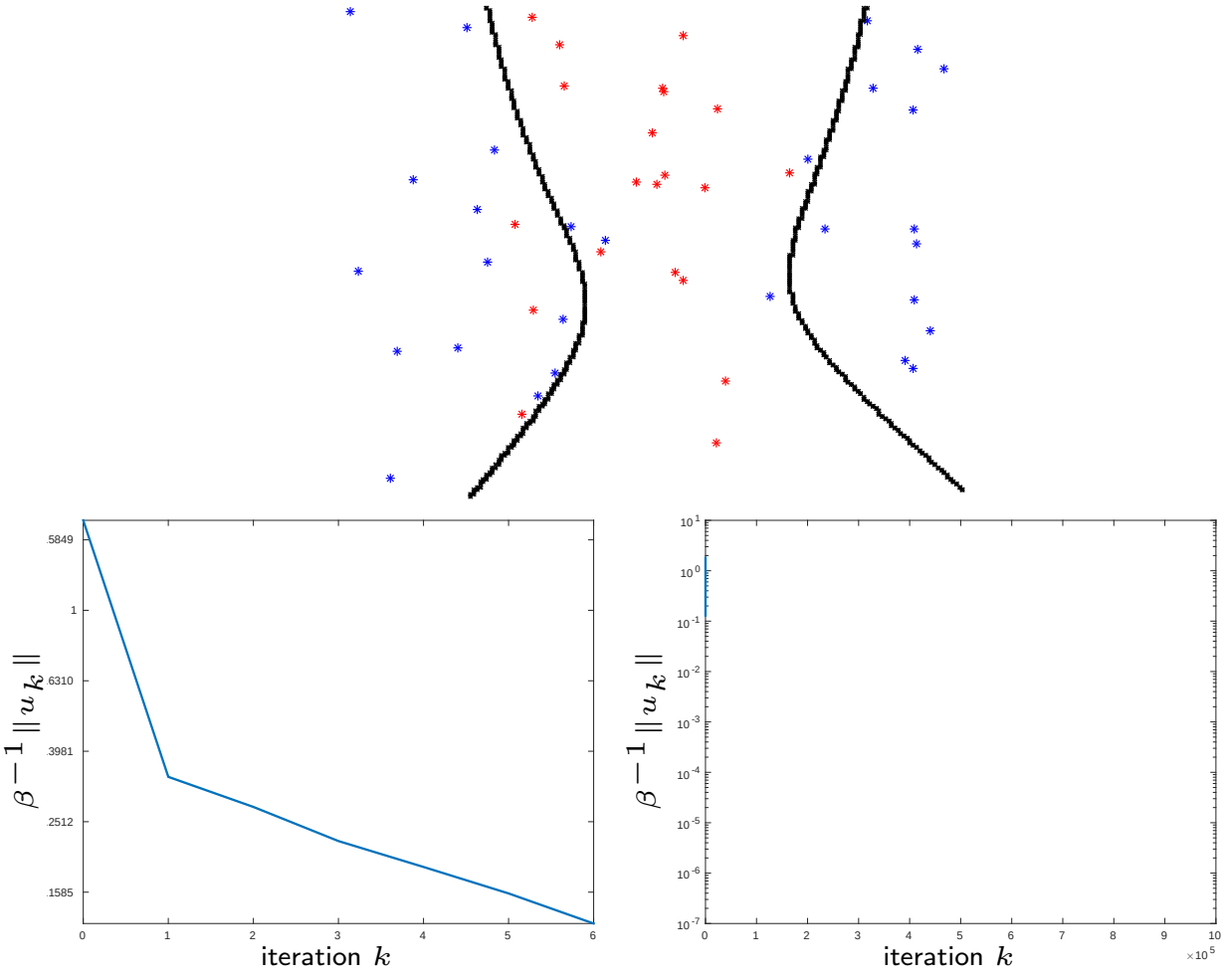
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 6 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.9e^{-1}$



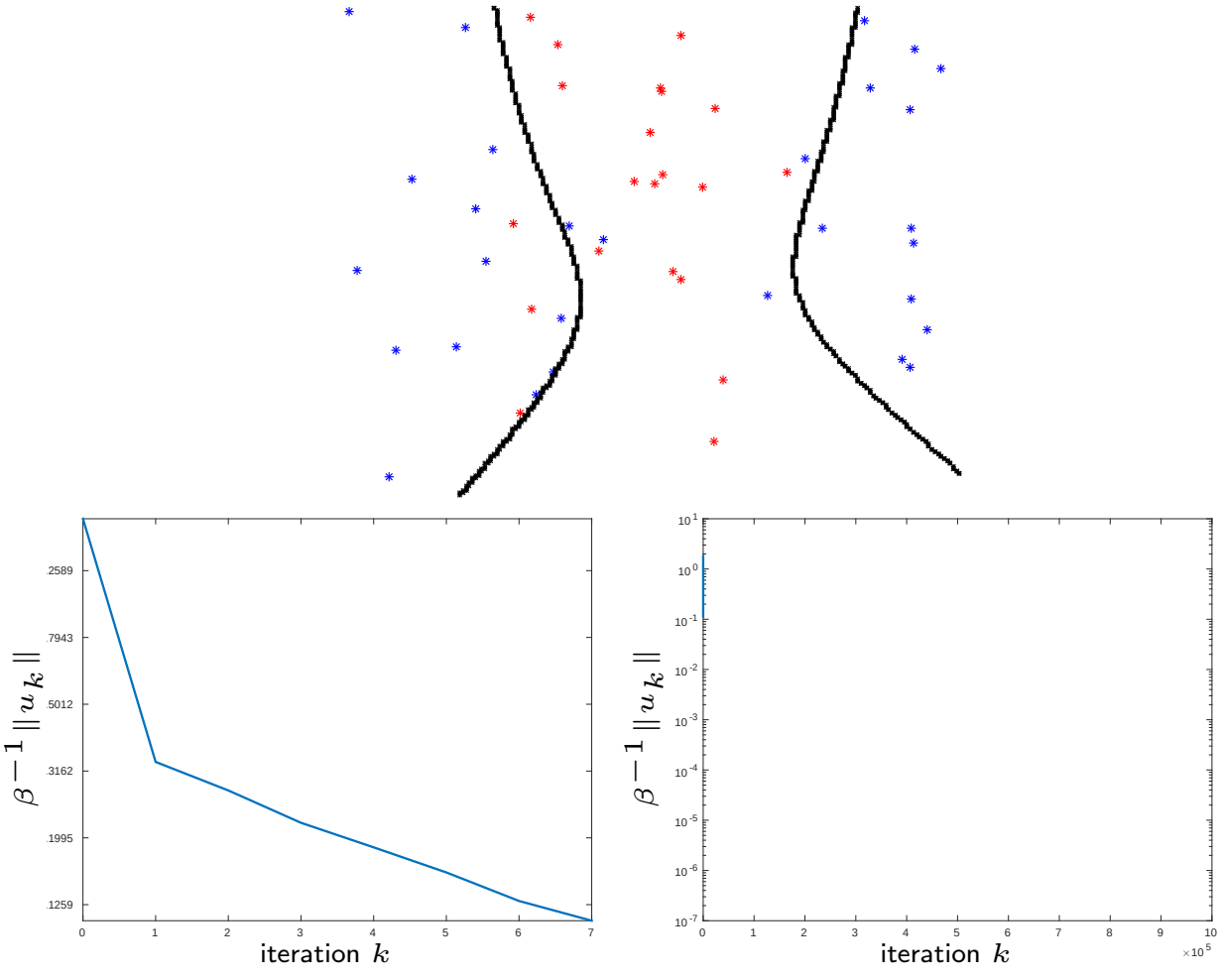
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 7 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.5e^{-1}$



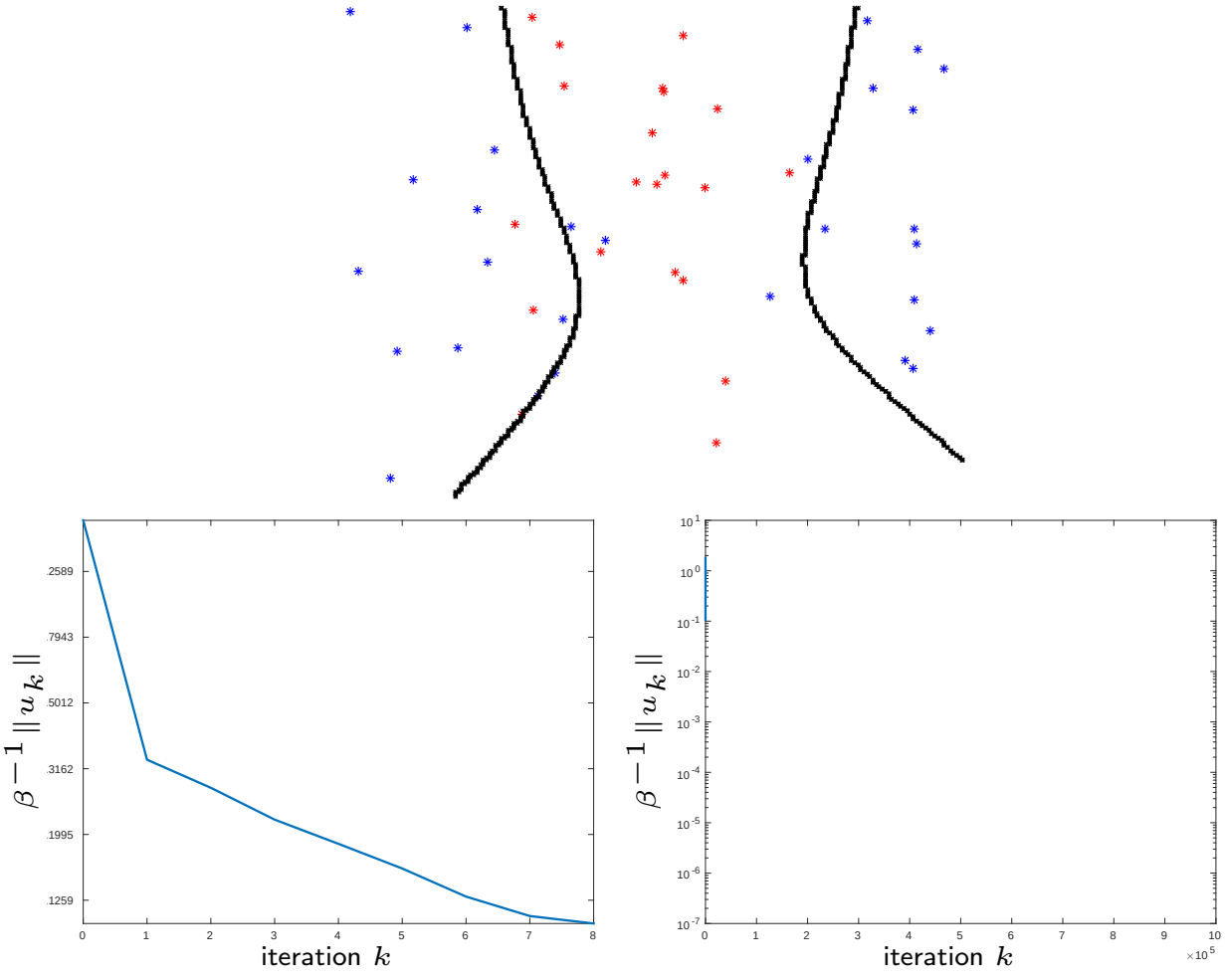
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 8 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.3e^{-1}$



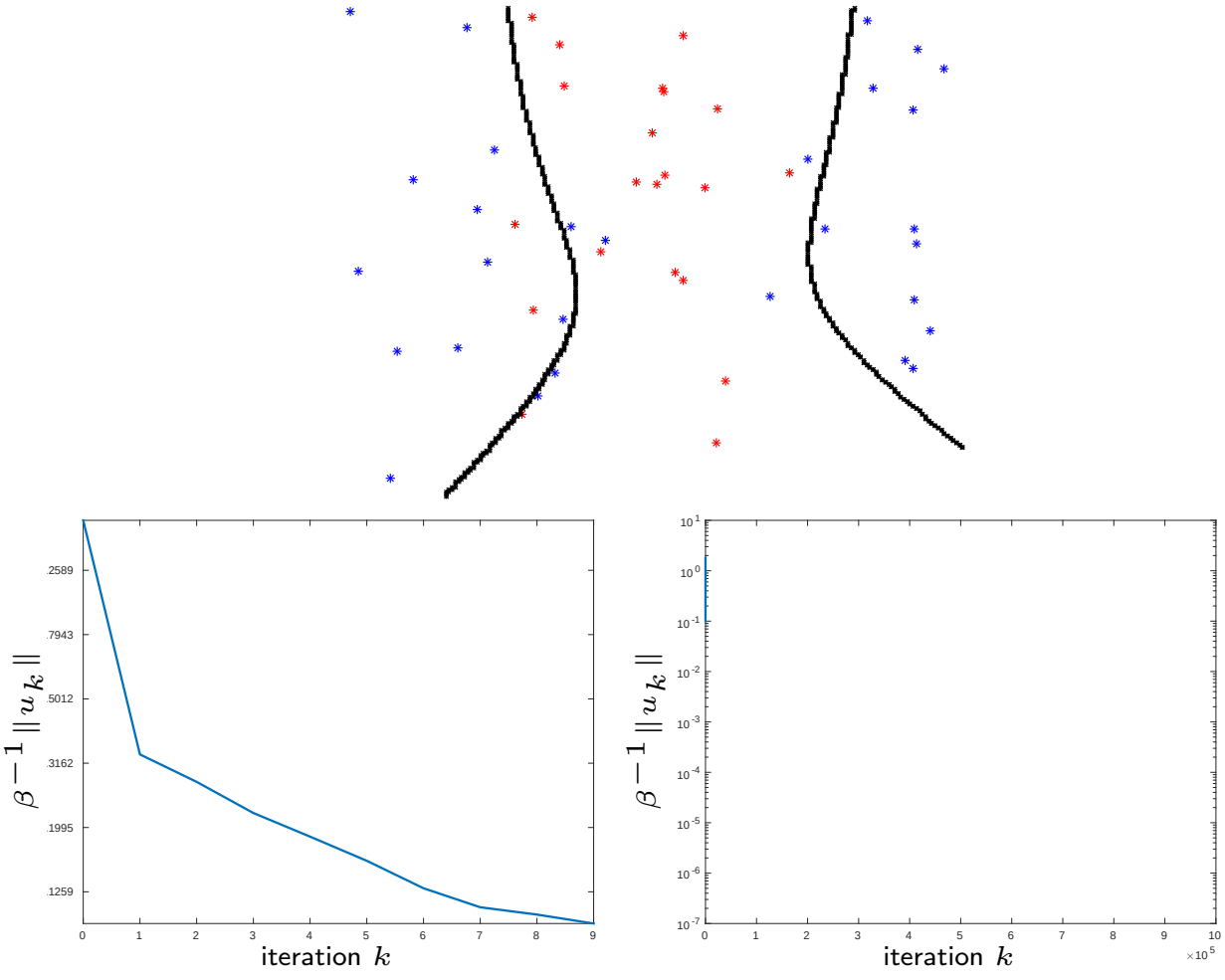
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 9 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.2e^{-1}$



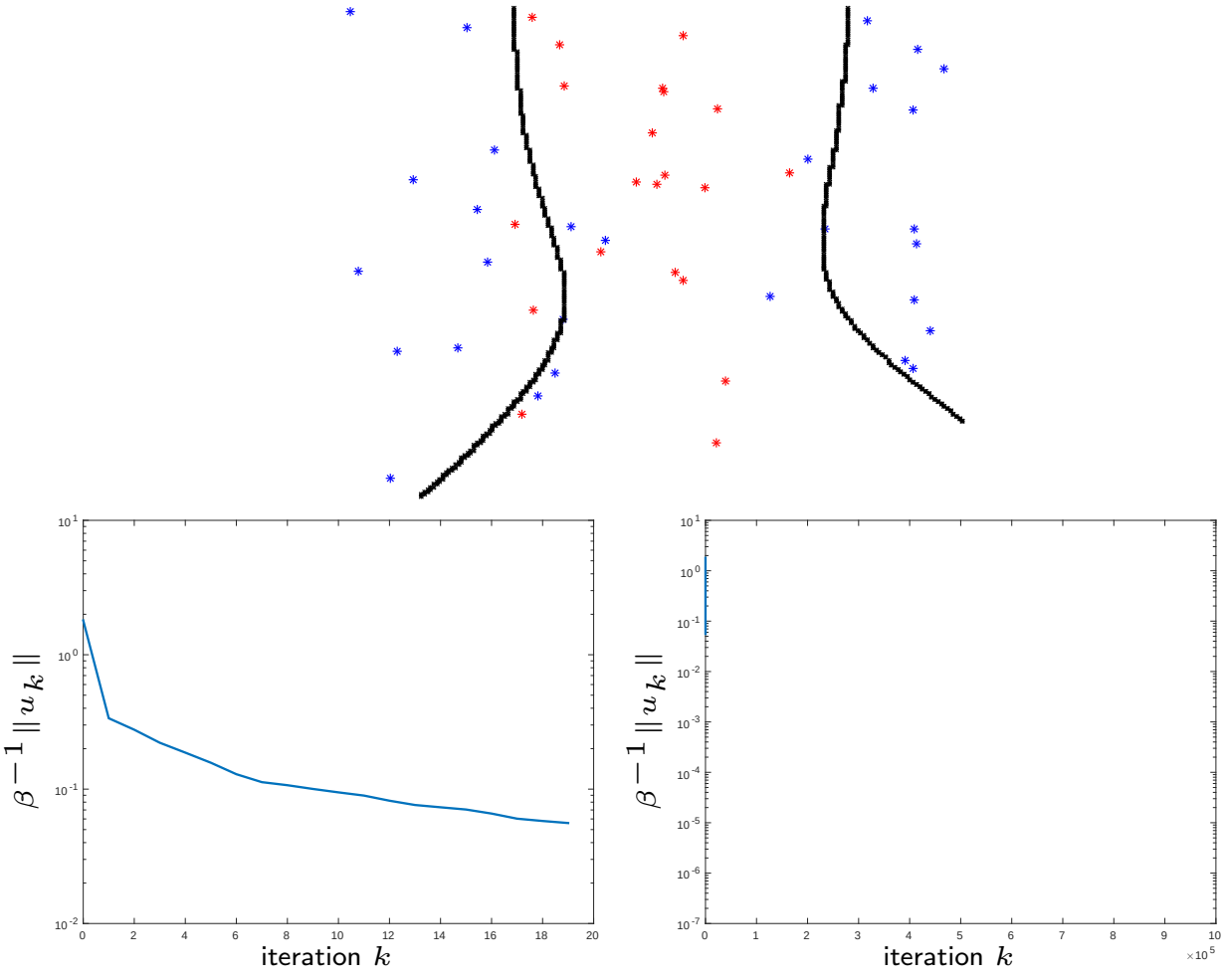
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 10 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.1e^{-1}$



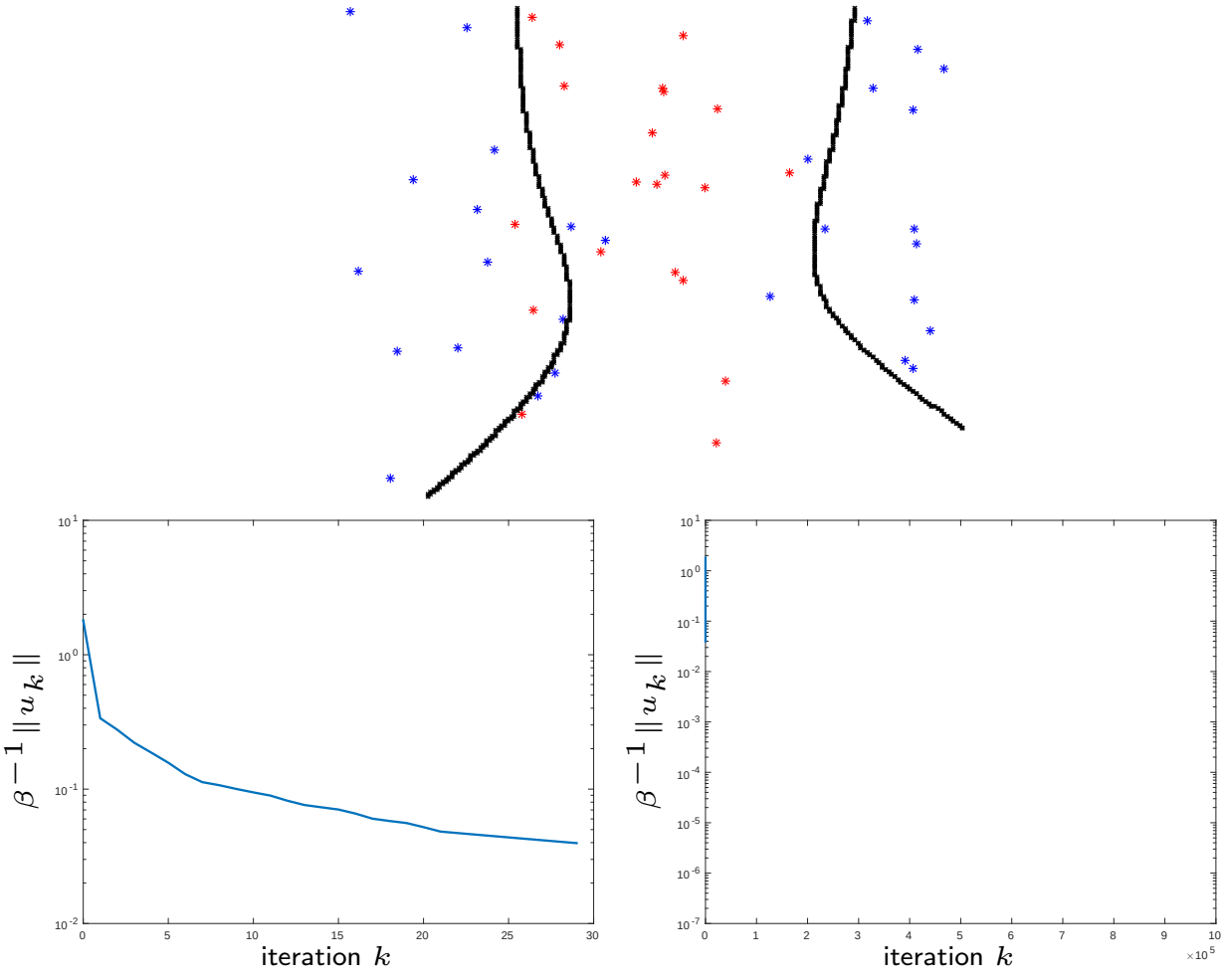
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 20 Residual norm: $\beta^{-1}\|u_k\|_2 = 5.8e^{-2}$



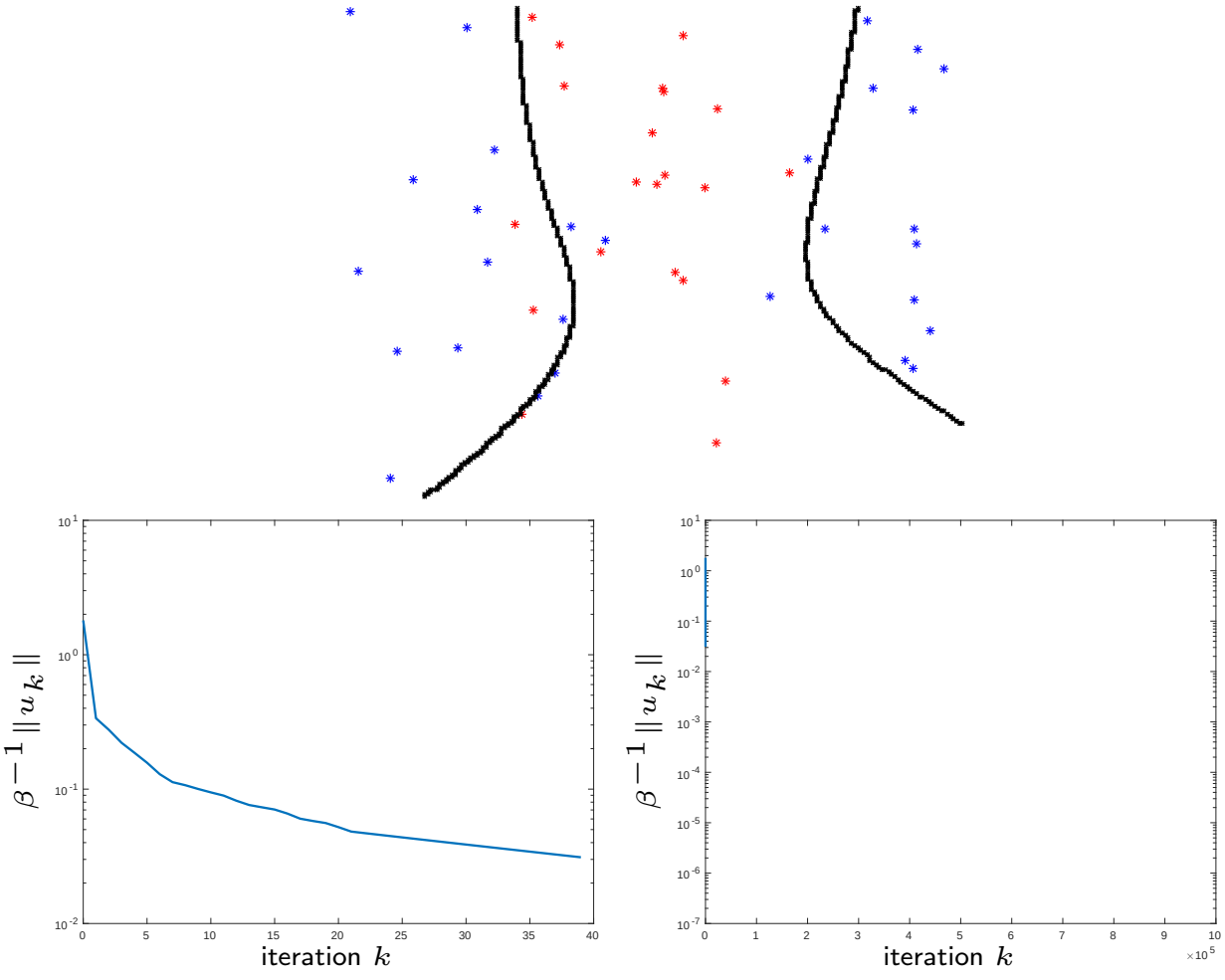
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 30 Residual norm: $\beta^{-1}\|u_k\|_2 = 4.1e^{-2}$



