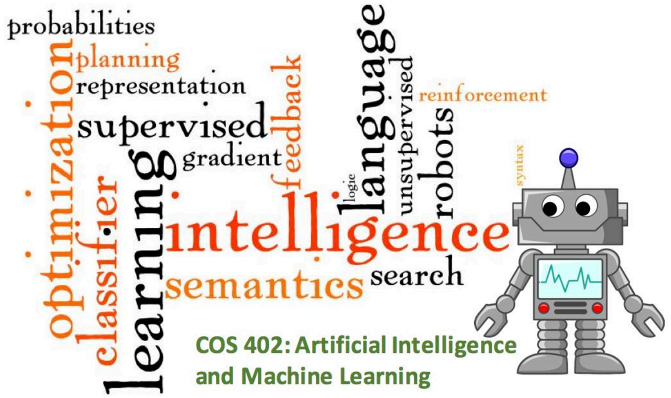


COS 402 – Machine Learning and Artificial Intelligence
Fall 2016

Lecture 5: optimization and convexity

Sanjeev Arora   Elad Hazan

PRINCETON UNIVERSITY

Admin

- Exercise 2 (implementation) next Thu, in class
- Exercise 3 (written), due next Thu
- Movie – “Ex Machina” + discussion panel w. Prof. Hasson (PNI)
  Wed Oct. 4th 19:30
  tickets still available @ Bella room 204 COS

- Next Tue: special guest - Dr. Yoram Singer @ Google

Recap

- Definition + fundamental theorem of statistical learning
- Powerful classes w. low sample complexity error exist (i.e. python programs), but computationally hard
- Perceptron
- SVM