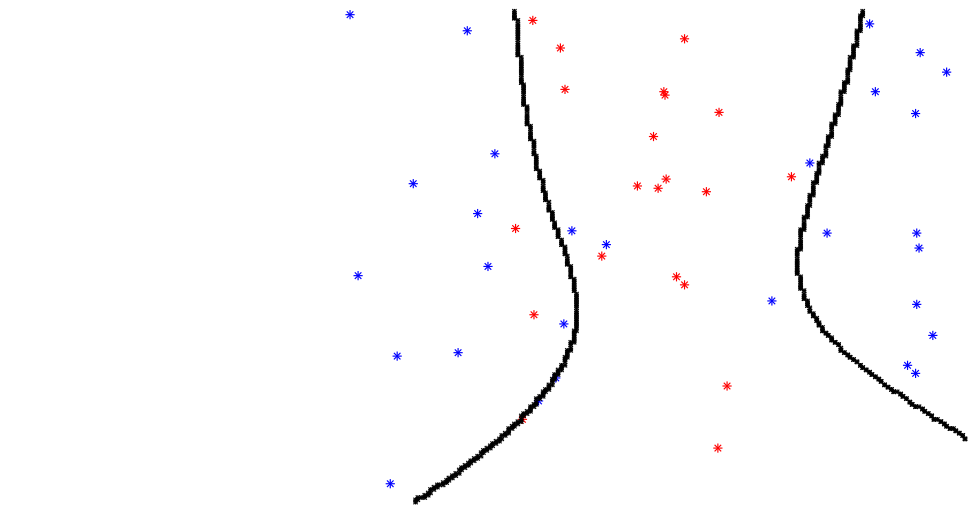
Early termination

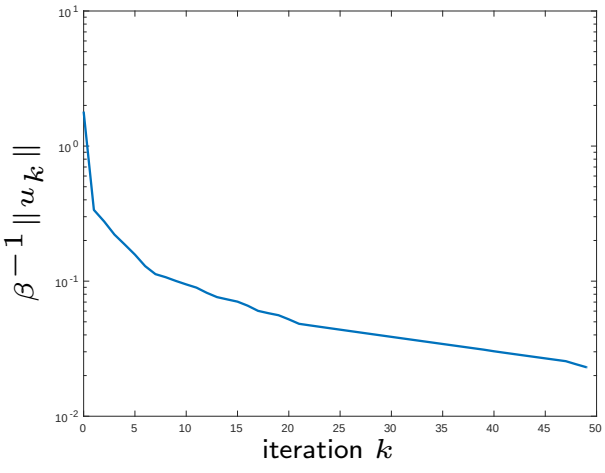
- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 40 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.2e^{-2}$

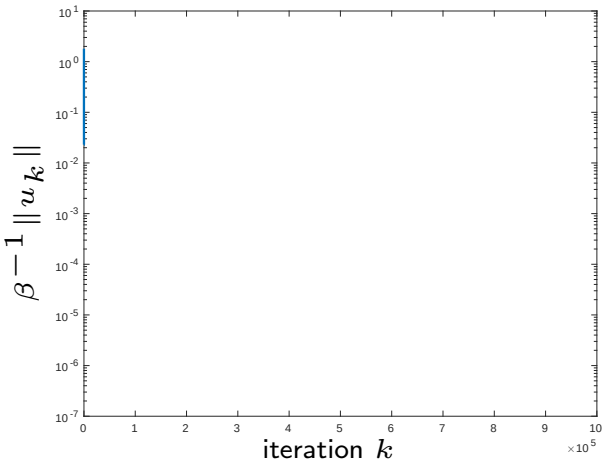


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 50 Residual norm: $\beta^{-1}\|u_k\|_2 = 2.4e^{-2}$

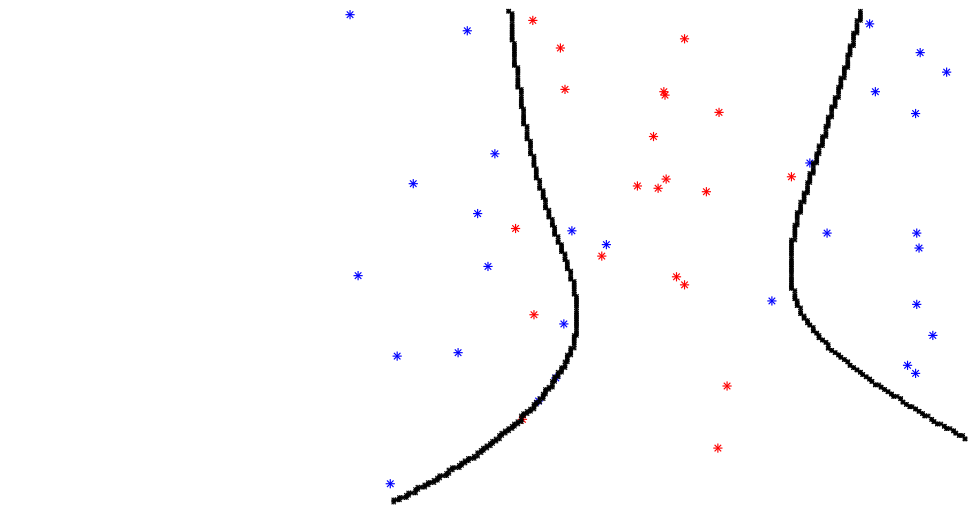


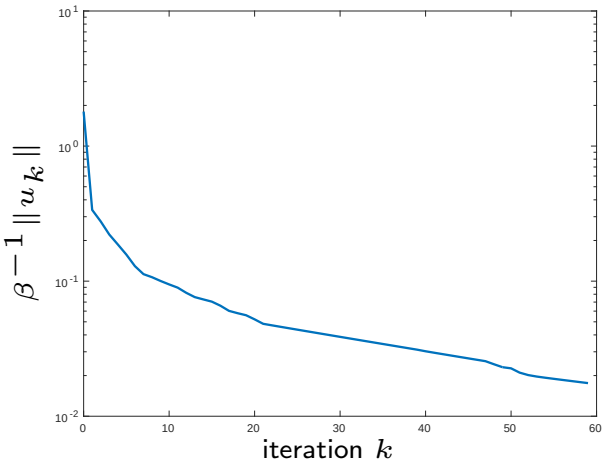


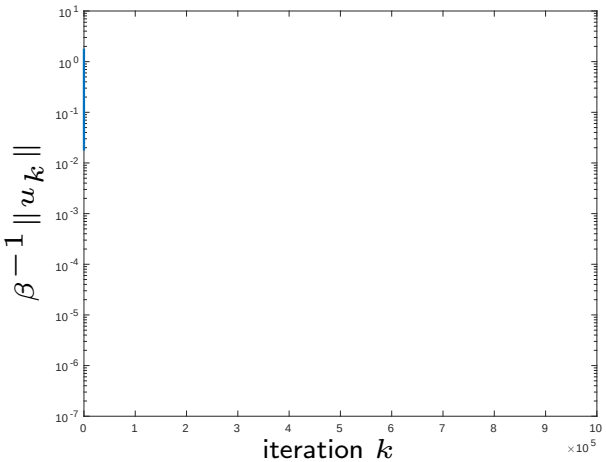


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 60 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.8e^{-2}$

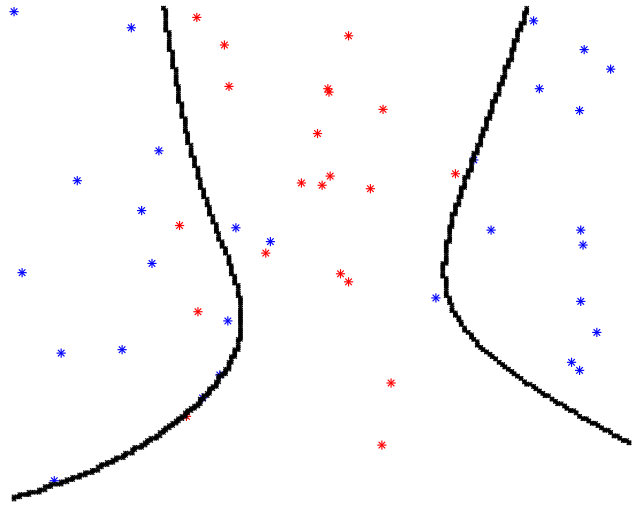


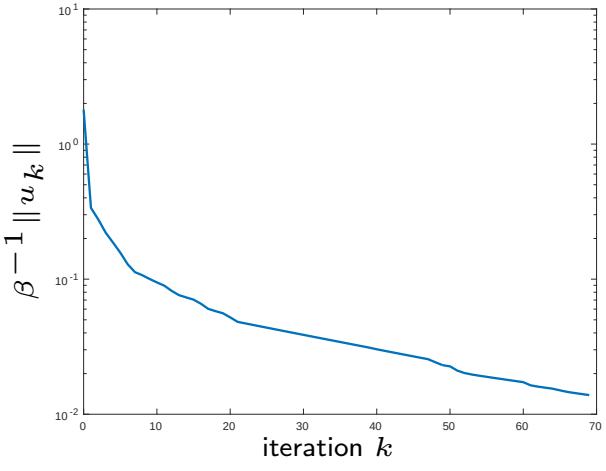


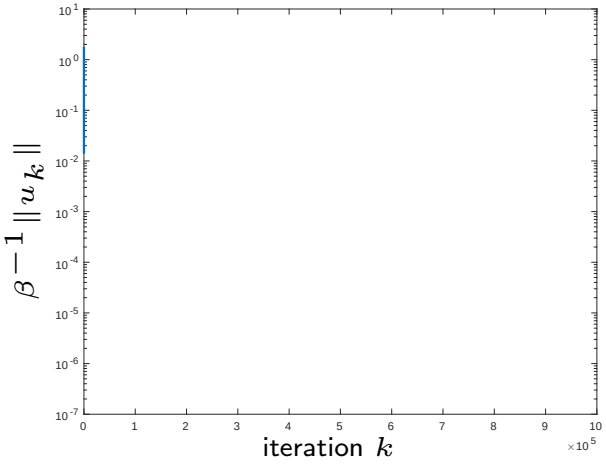


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 70 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.4e^{-2}$

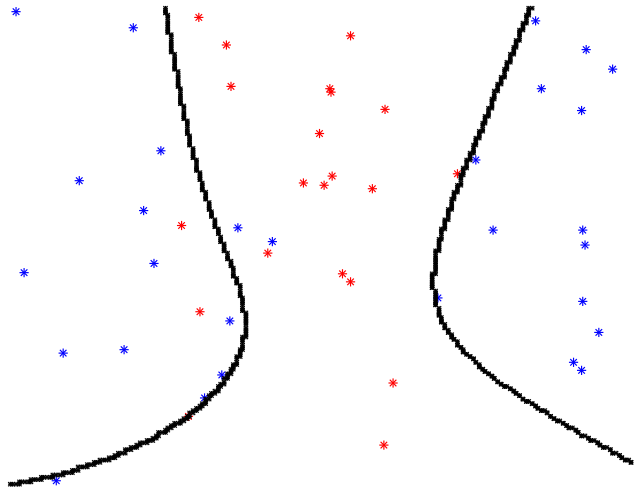


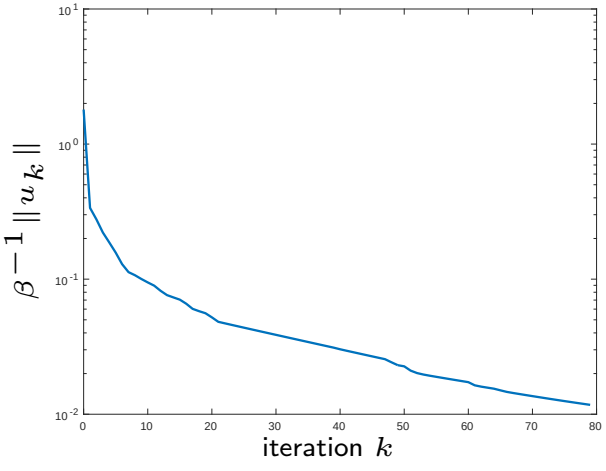


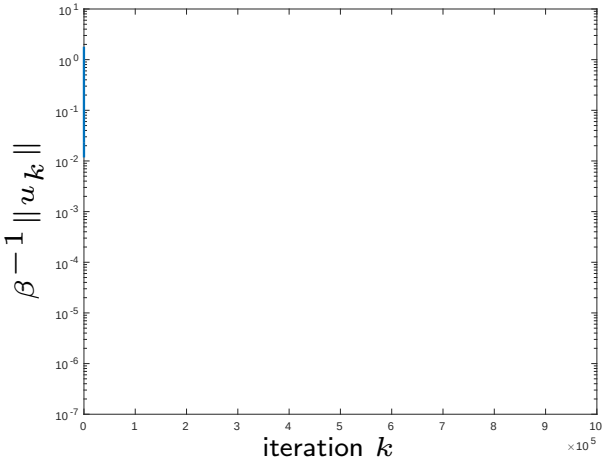


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 80 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.2e^{-2}$

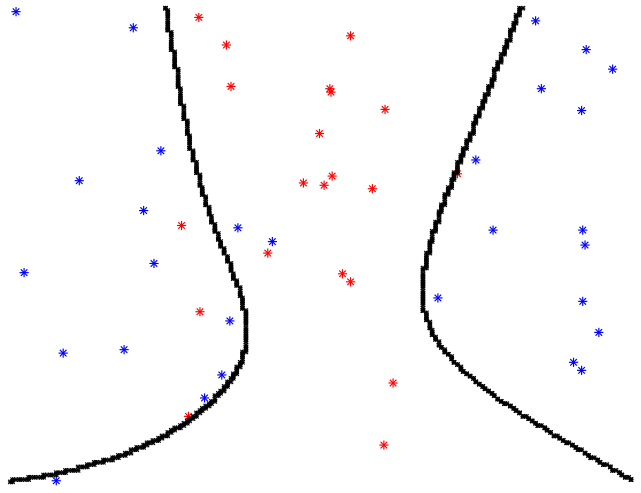


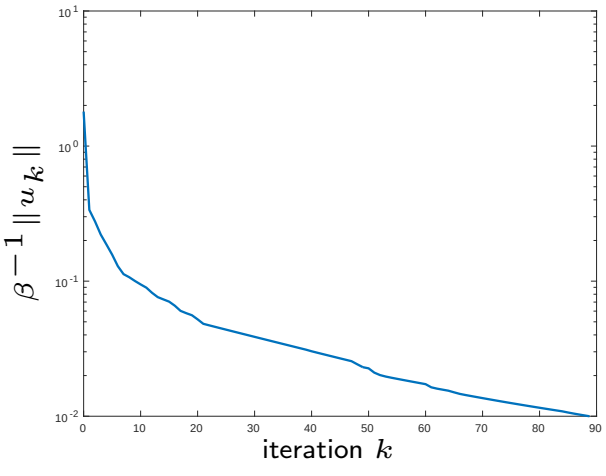


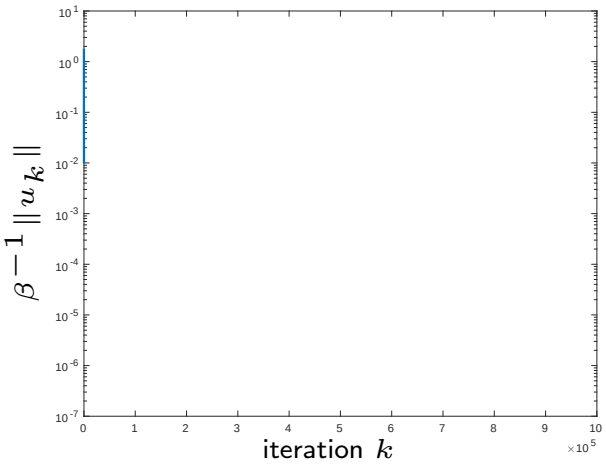


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 90 Residual norm: $\beta^{-1}\|u_k\|_2 = 1e^{-2}$

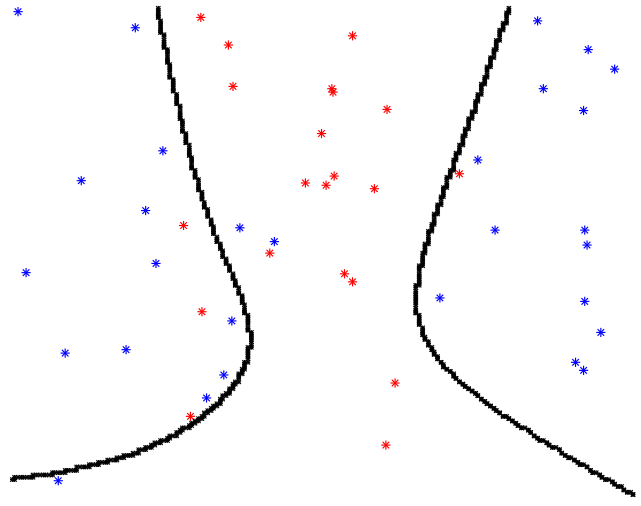


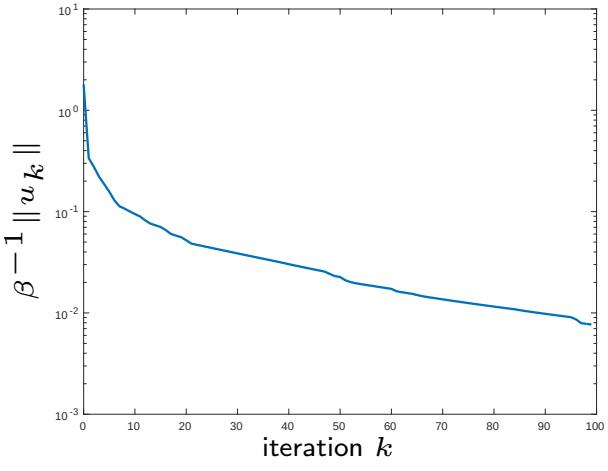


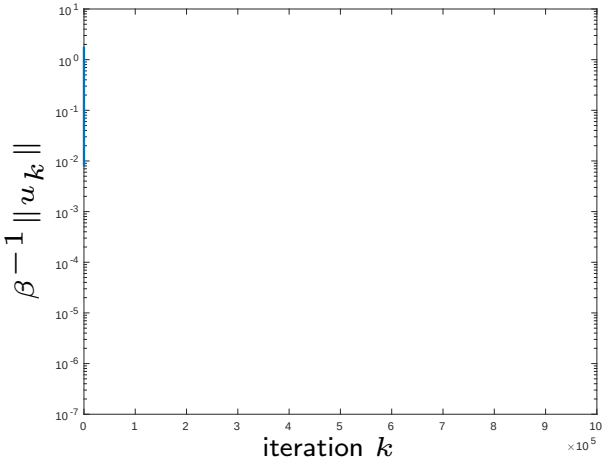


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 100 Residual norm: $\beta^{-1}\|u_k\|_2 = 7.8e^{-3}$

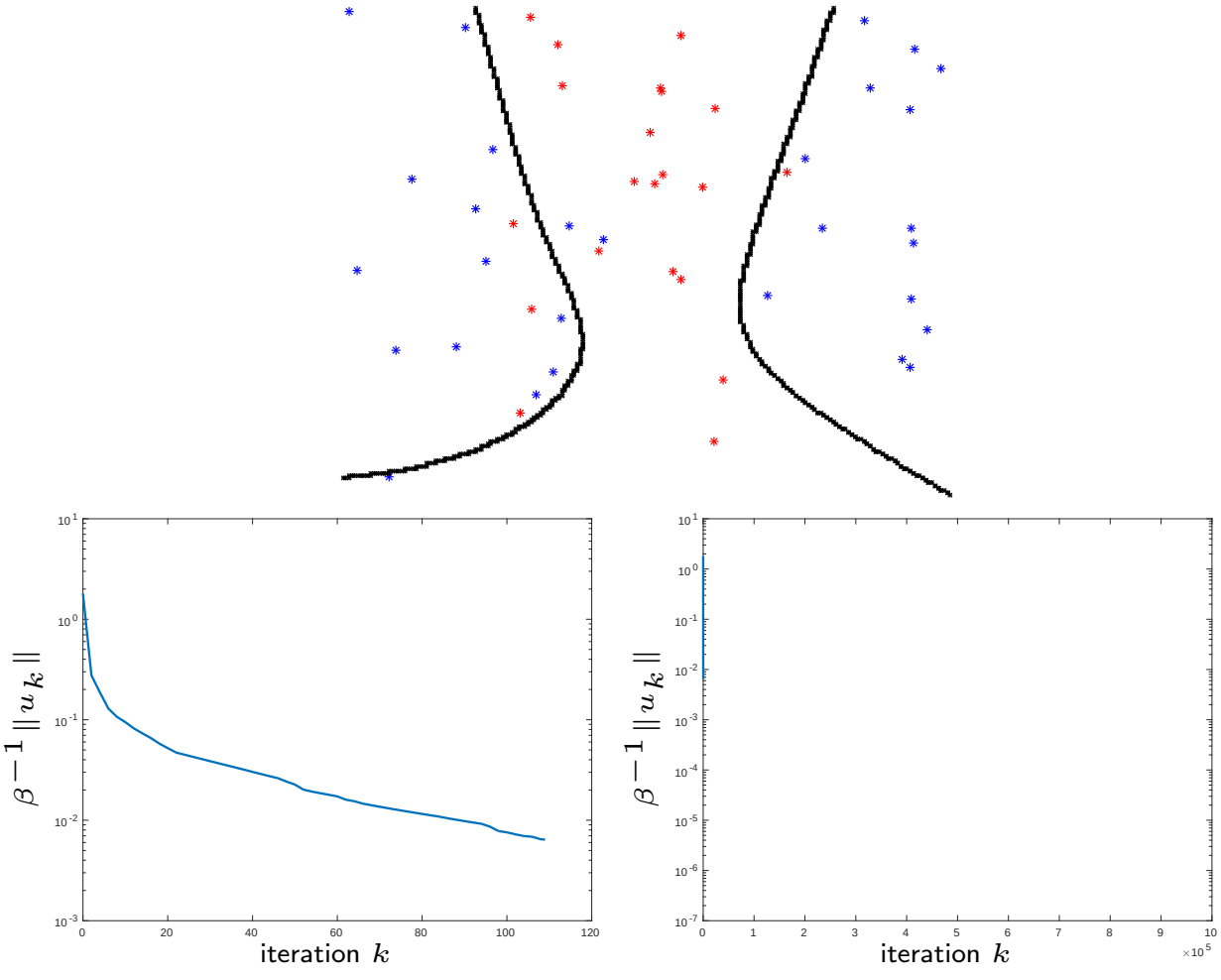






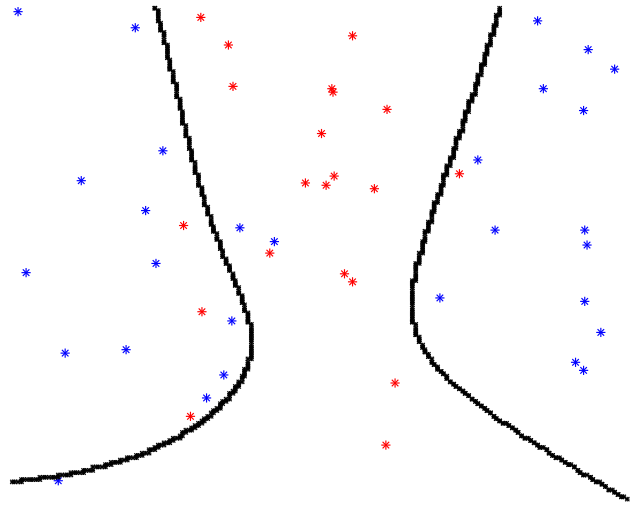
Early termination

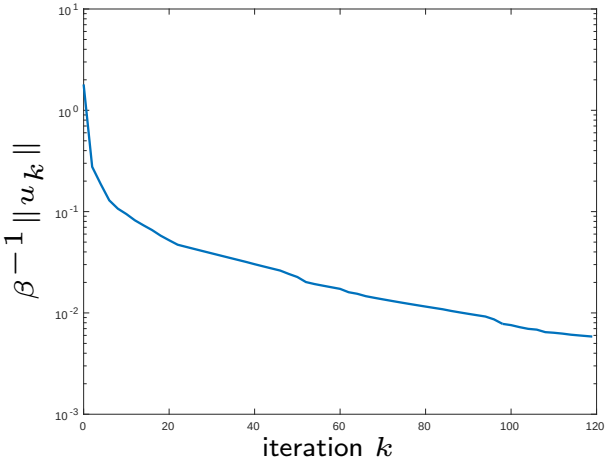
- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 110 Residual norm: $\beta^{-1}\|u_k\|_2 = 6.5e^{-3}$

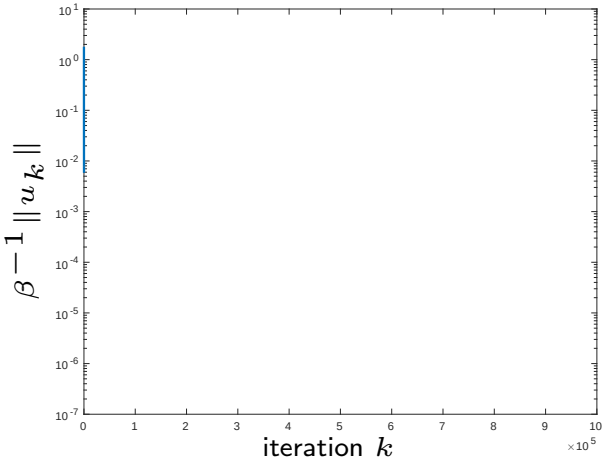


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 120 Residual norm: $\beta^{-1}\|u_k\|_2 = 5.9e^{-3}$

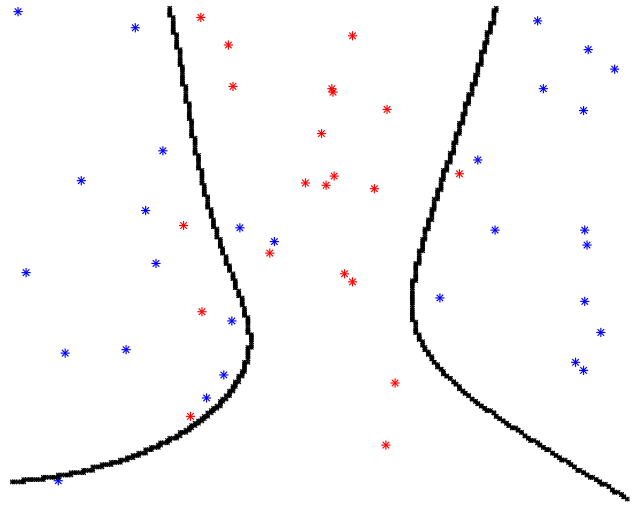


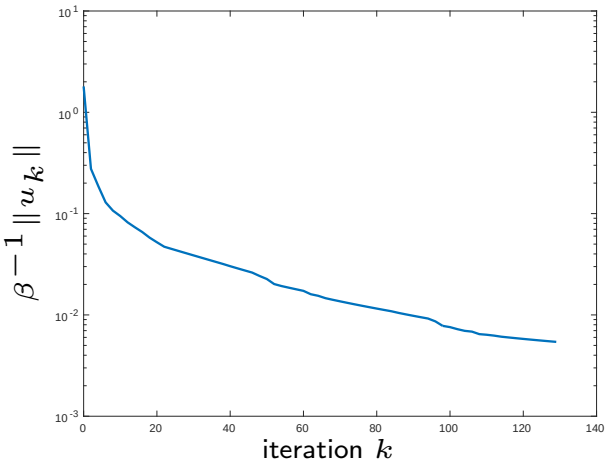


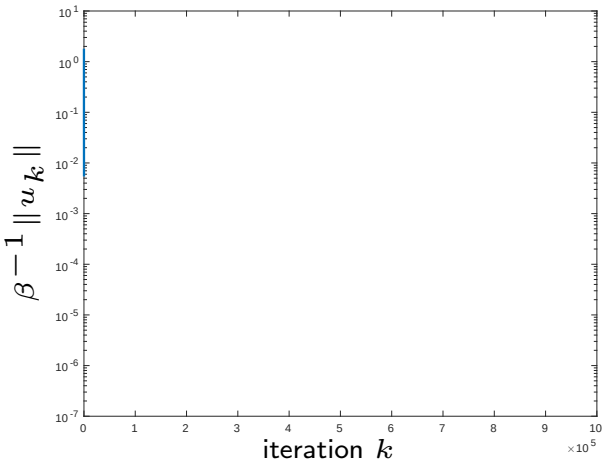


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 130 Residual norm: $\beta^{-1}\|u_k\|_2 = 5.5e^{-3}$

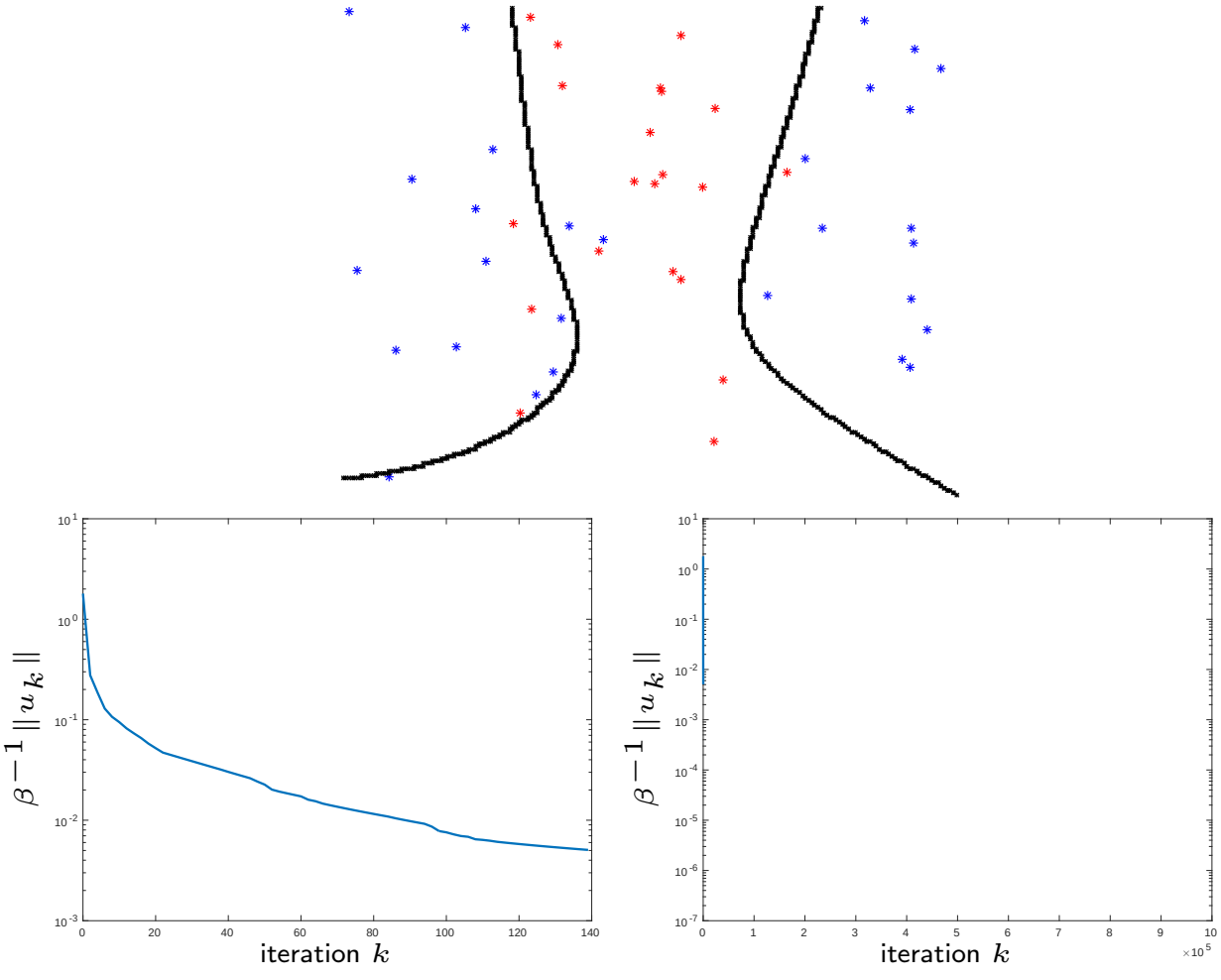






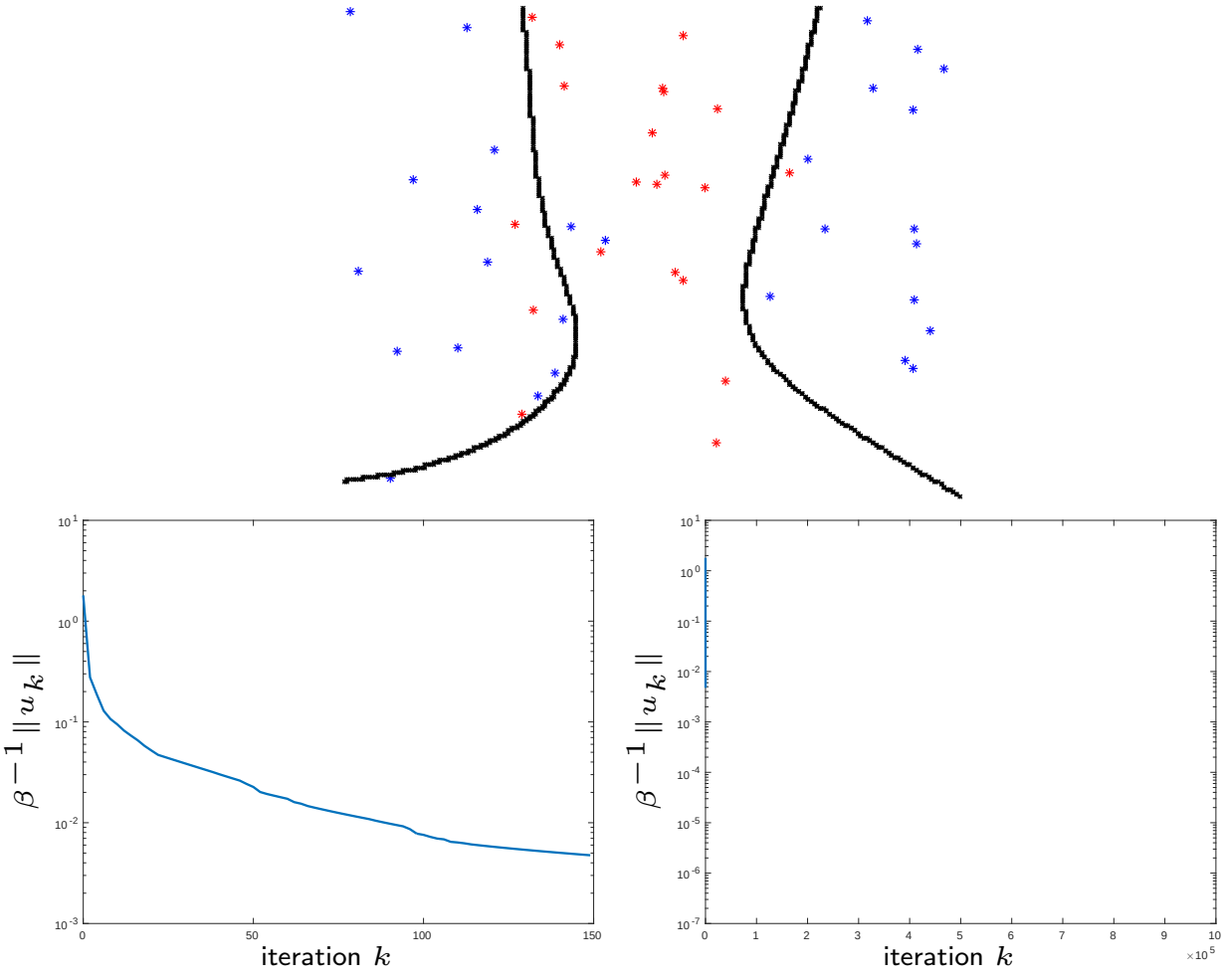
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 140 Residual norm: $\beta^{-1}\|u_k\|_2 = 5.1e^{-3}$



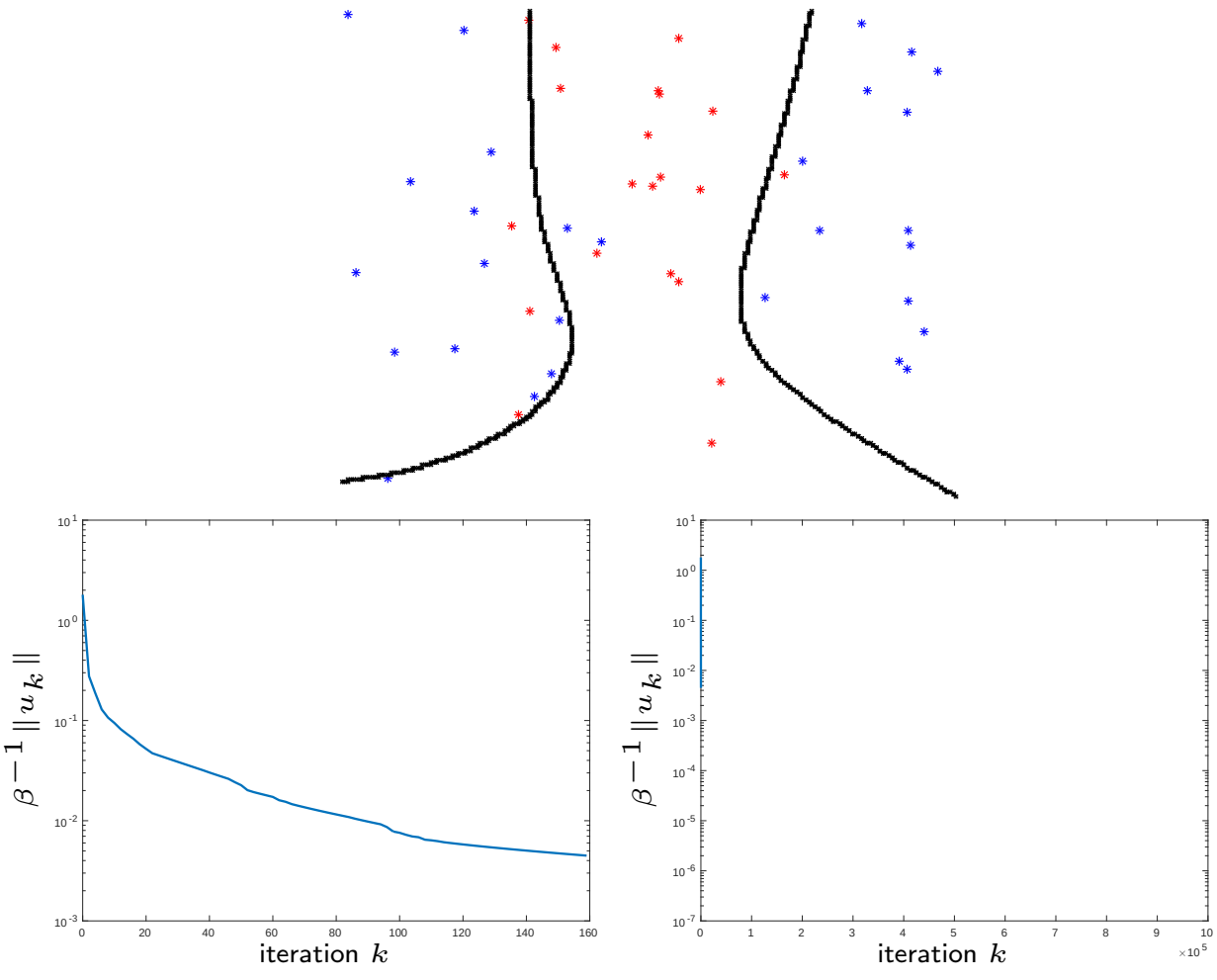
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 150 Residual norm: $\beta^{-1}\|u_k\|_2 = 4.8e^{-3}$



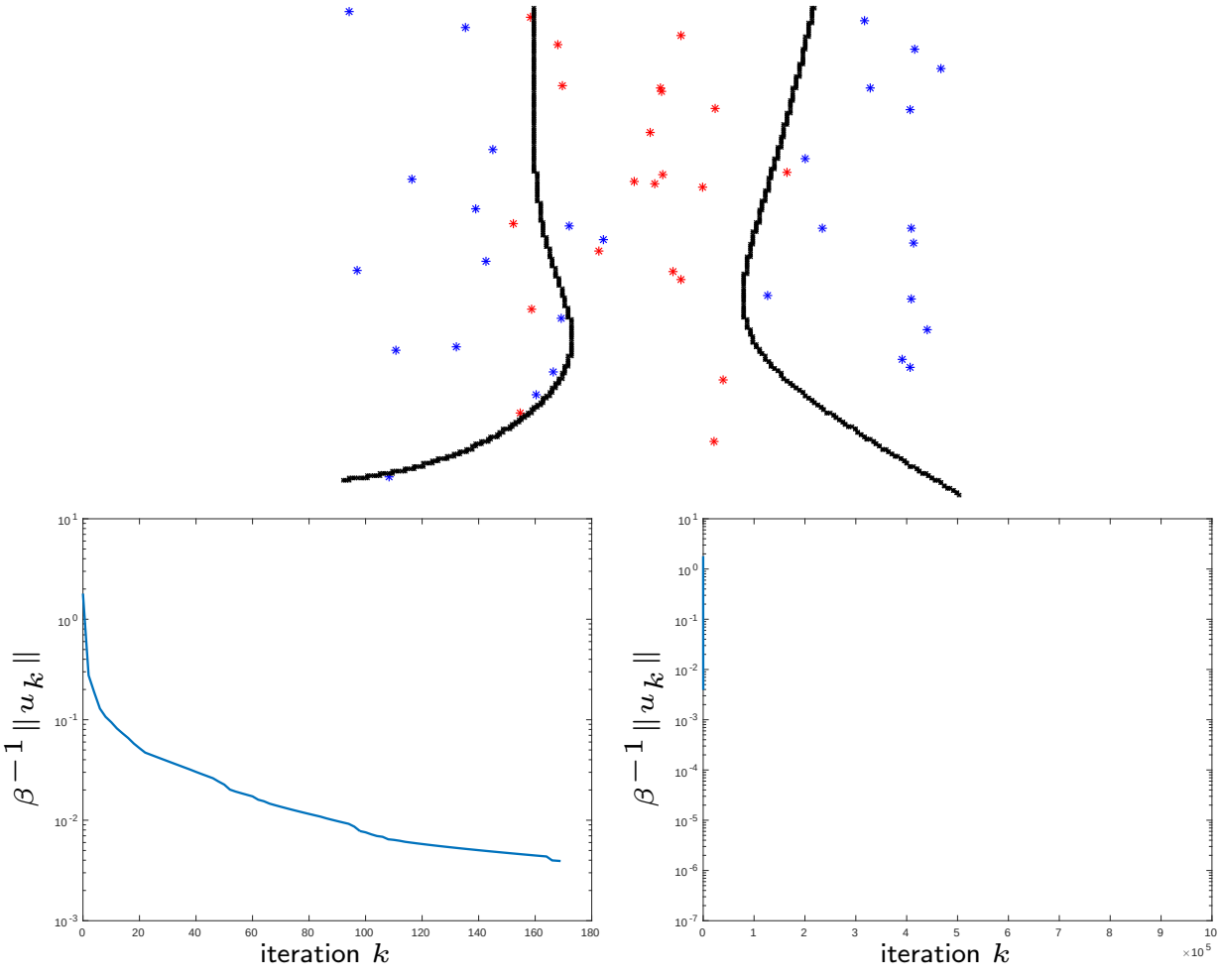
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 160 Residual norm: $\beta^{-1}\|u_k\|_2 = 4.5e^{-3}$



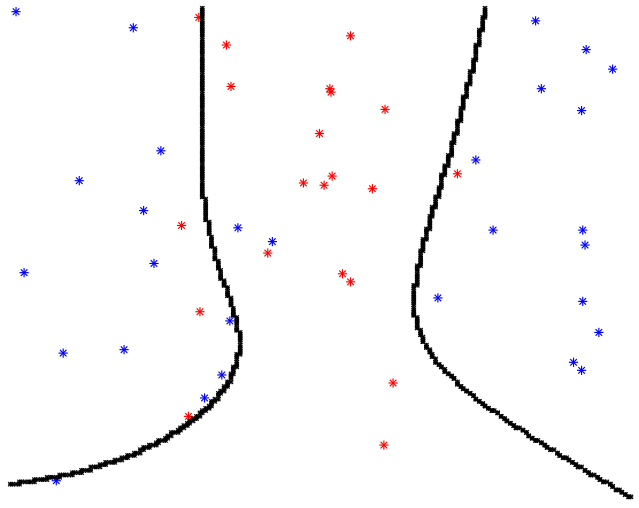
Early termination

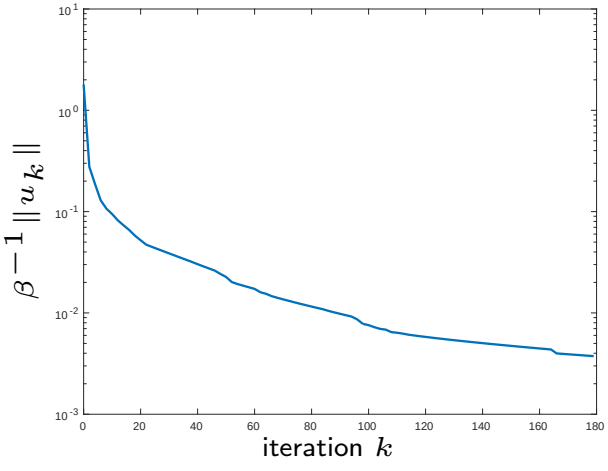
- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 170 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.9e^{-3}$

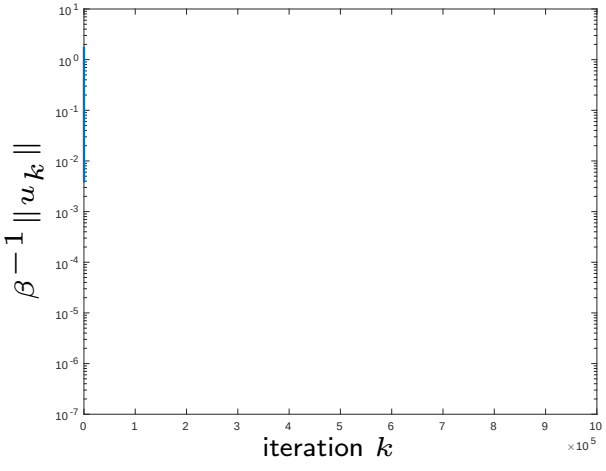


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 180 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.8e^{-3}$

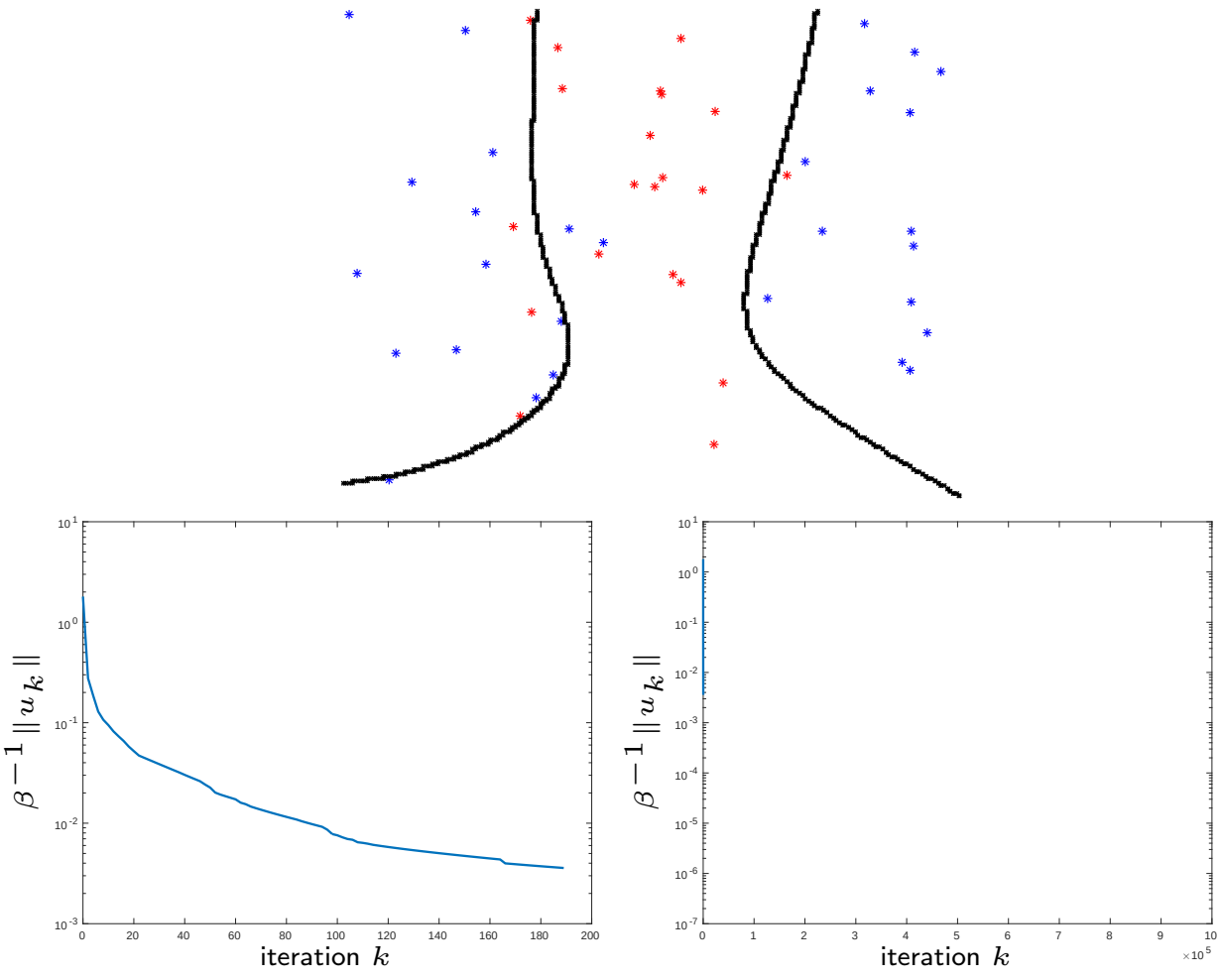






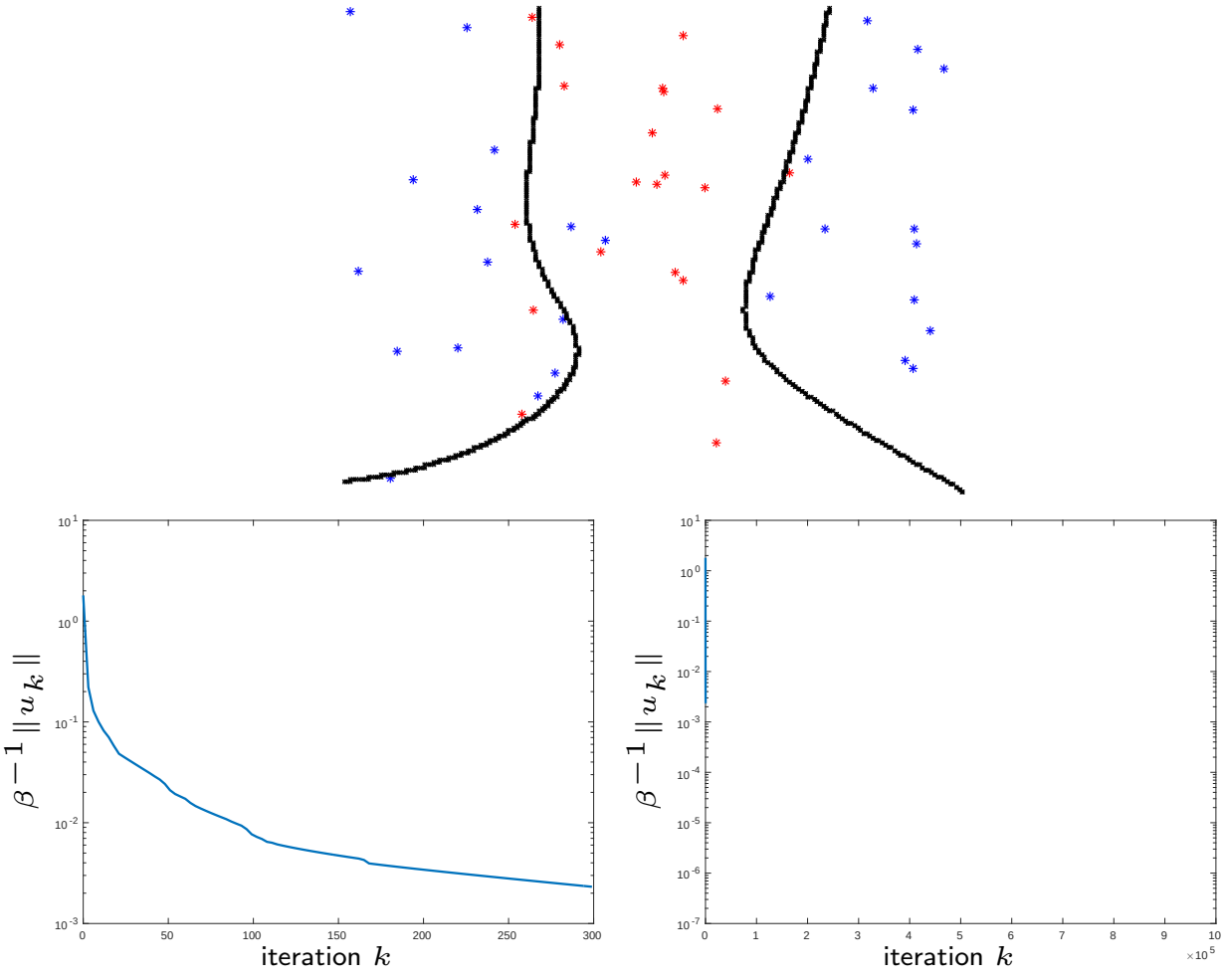
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 190 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.6e^{-3}$



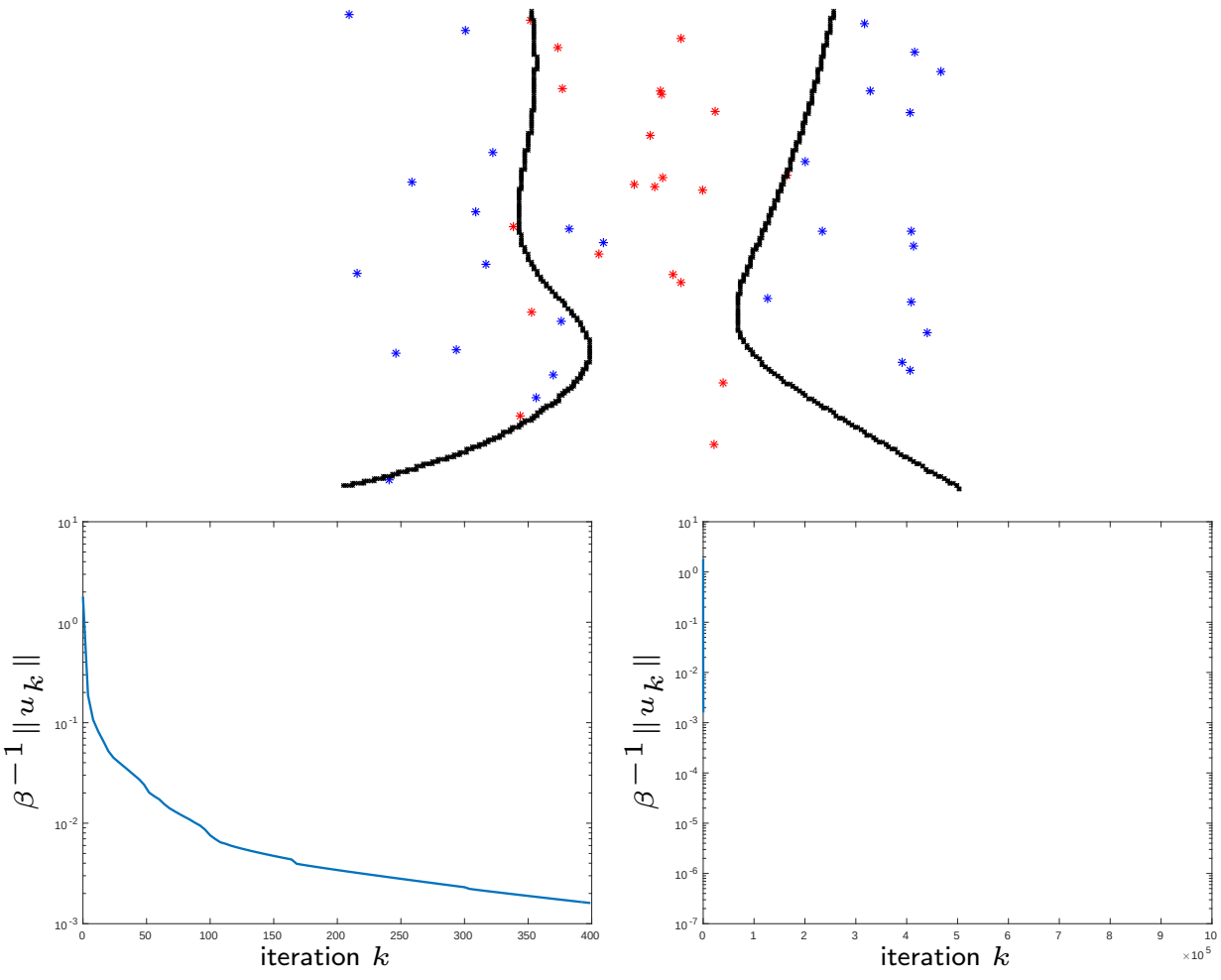
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 300 Residual norm: $\beta^{-1}\|u_k\|_2 = 2.3e^{-3}$



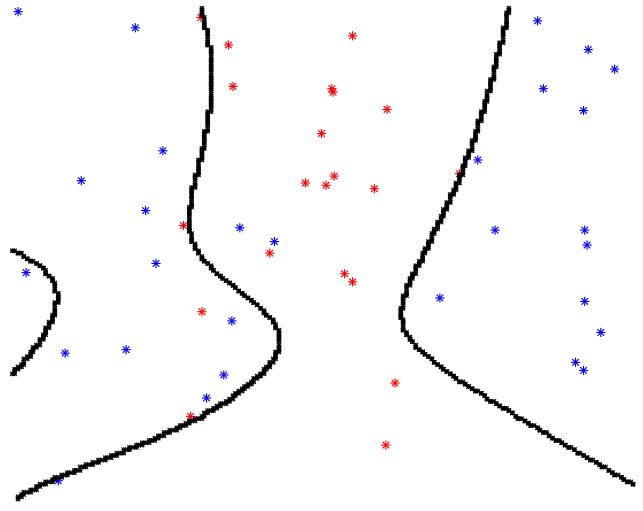
Early termination

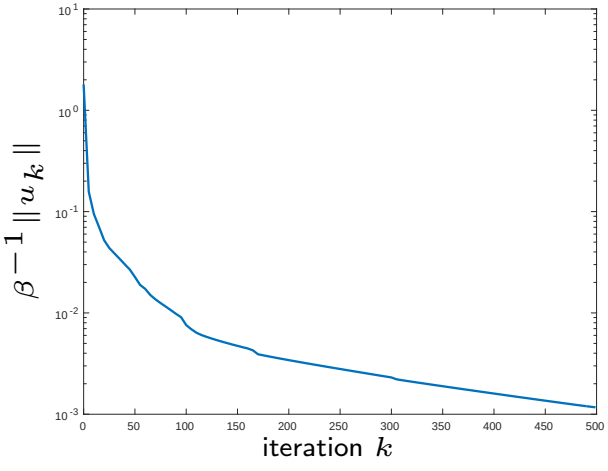
- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 400 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.6e^{-3}$

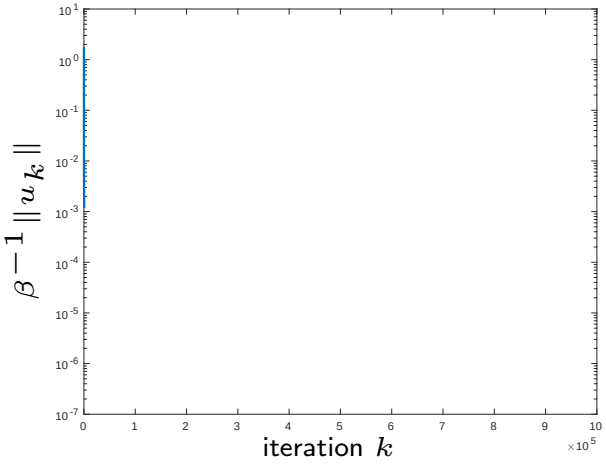


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 500 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.2e^{-3}$

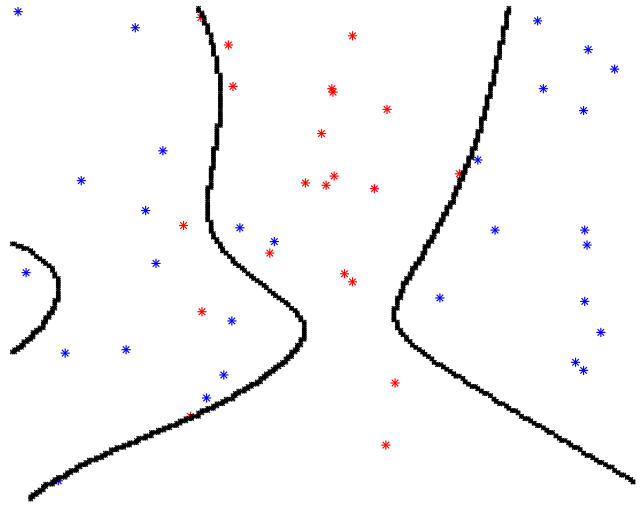


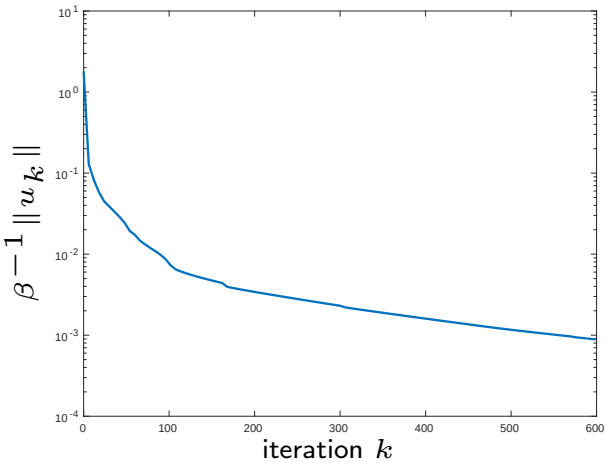


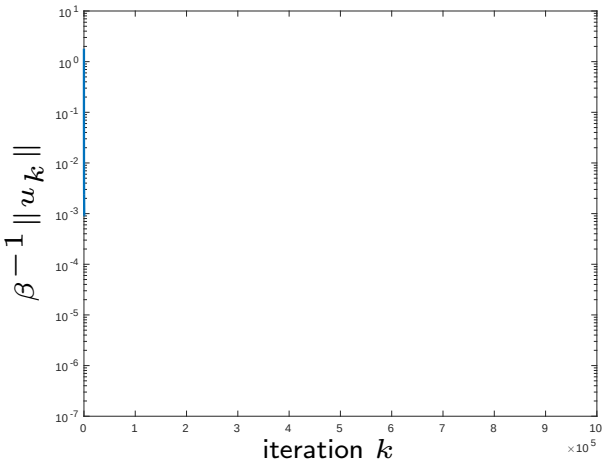


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 600 Residual norm: $\beta^{-1}\|u_k\|_2 = 8.9e^{-4}$

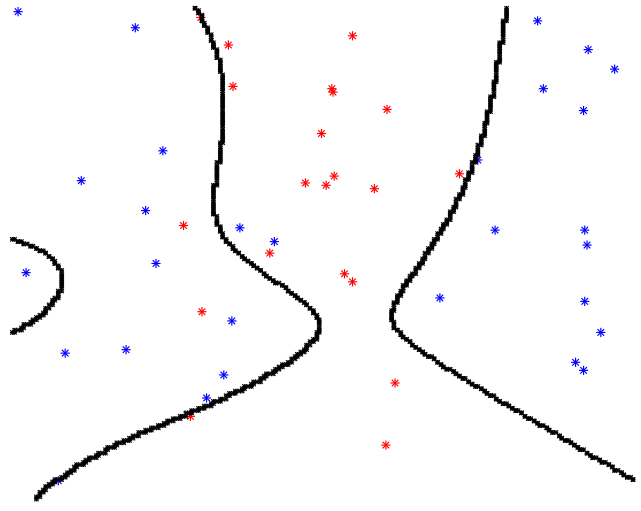


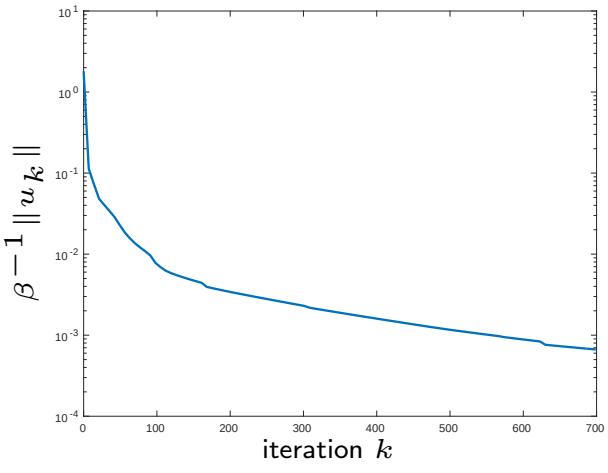


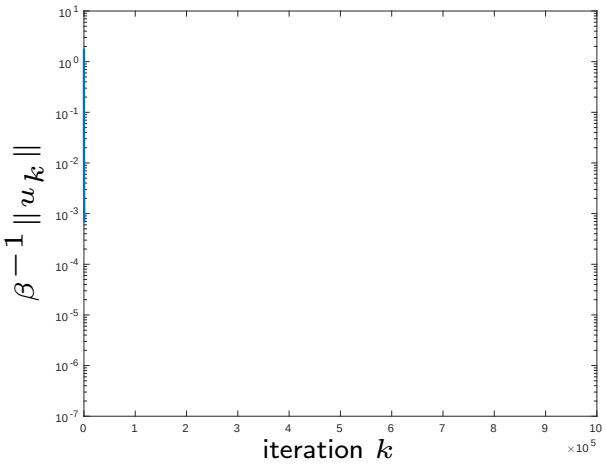


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 700 Residual norm: $\beta^{-1}\|u_k\|_2 = 6.7e^{-4}$

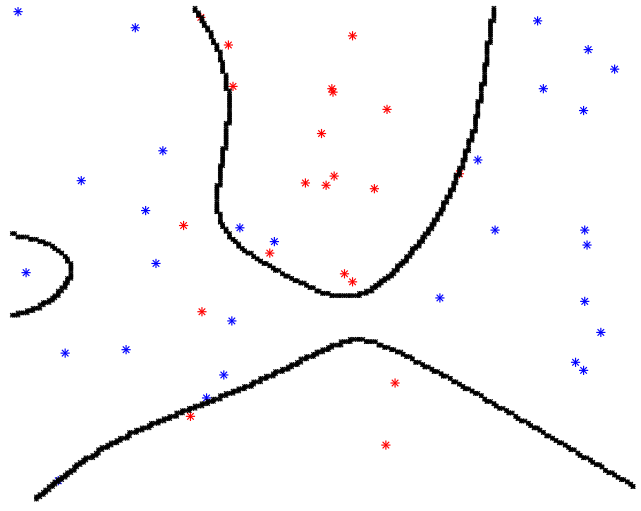


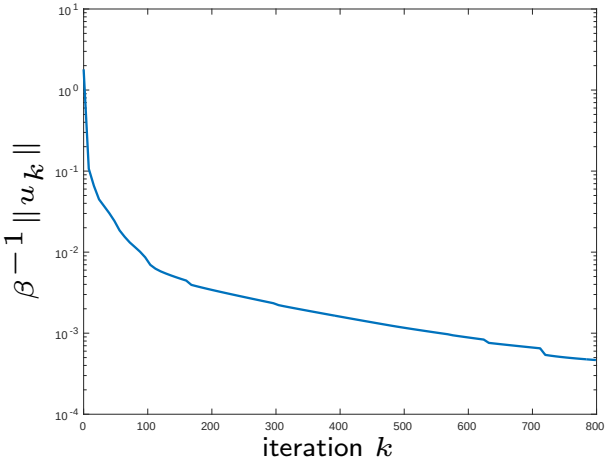


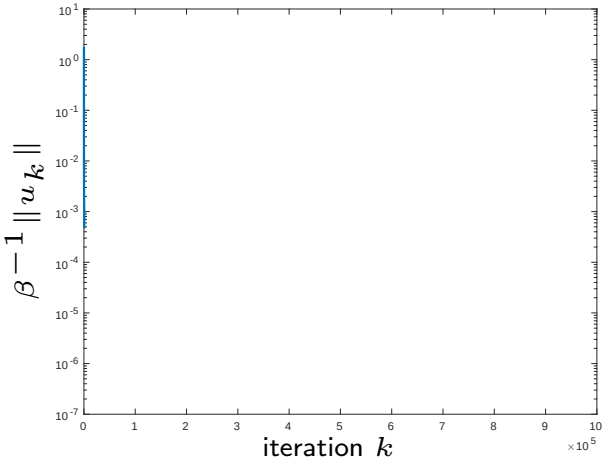


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 800 Residual norm: $\beta^{-1}\|u_k\|_2 = 4.7e^{-4}$

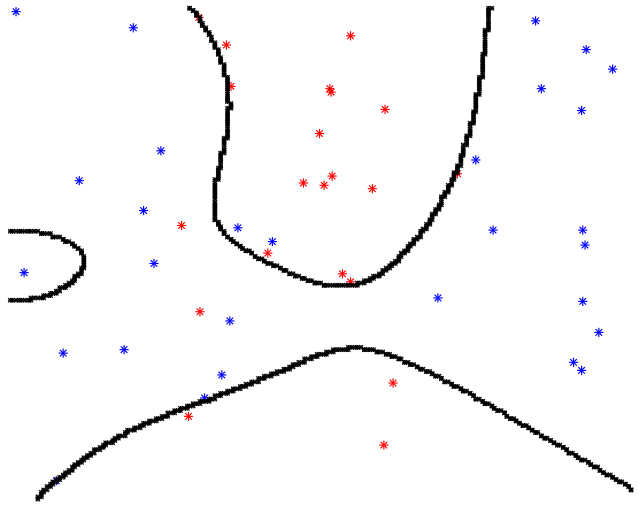


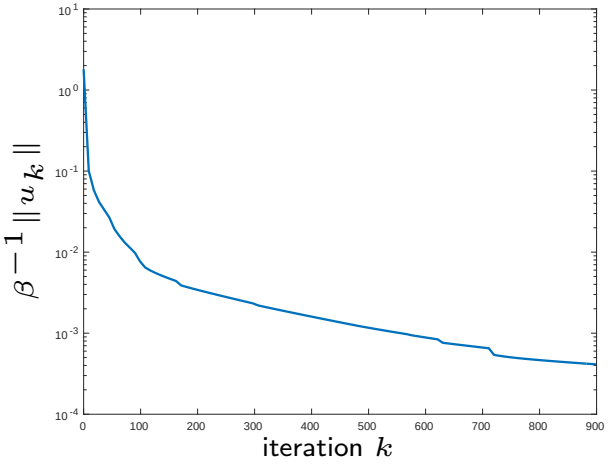


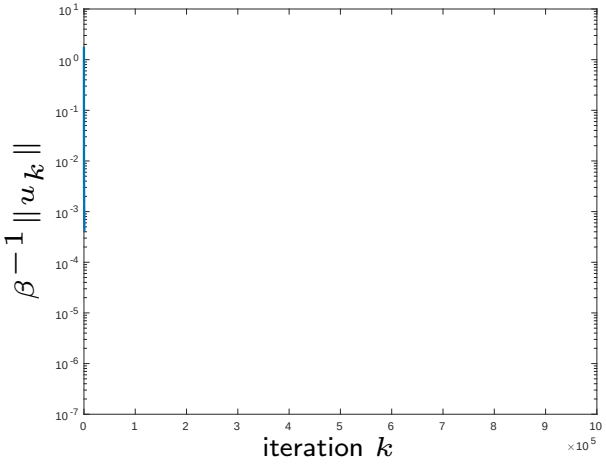


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 900 Residual norm: $\beta^{-1}\|u_k\|_2 = 4.1e^{-4}$

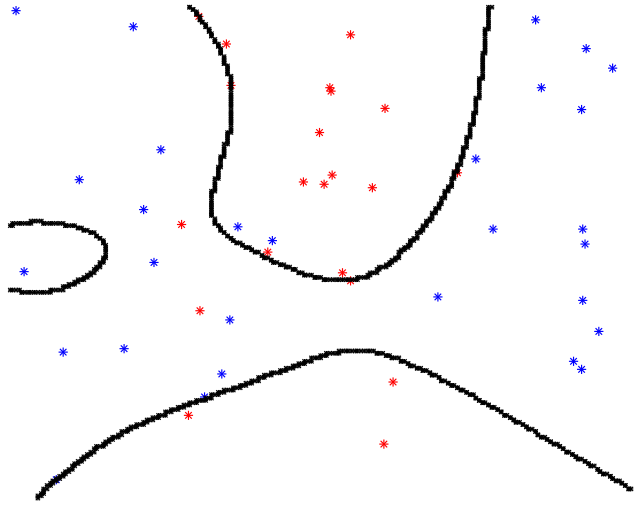


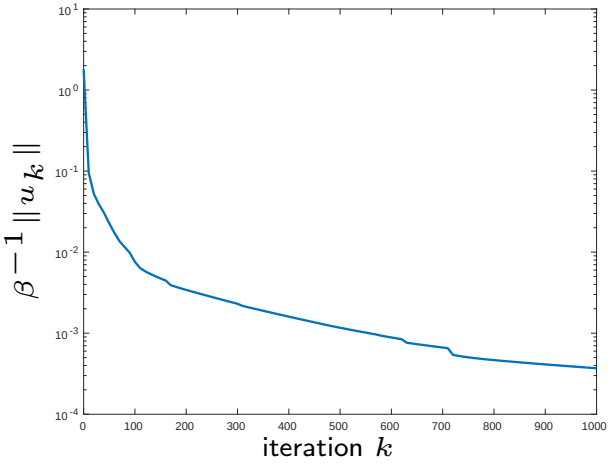


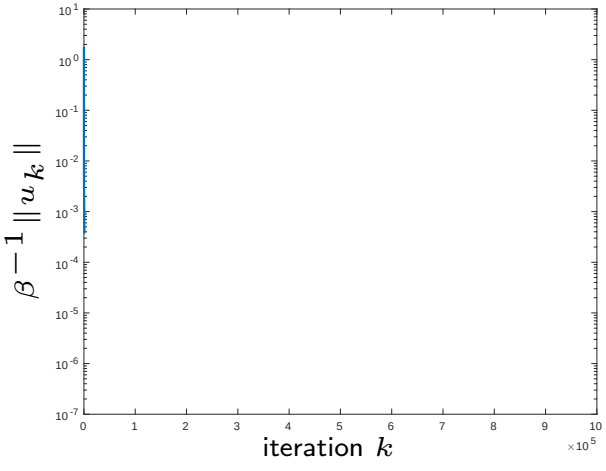


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 1000 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.7e^{-4}$

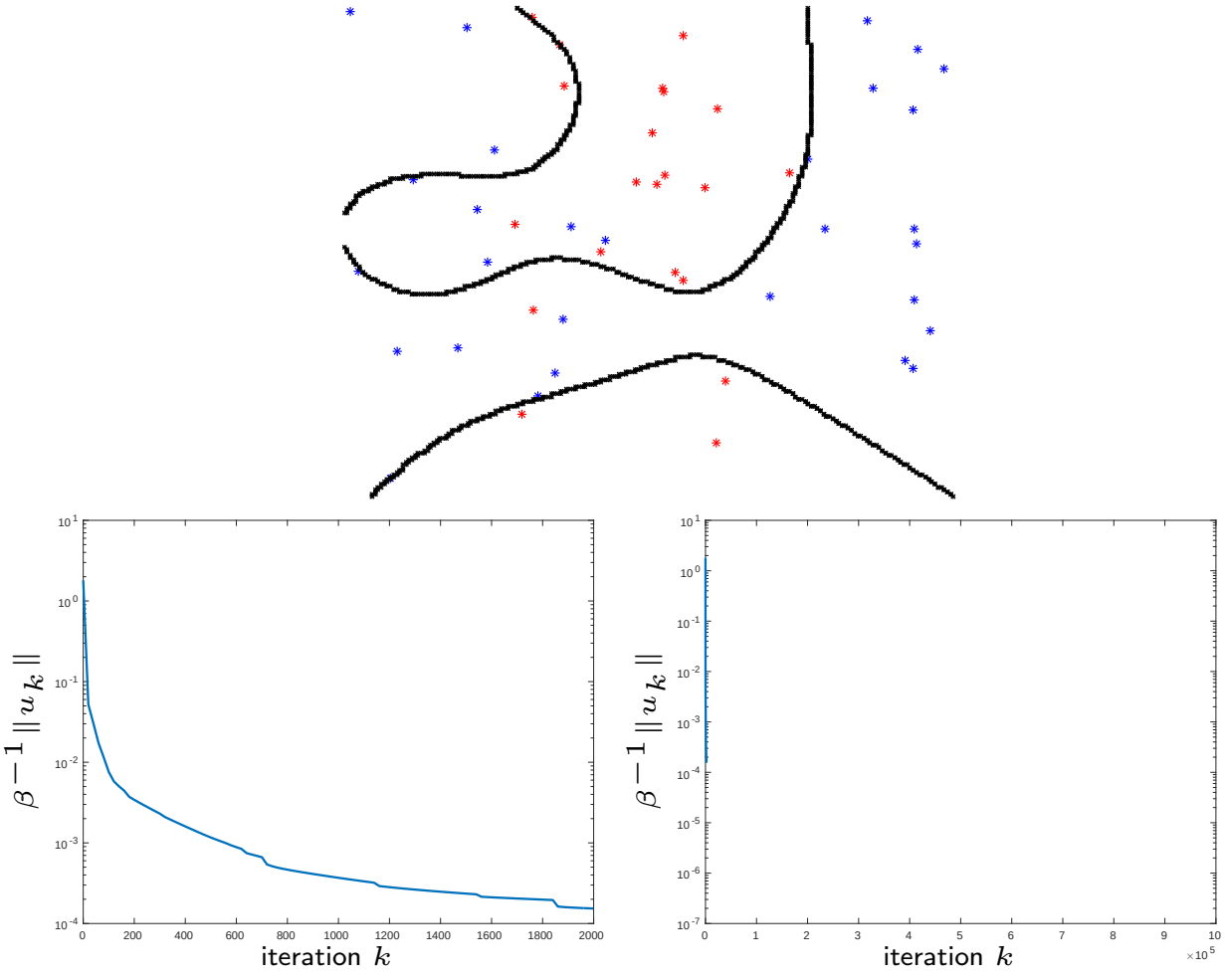






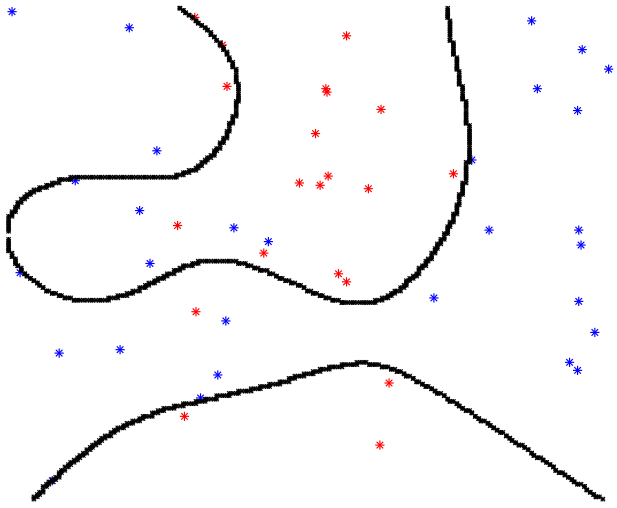
Early termination

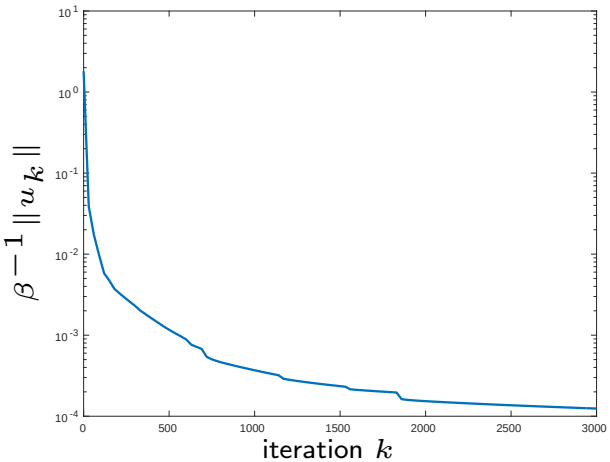
- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 2000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.5e^{-4}$

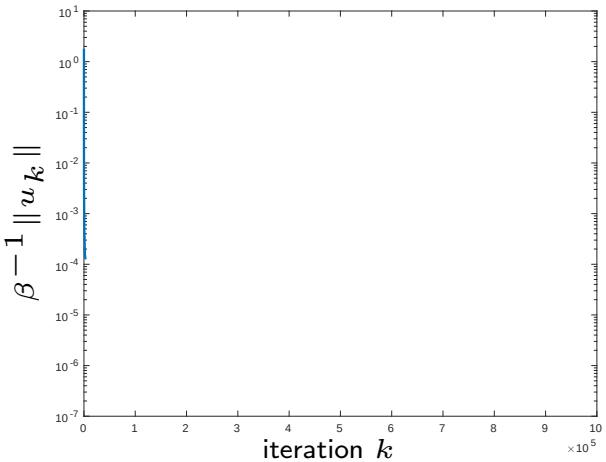


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 3000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.2e^{-4}$

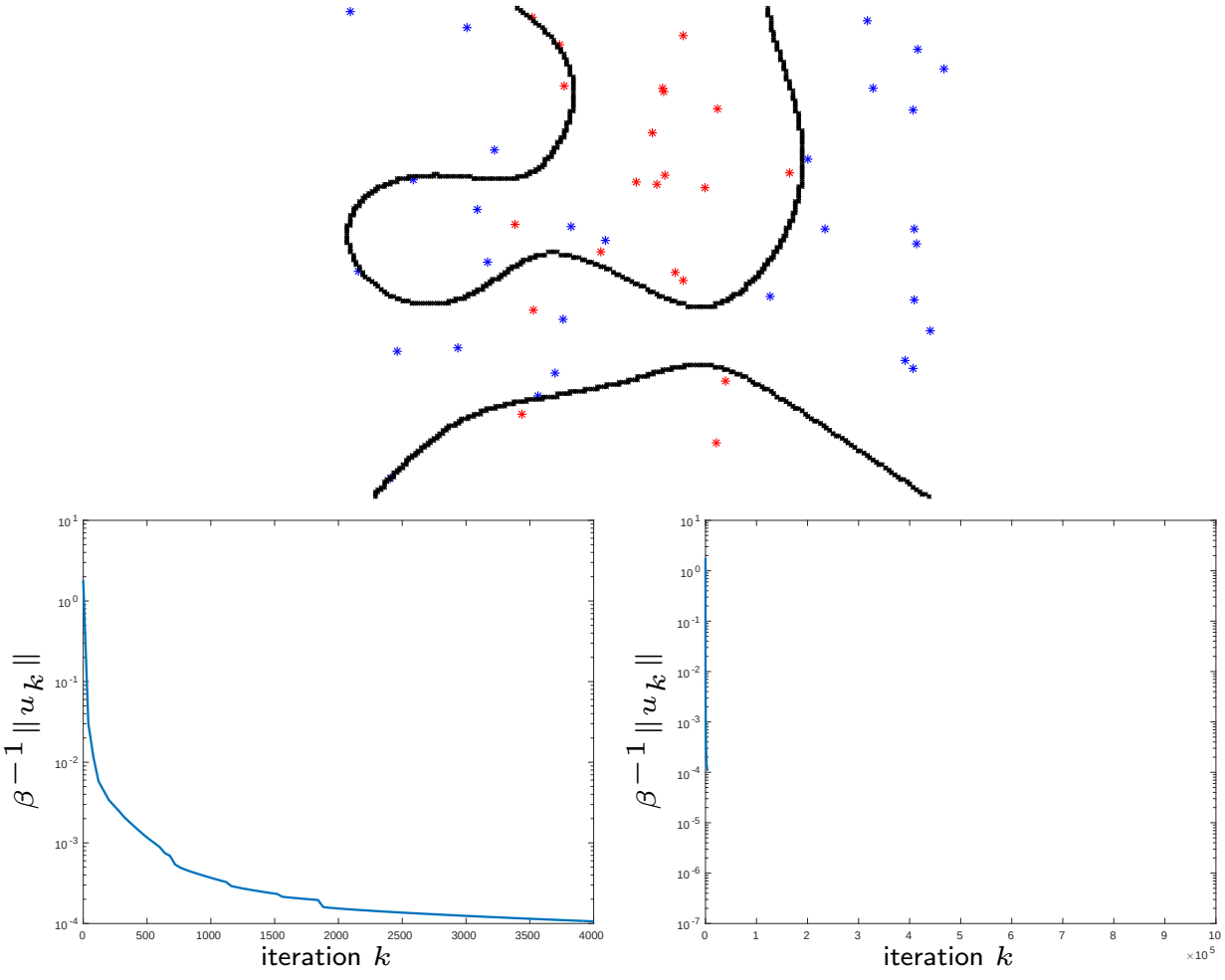






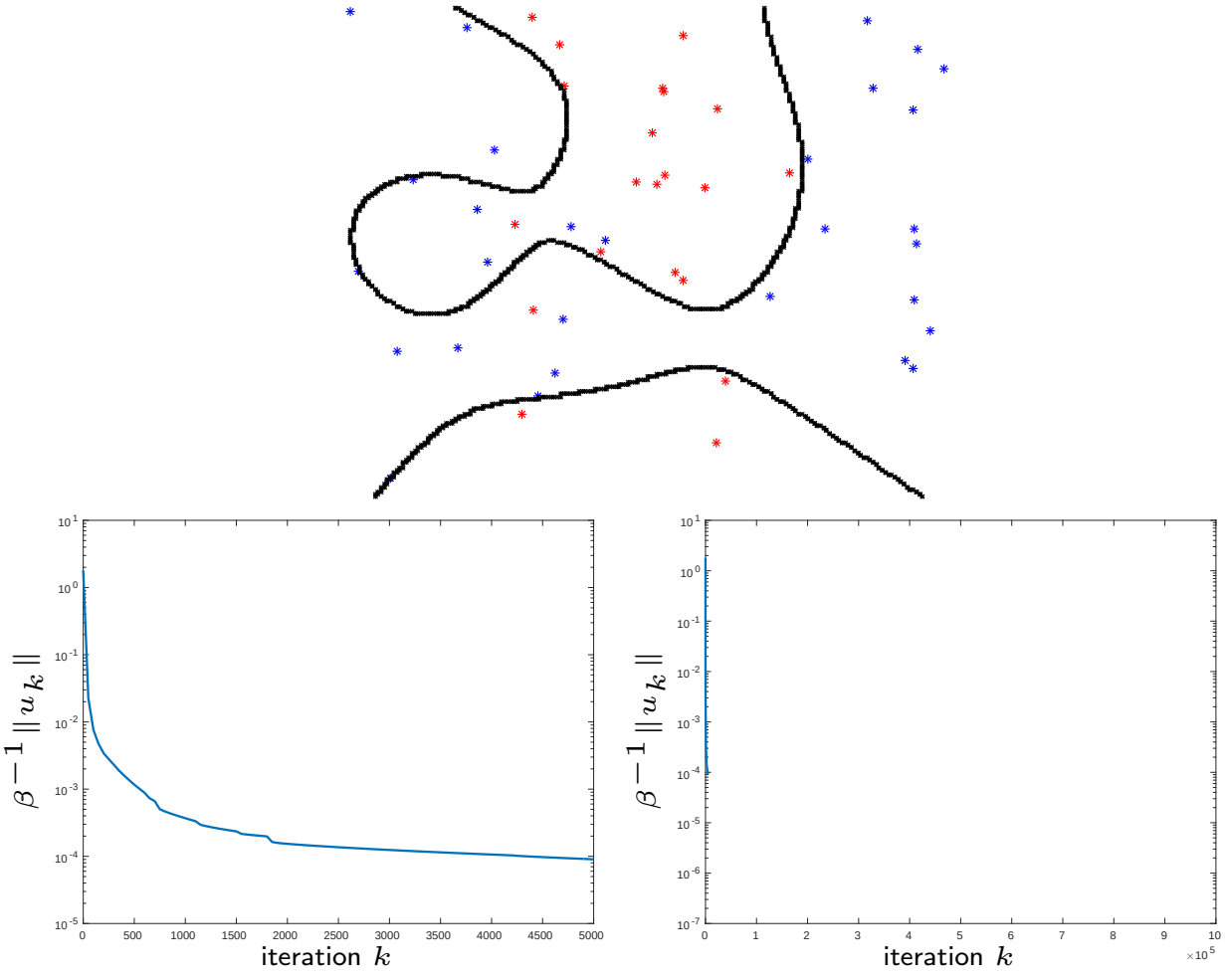
Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 4000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.1e^{-4}$



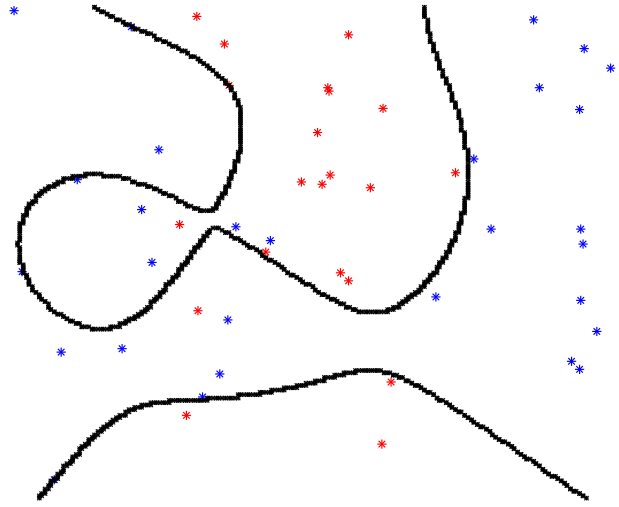
Early termination

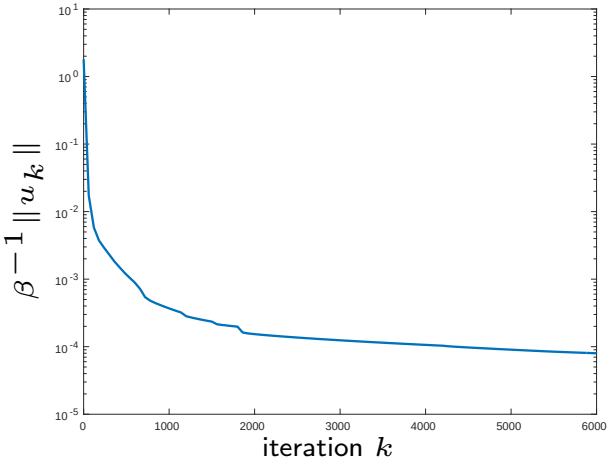
- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 5000 Residual norm: $\beta^{-1}\|u_k\|_2 = 9e^{-5}$

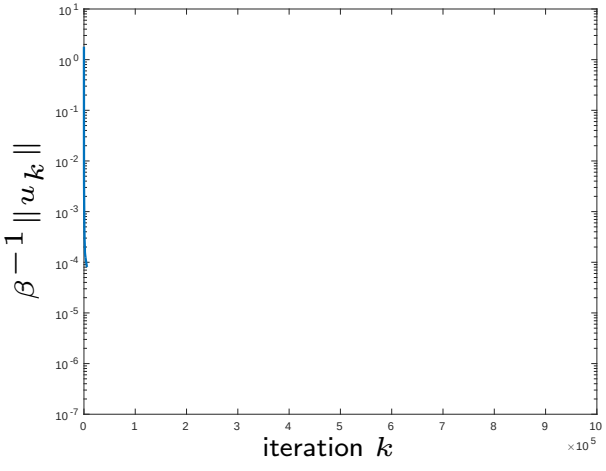


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 6000 Residual norm: $\beta^{-1}\|u_k\|_2 = 8e^{-5}$

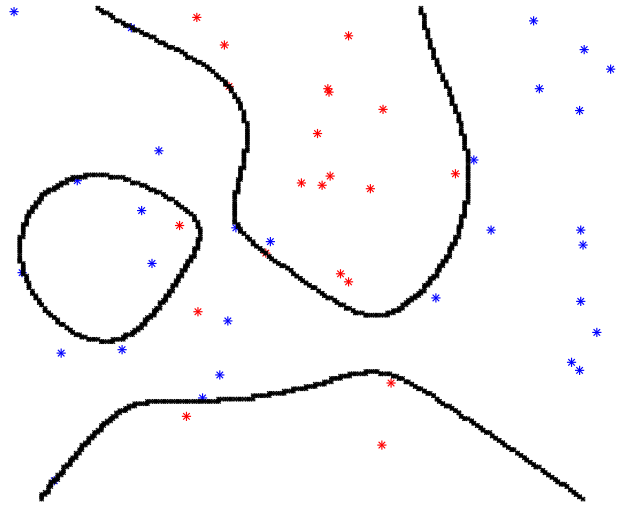


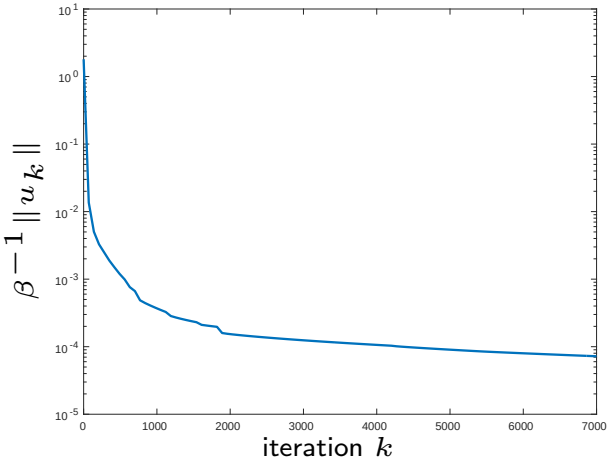


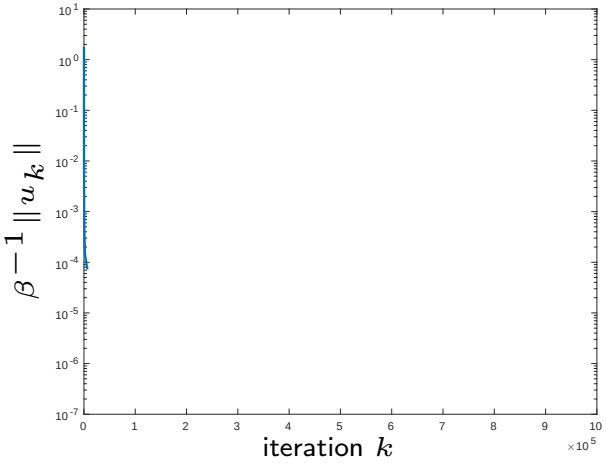


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 7000 Residual norm: $\beta^{-1}\|u_k\|_2 = 7.2e^{-5}$

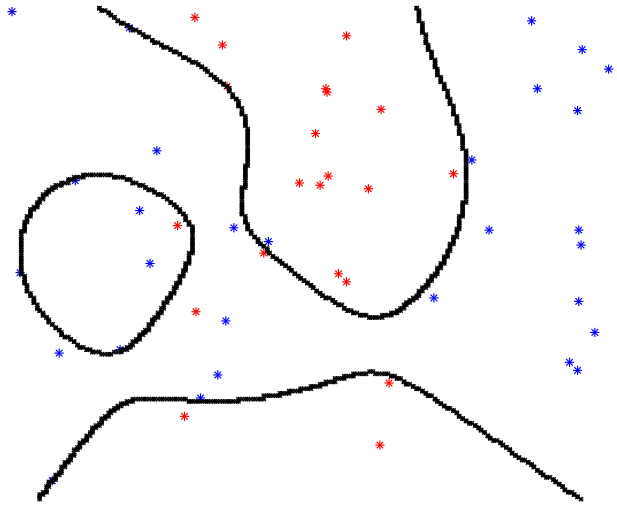


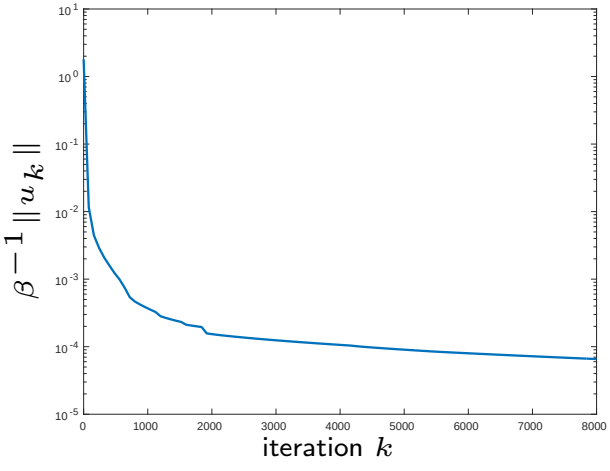


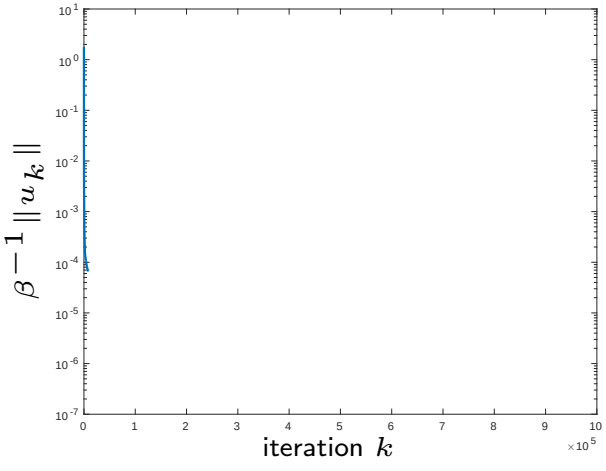


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 8000 Residual norm: $\beta^{-1}\|u_k\|_2 = 6.6e^{-5}$

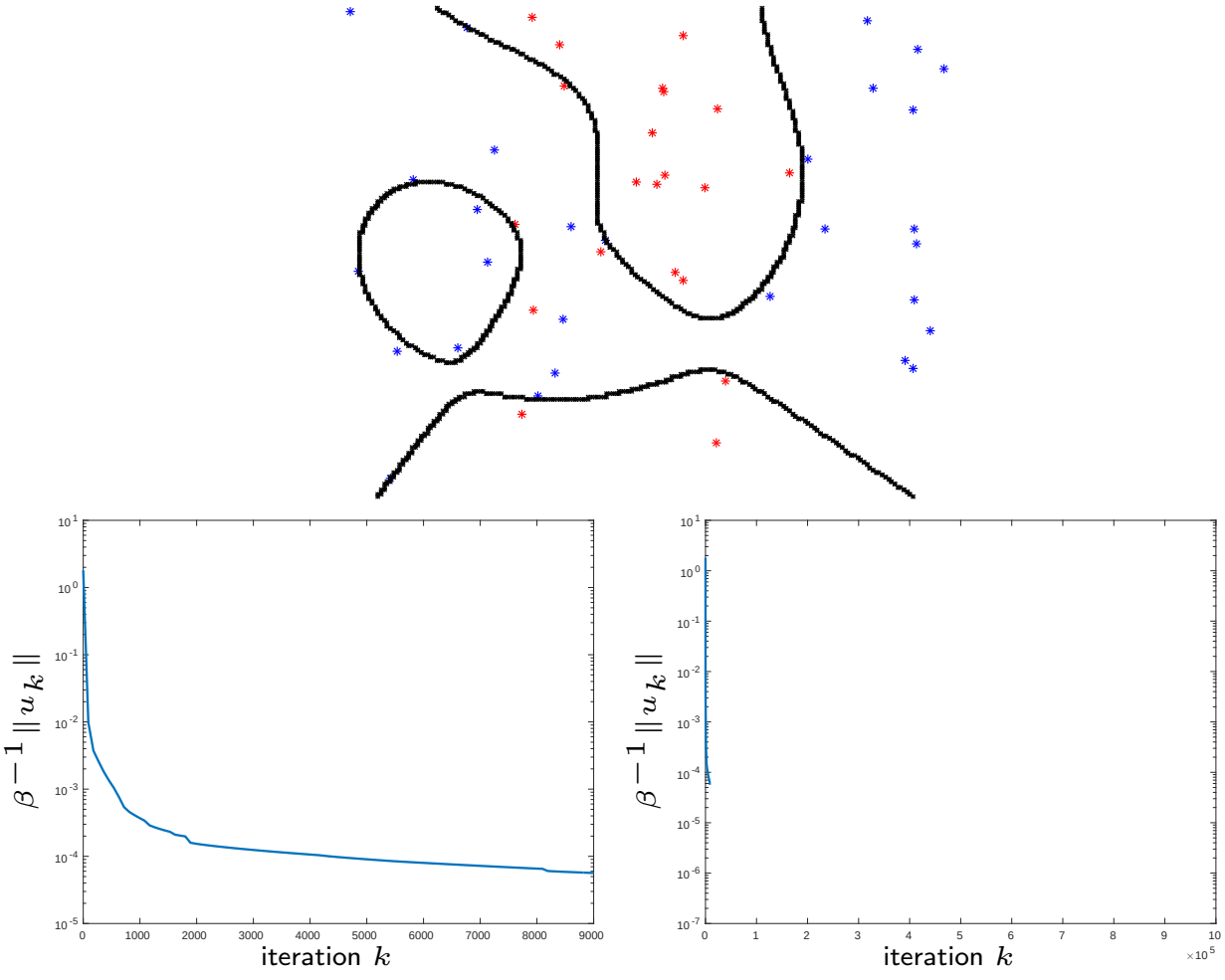






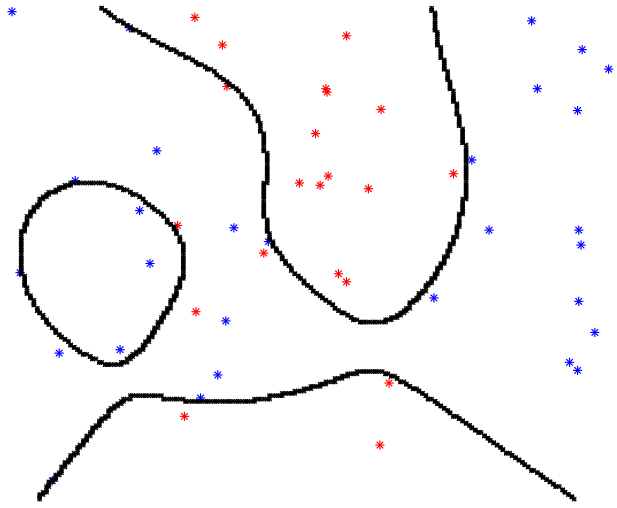
Early termination

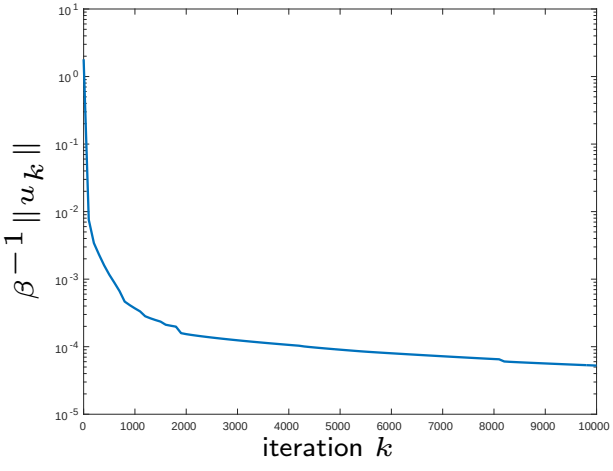
- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 9000 Residual norm: $\beta^{-1}\|u_k\|_2 = 5.6e^{-5}$

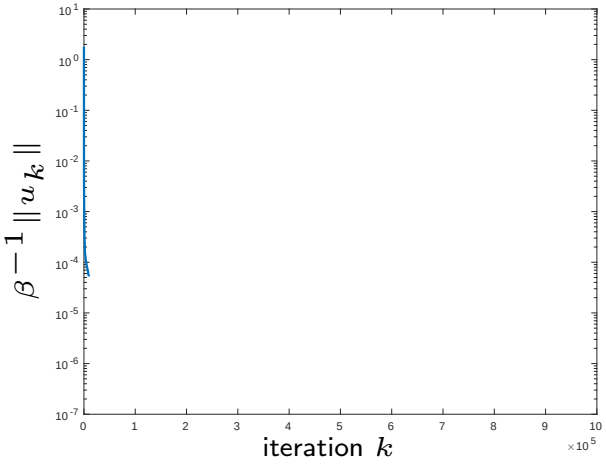


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 10000 Residual norm: $\beta^{-1}\|u_k\|_2 = 5.3e^{-5}$

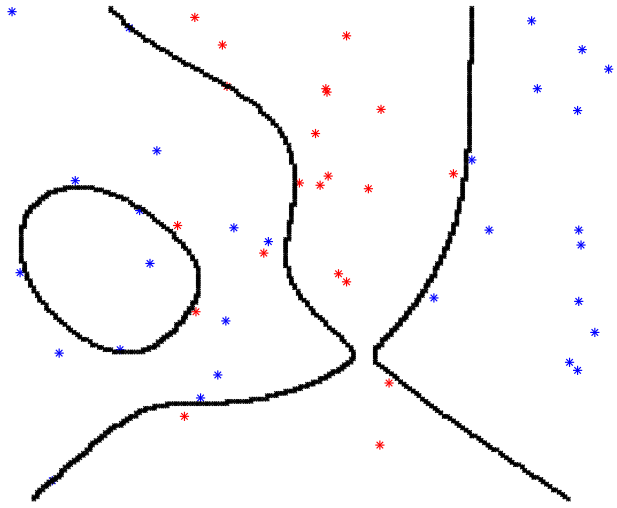


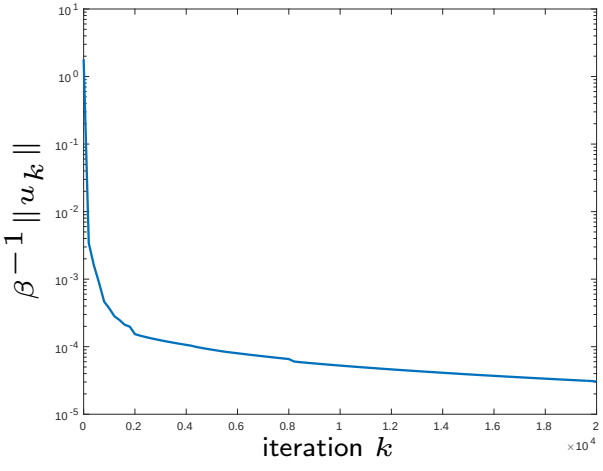


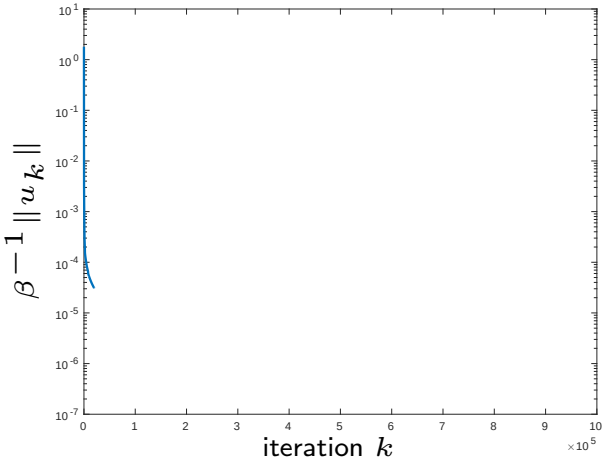


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 20000 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.1e^{-5}$

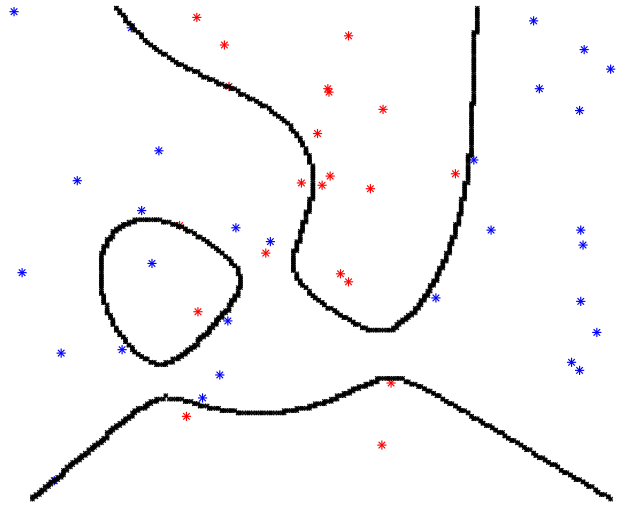


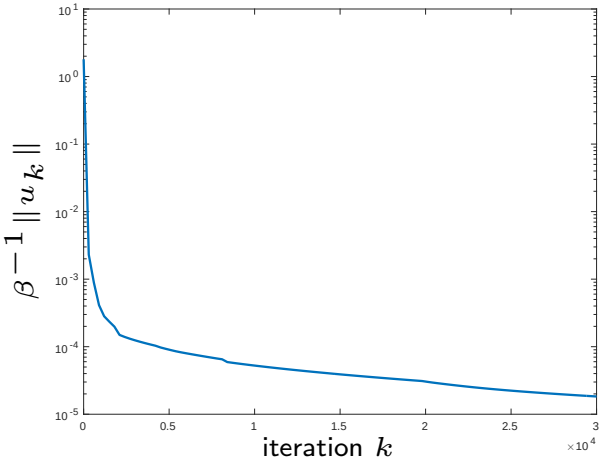


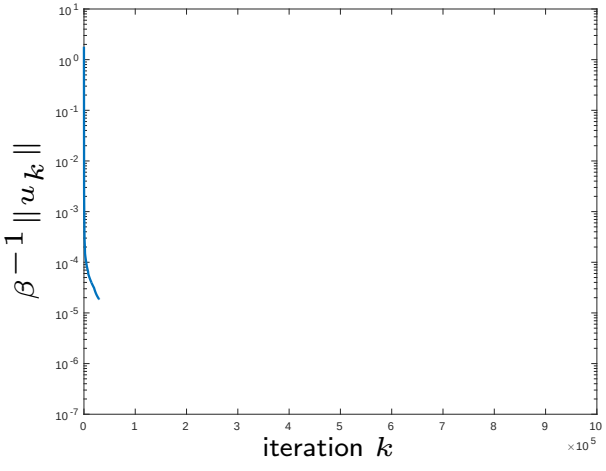


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 30000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.8e^{-5}$

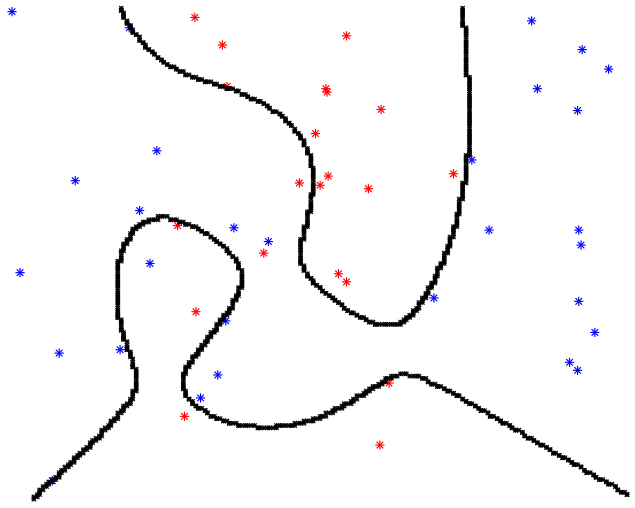


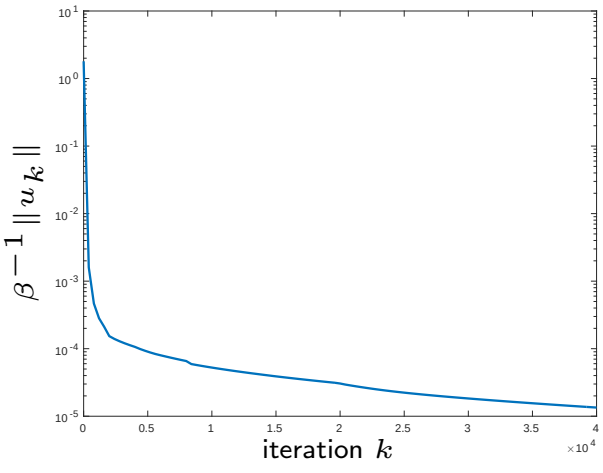


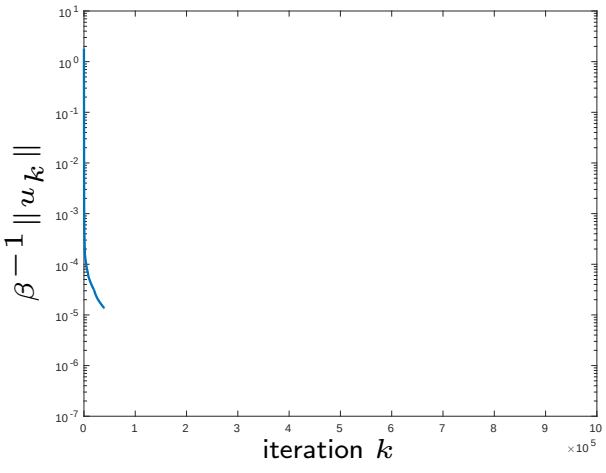


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 40000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.3e^{-5}$

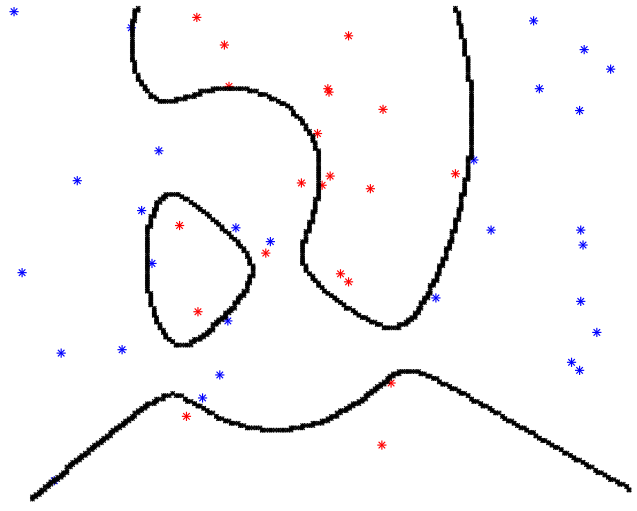


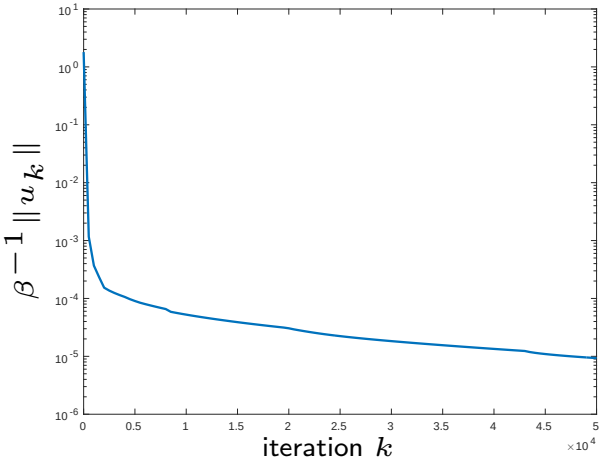


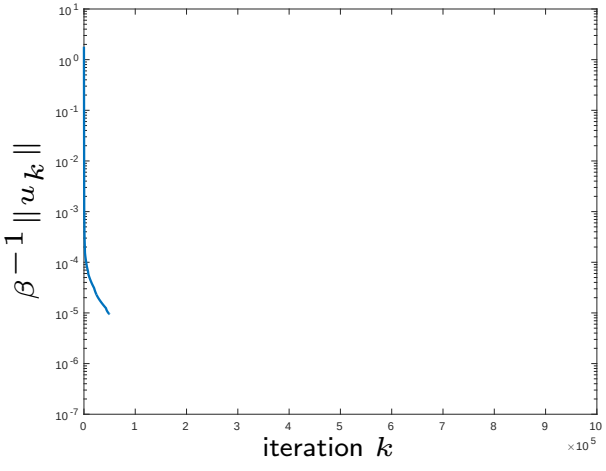


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 50000 Residual norm: $\beta^{-1}\|u_k\|_2 = 9.3e^{-6}$

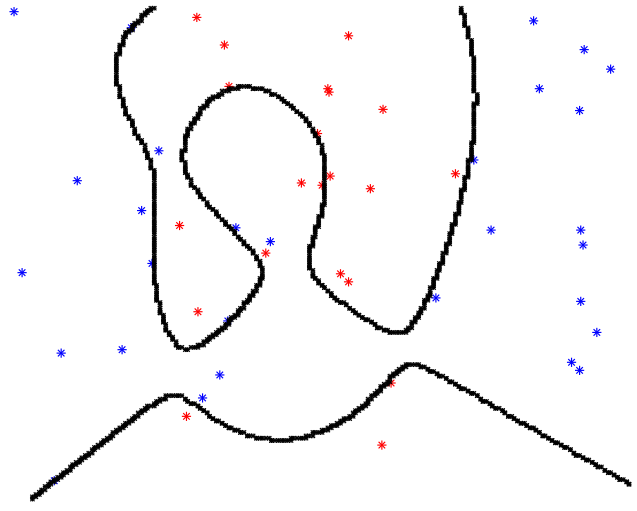


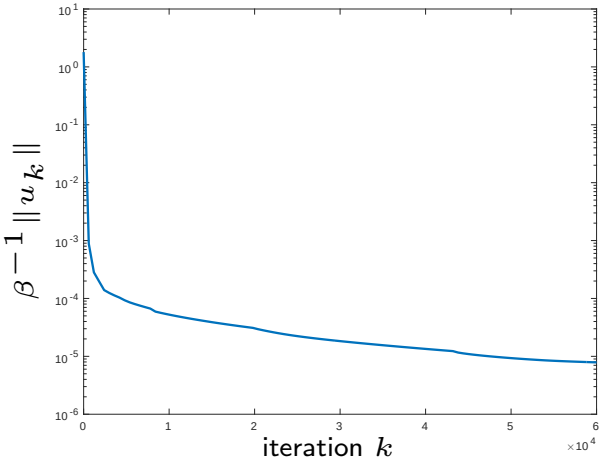


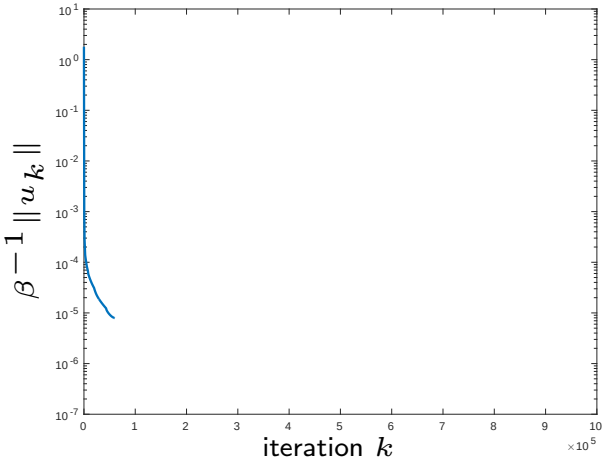


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 60000 Residual norm: $\beta^{-1}\|u_k\|_2 = 7.9e^{-6}$

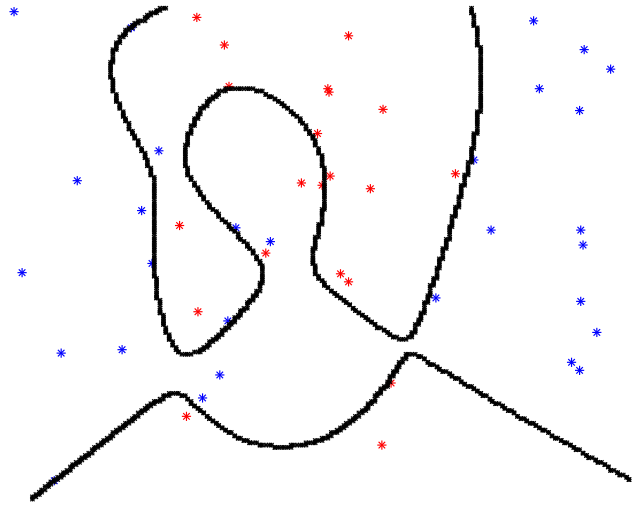


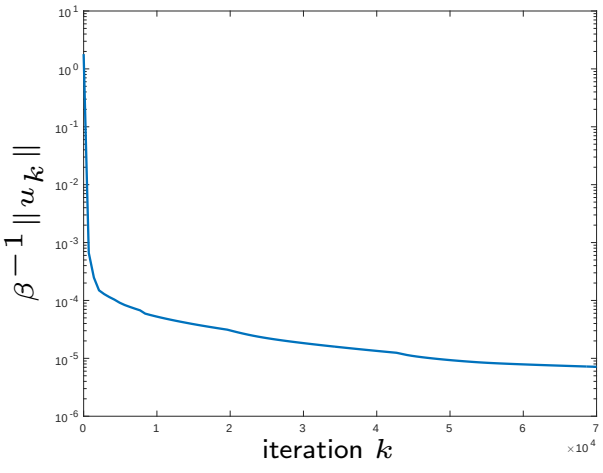


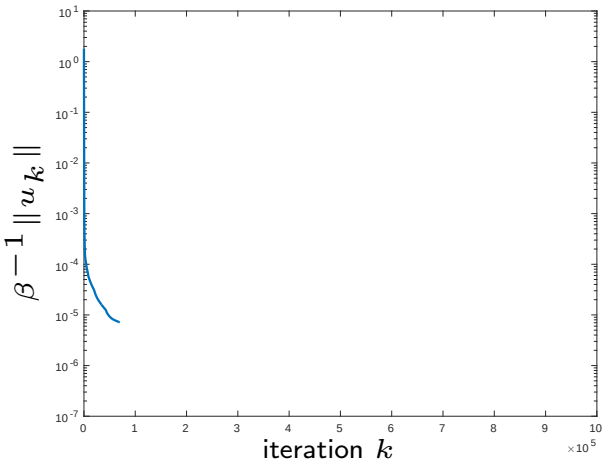


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 70000 Residual norm: $\beta^{-1}\|u_k\|_2 = 7.1e^{-6}$

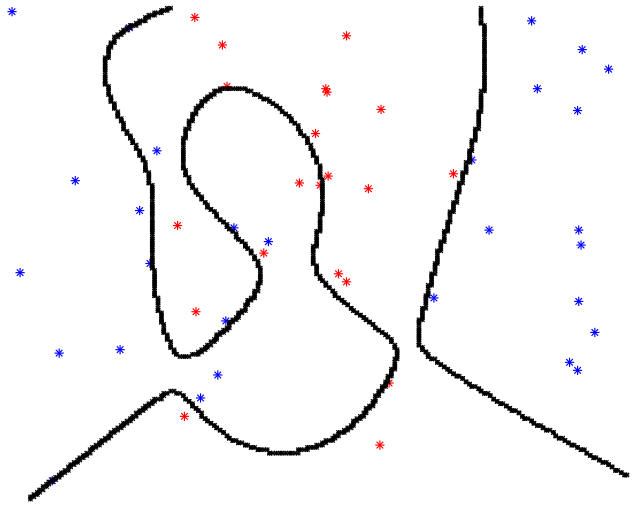


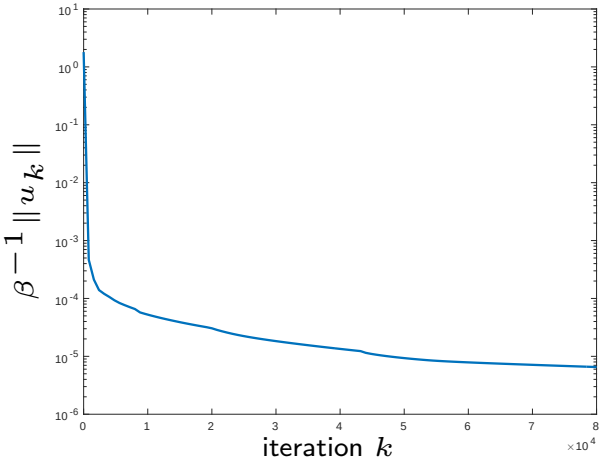


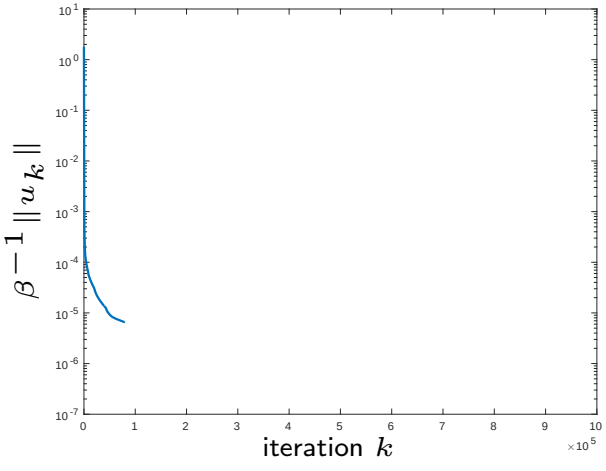


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 80000 Residual norm: $\beta^{-1}\|u_k\|_2 = 6.5e^{-6}$

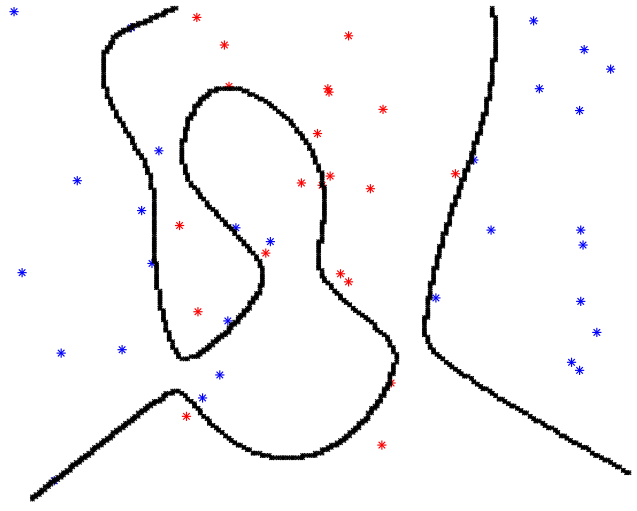


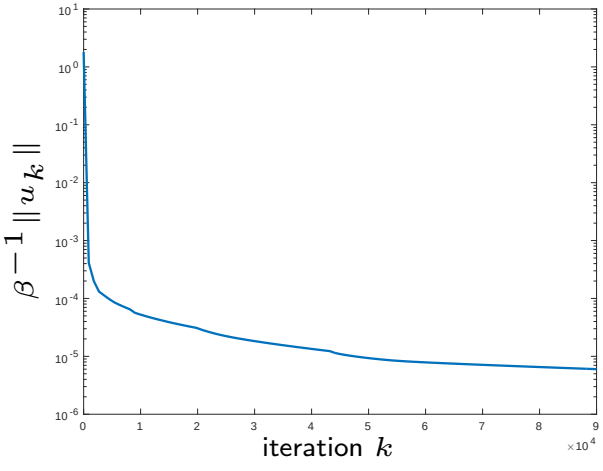


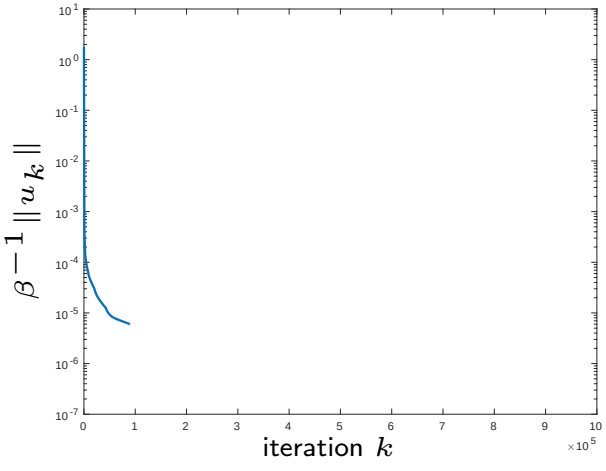


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 90000 Residual norm: $\beta^{-1}\|u_k\|_2 = 6e^{-6}$

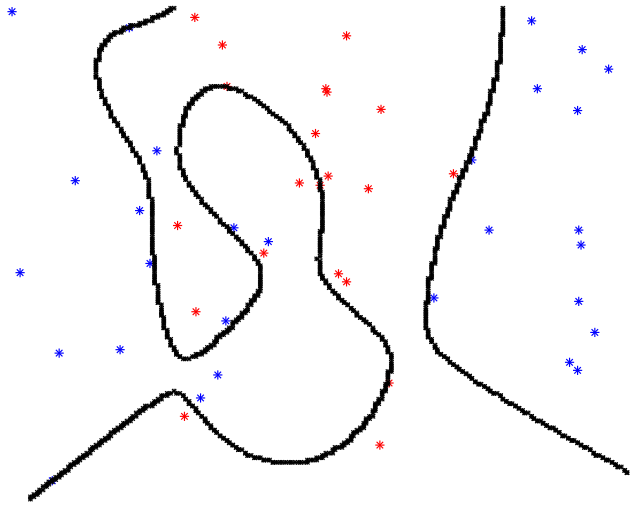


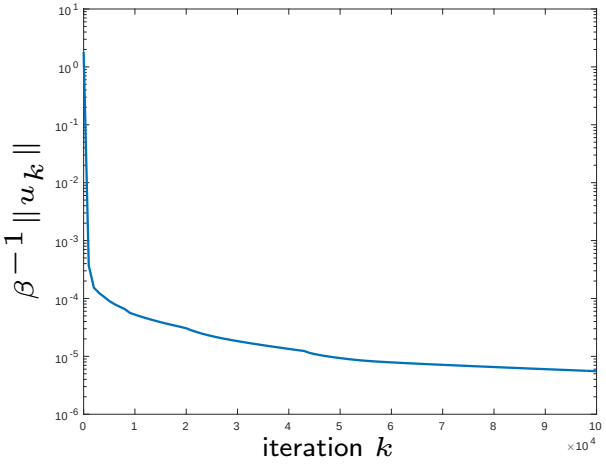


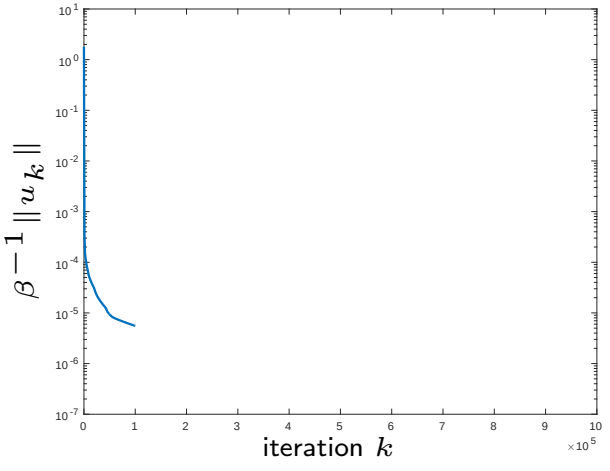


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 100000 Residual norm: $\beta^{-1}\|u_k\|_2 = 5.5e^{-6}$

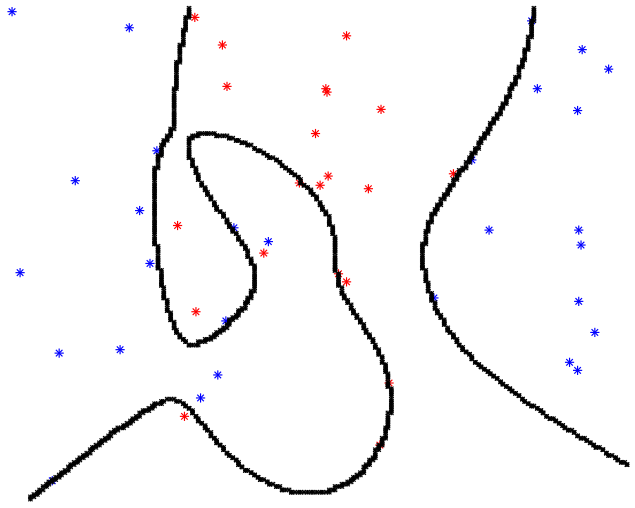


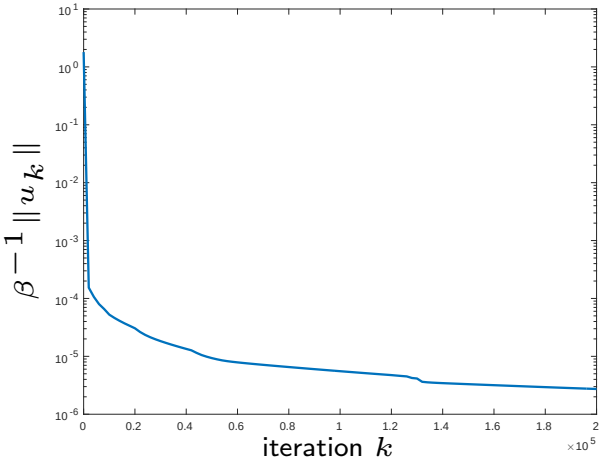


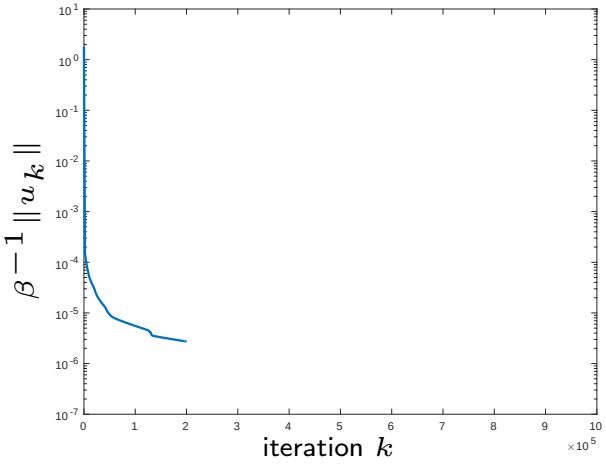


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 200000 Residual norm: $\beta^{-1}\|u_k\|_2 = 2.7e^{-6}$

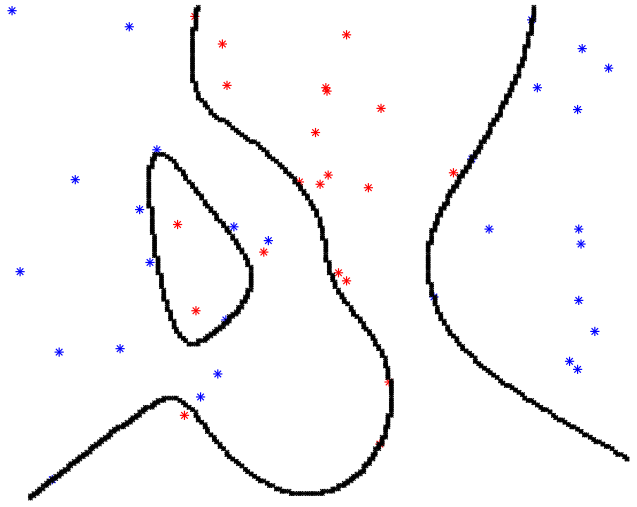


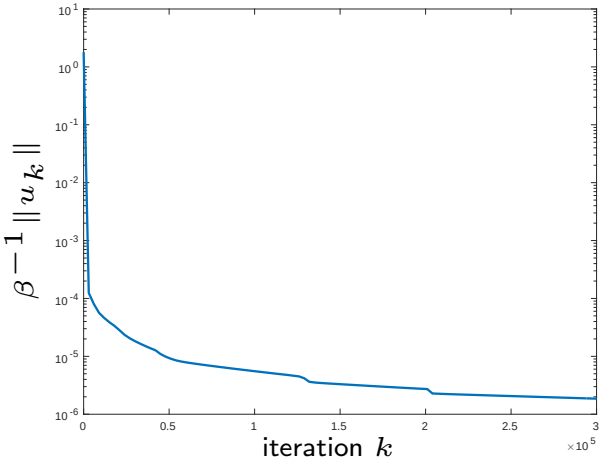


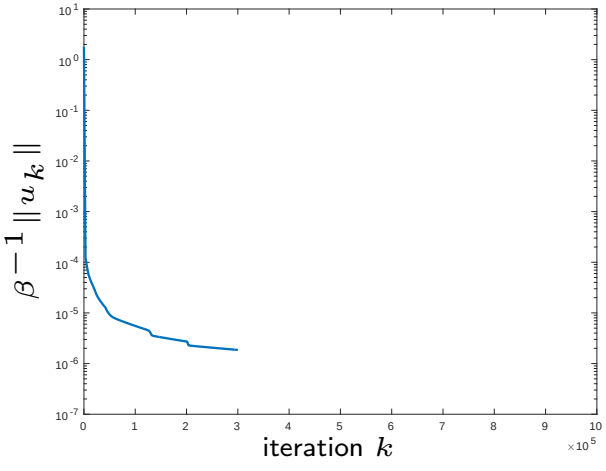


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 300000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.9e^{-6}$

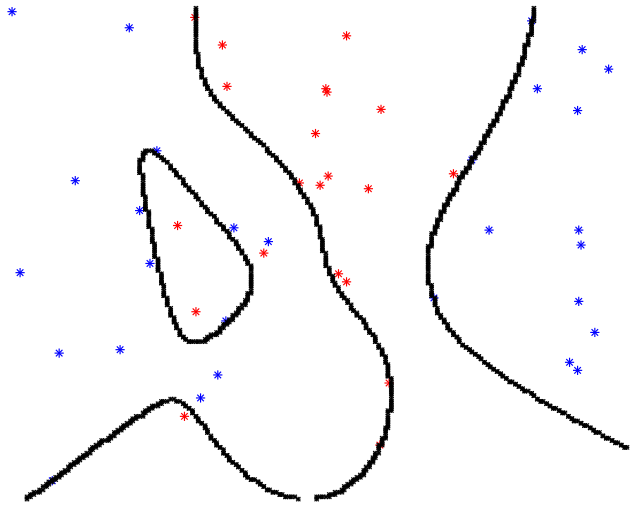


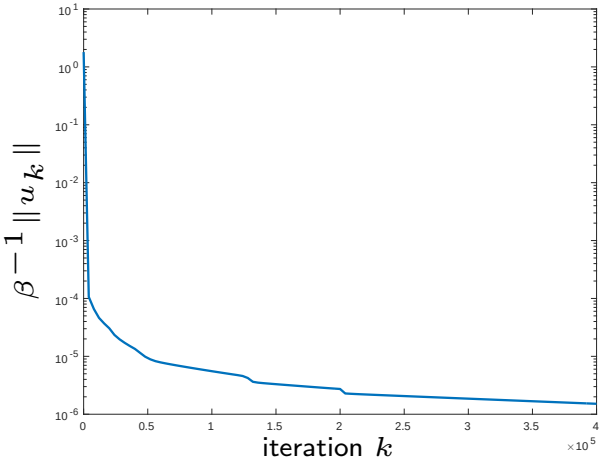


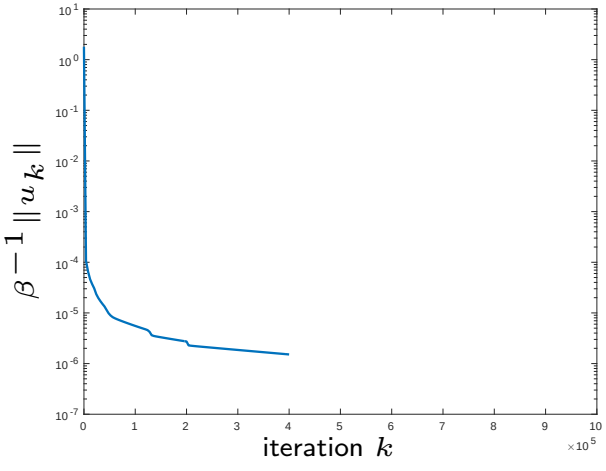


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 400000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.5e^{-6}$

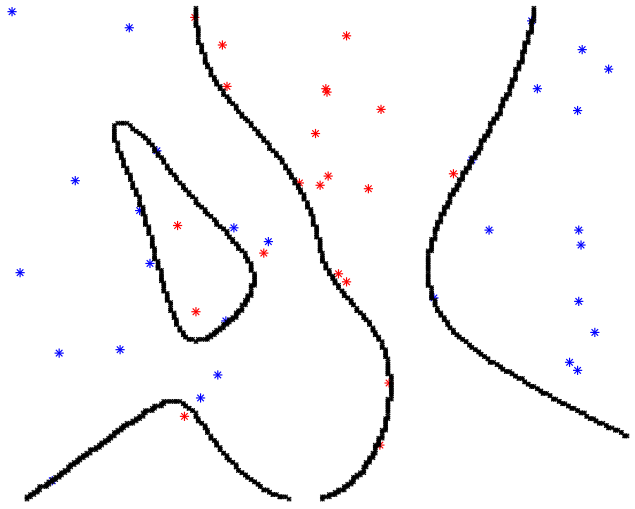


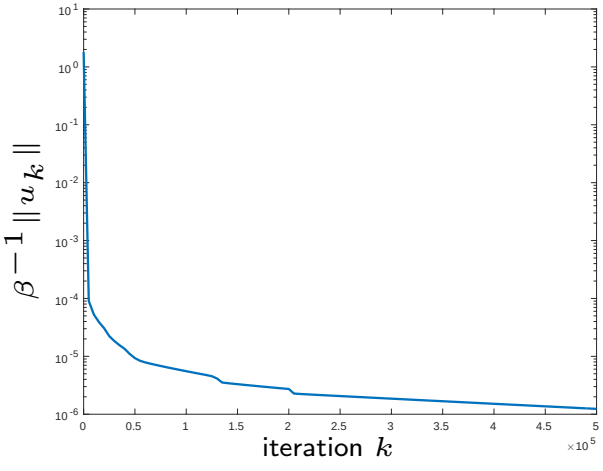


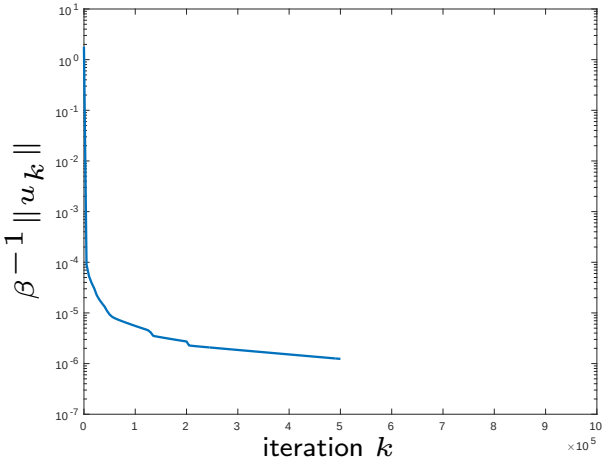


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 500000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1.2e^{-6}$

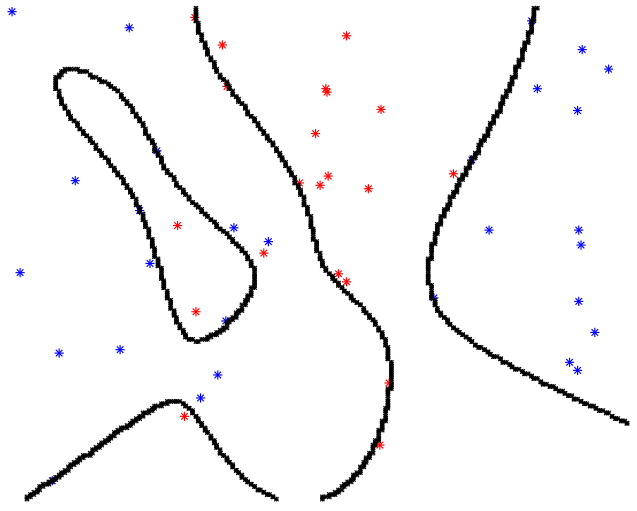


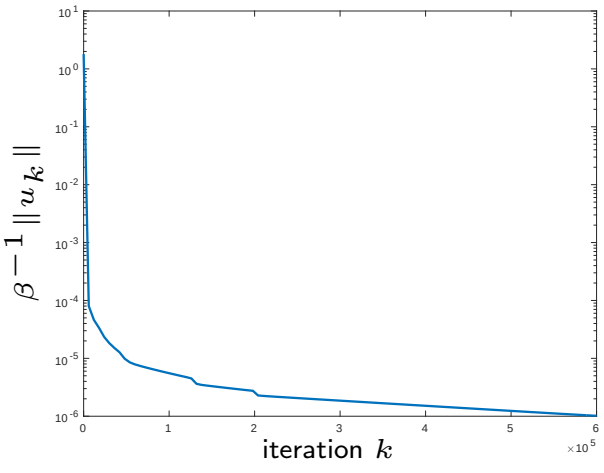


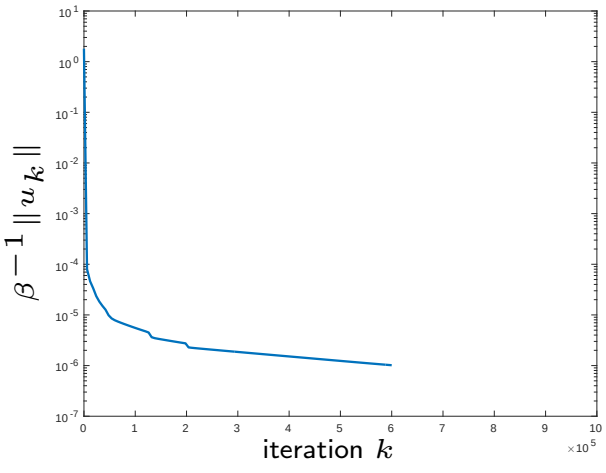


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 600000 Residual norm: $\beta^{-1}\|u_k\|_2 = 1e^{-6}$

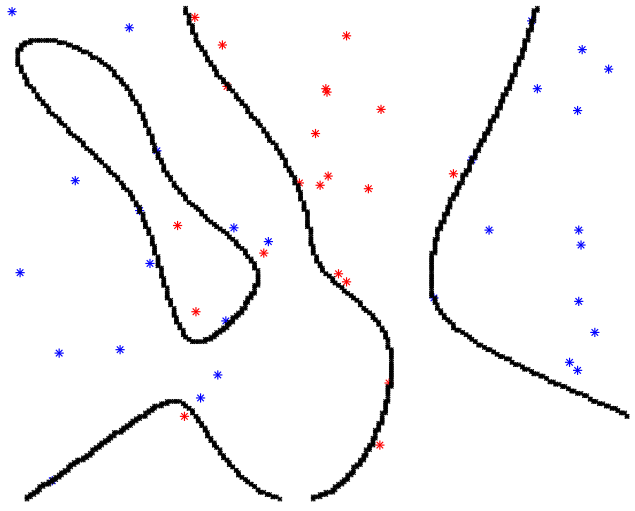


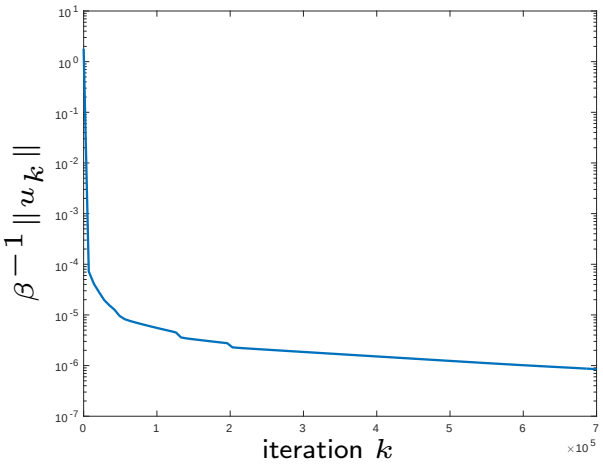


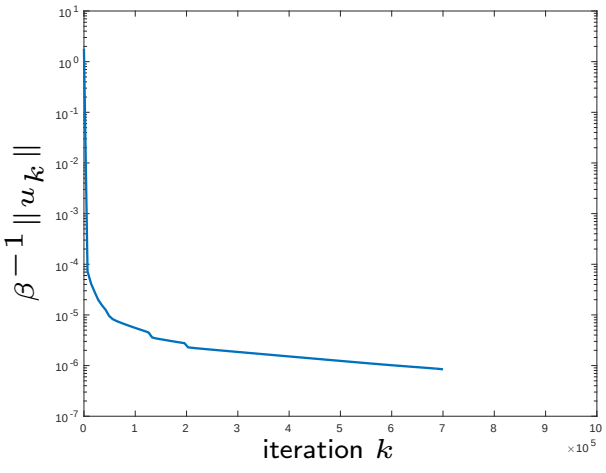


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 700000 Residual norm: $\beta^{-1}\|u_k\|_2 = 8.4e^{-7}$

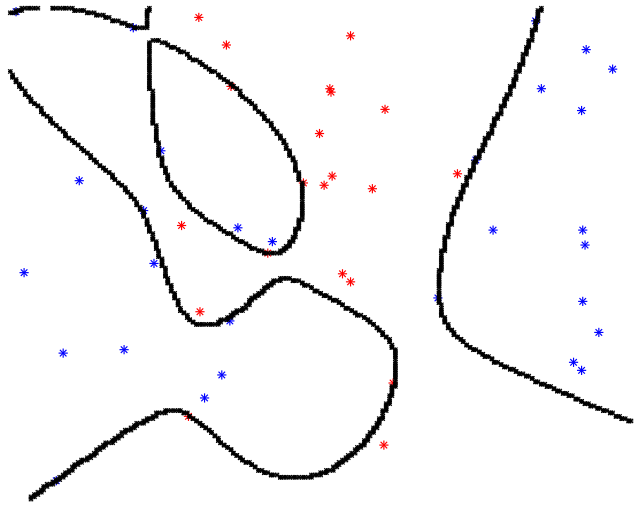


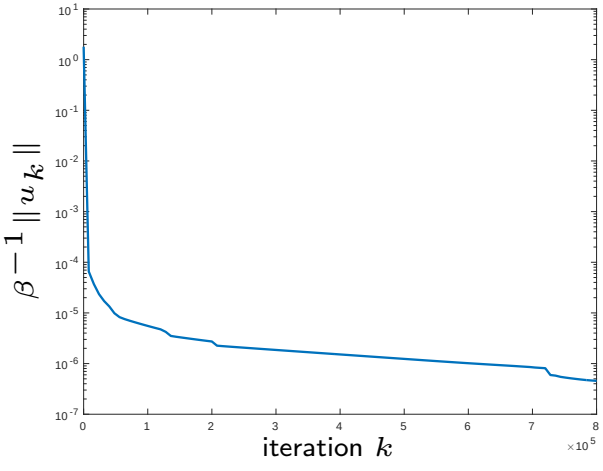


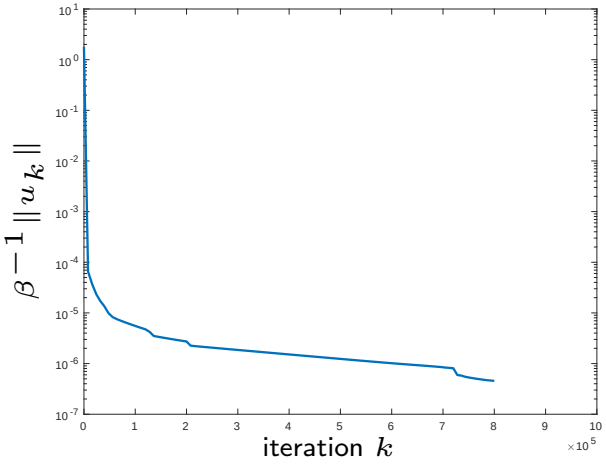


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 800000 Residual norm: $\beta^{-1}\|u_k\|_2 = 4.6e^{-7}$

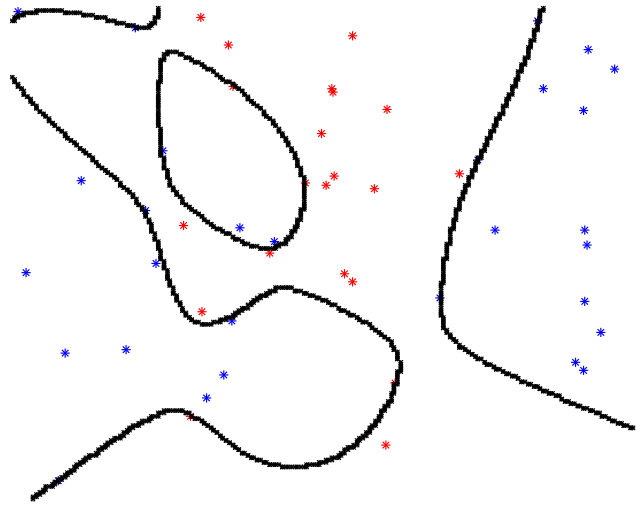


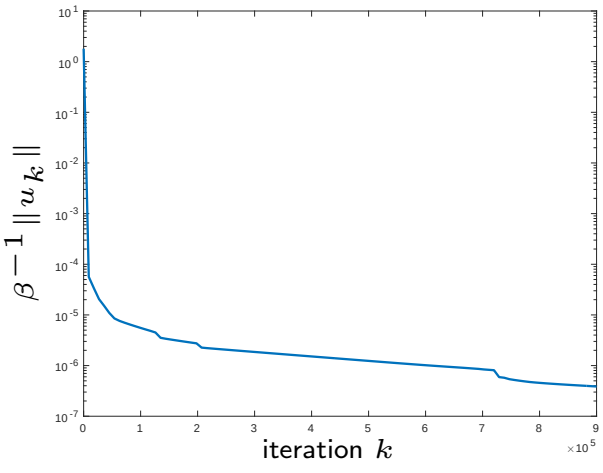


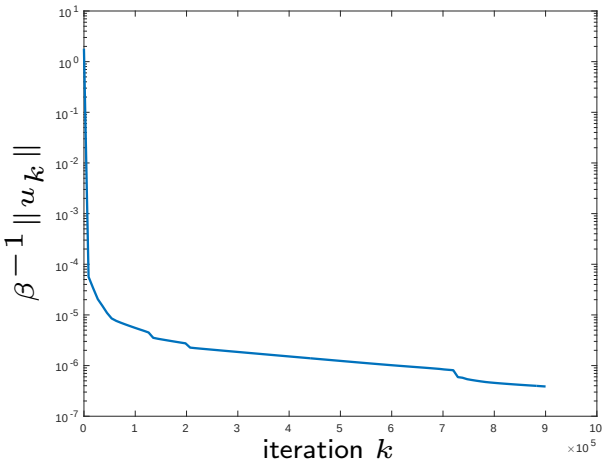


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 900000 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.9e^{-7}$

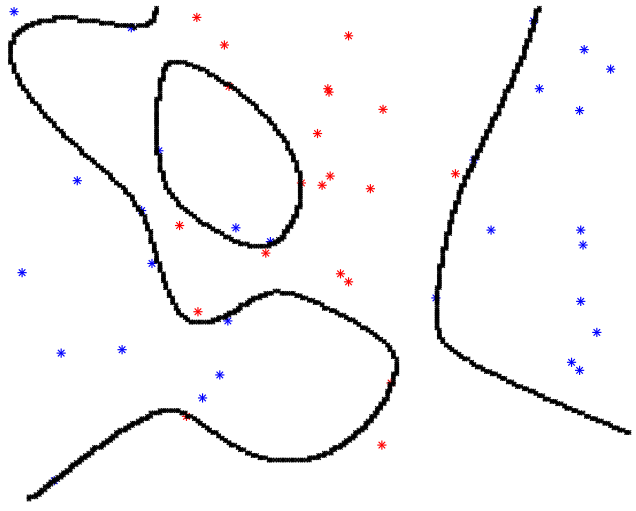


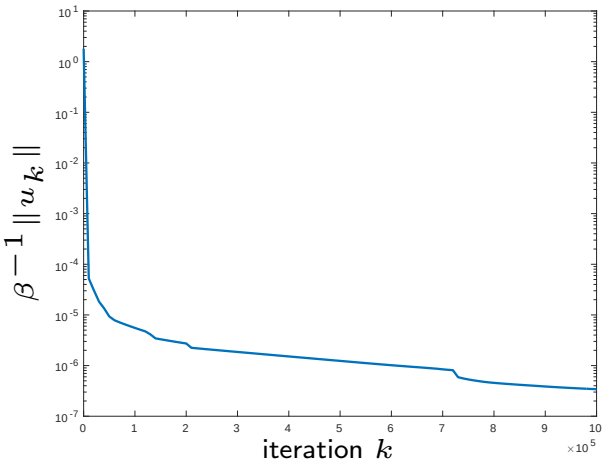


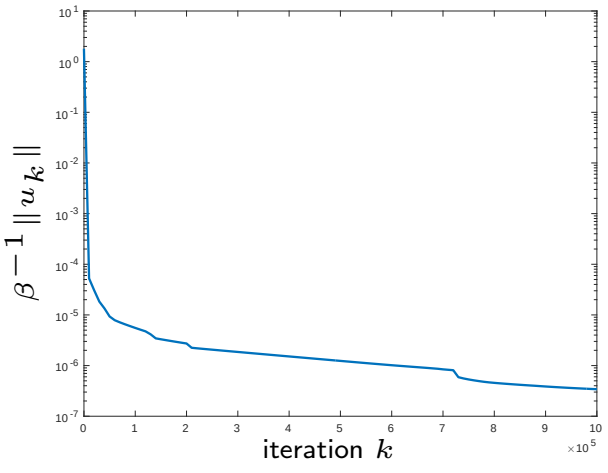


Early termination

- SVM polynomial features of degree 6, $\lambda = 0.00001$
- Iteration number: 1000000 Residual norm: $\beta^{-1}\|u_k\|_2 = 3.4e^{-7}$







Rates and Acceleration

Convergence rates

- We have only talked about convergence, not convergence *rates*
- Rates indicate how fast (in iterations) algorithm reaches solution
- Typically divided into:
  - Sublinear rates
  - Linear rates (also called geometric rates)
  - Quadratic rates
- Sublinear rates slowest, quadratic rates fastest

Sublinear rates

Can be of the form

$$f(x_k) + g(x_k) - p^\star \leq \frac{D}{\psi(k)}$$
$$\|x_{k+1} - x_k\|_2^2 \leq \frac{D}{\psi(k)}$$
$$\min_{t=1,\ldots,k} \mathbb{E}[\|\nabla f(x_t)\|_2^2] \leq \frac{D}{\psi(k)}$$

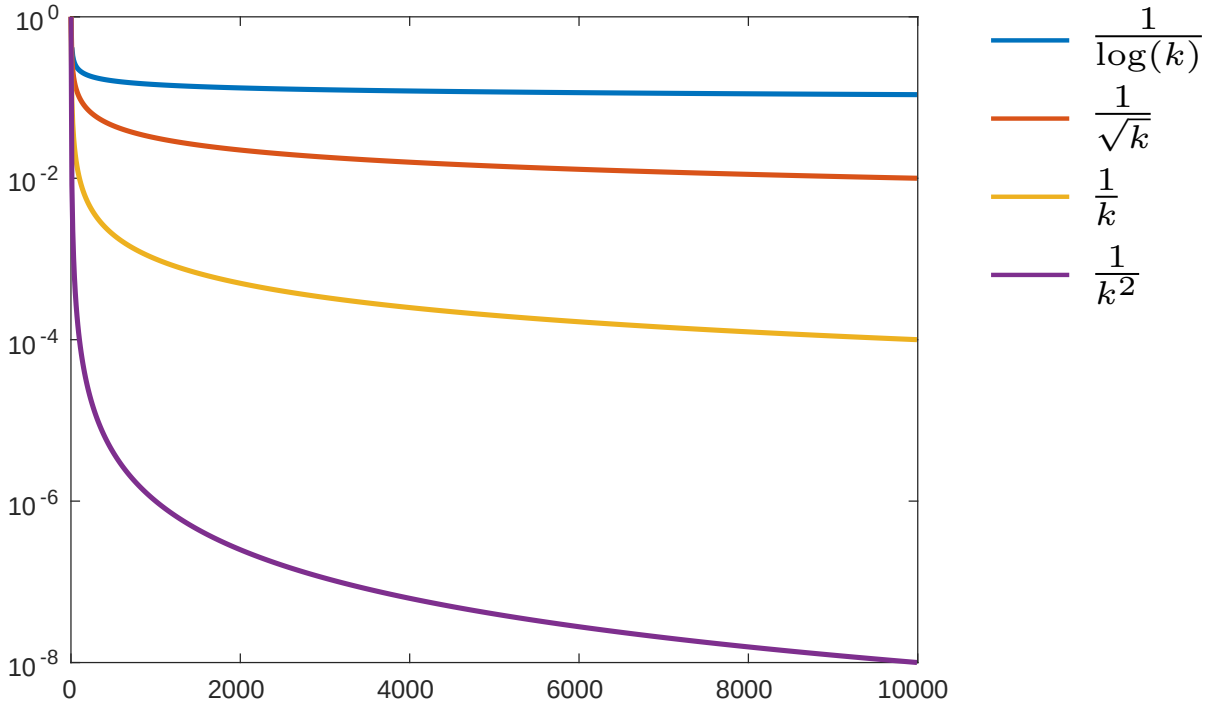
where ψ decides how fast it decreases, e.g.,

- $\psi(k) = \log(k)$
  - Nonconvex SGD, γ_k square summable not summable
- $\psi(k) = \sqrt{k}$
  - Convex SGD, convex subgradient method (not covered)
- $\psi(k) = k$
  - Proximal gradient method, coordinate proximal gradient descent
- $\psi(k) = k^2$
  - Accelerated proximal gradient method (convex, discussed later)

with improved convergence further down the list

Sublinear rates – Comparison

Different rates on lin-log scale



Linear rates

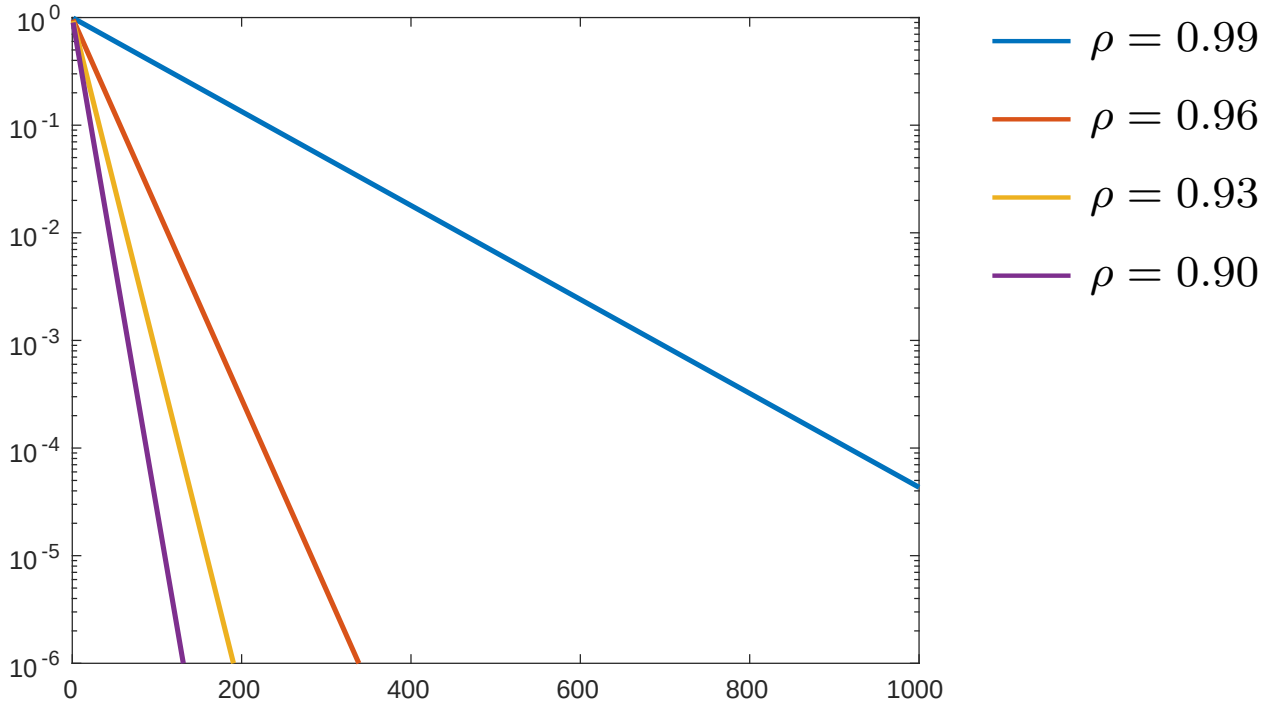
- Can be of the form

$$f(x_{k+1}) - f^\star \leq \rho_k (f(x_k) - f^\star)$$
$$\|x_{k+1} - x^\star\|_2 \leq \rho_k \|x_k - x^\star\|_2$$

  with $\rho_k \in [0, 1)$, i.e., a contraction, also called geometric rate
- Is called superlinear if $\rho_k \to 0$
- Examples:
  - (Accelerated) proximal gradient with strongly convex cost
  - Coordinate descent with strongly convex cost
  - BFGS has *local* superlinear with strongly convex cost
- However, SGD with strongly convex cost gives sublinear rate

Linear rates – Comparison

Different rates on lin-log scale



Quadratic rates

- Can be of the form

$$f(x_{k+1}) - f^\star \leq \rho_k (f(x_k) - f^\star)^2$$
$$\|x_{k+1} - x^\star\|_2 \leq \rho_k \|x_k - x^\star\|_2^2$$

  with $\rho_k \in [0, 1)$, very fast convergence when close to solution

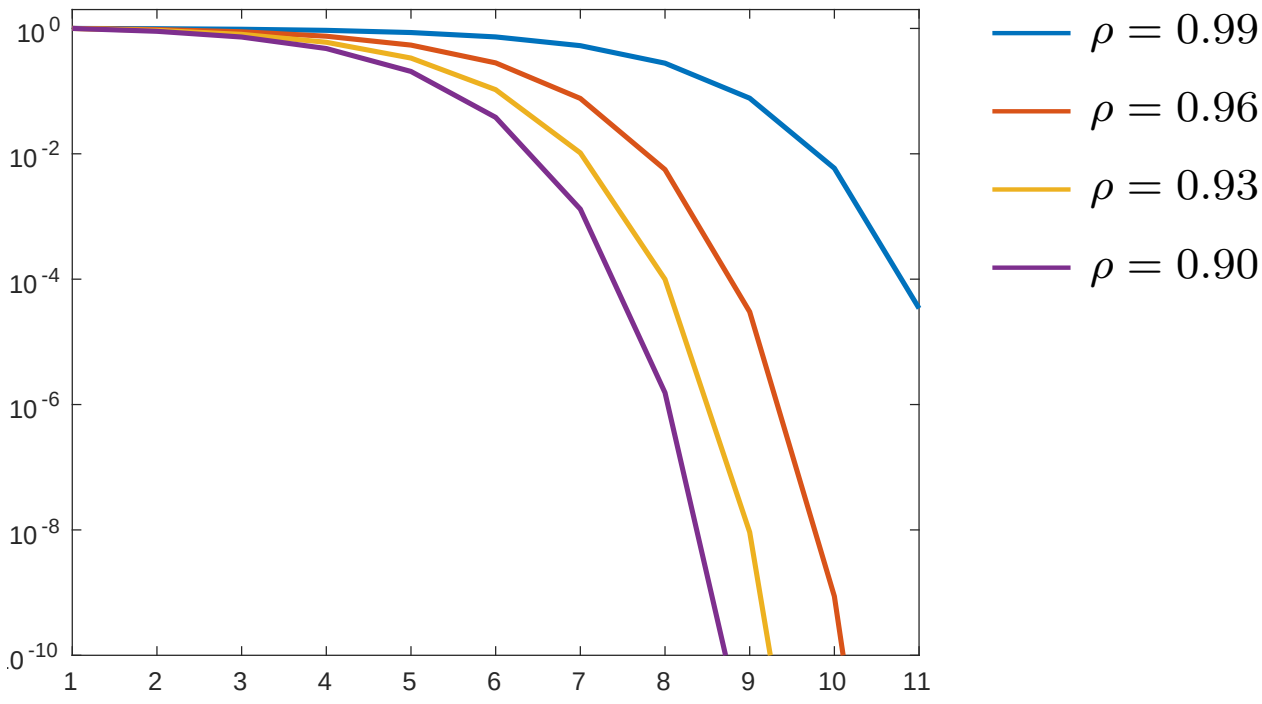
- Example with $\rho = 0.9$ compared with linear rate

Quadratic	Linear
1.000000000000	1.000000000000
0.900000000000	0.900000000000
0.729000000000	0.810000000000
0.478296799000	0.729000000000
0.205891068000	0.656099945000
0.038152029400	0.590490005000
0.001310019380	0.531440964000
0.000001544535	0.478296936000
0.000000000002	0.430467270000

- Example: *Locally* for Newton's method with strongly convex cost

Quadratic rates – Comparison

Different rates on lin-log scale



Accelerated proximal gradient method

- Consider convex composite problem

$$\underset{x}{\text{minimize}}\, f(x) + g(x)$$

  where
  - $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth and convex
  - $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is closed and convex
- Proximal gradient descent

$$x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k))$$

  achieves $O(1/k)$ convergence rate
- *Accelerated* proximal gradient method

$$y_k = x_k + \beta_k(x_k - x_{k-1})$$
$$x_{k+1} = \text{prox}_{\gamma g}(y_k - \gamma \nabla f(y_k))$$

  (with specific β_k) achieves faster $O(1/k^2)$ convergence rate

Accelerated proximal gradient method

- Stepsizes are restricted $\gamma \in (0, \frac{1}{\beta}]$
- The β_k parameters can be chosen either as

$$\beta_k = \tfrac{k-1}{k+2}$$

  or $\beta_k = \frac{t_{k-1}-1}{t_k}$ where

$$t_k = \tfrac{1+\sqrt{1+4t_{k-1}^2}}{2}$$

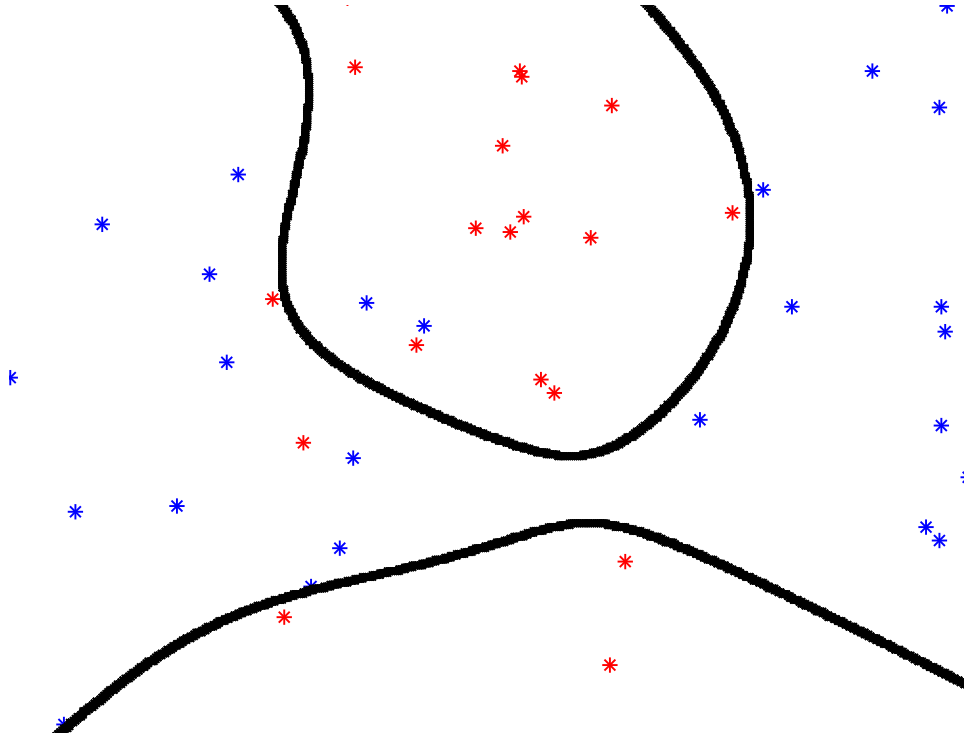
  these choices are very similar
- Algorithm behavior in nonconvex setting not well understood

Not descent method

- Descent method means function value is decreasing every iteration
- We have proven that proximal gradient is a descent method
- However, accelerated proximal gradient method is not!

Accelerated gradient method – Example

- Accelerated vs nominal proximal gradient method
- Problem from SVM lecture, polynomial deg 6 and $\lambda = 0.0215$



Accelerated gradient method – Example

- Accelerated vs nominal proximal gradient method
- Problem from SVM lecture, polynomial deg 6 and $\lambda = 0.0215$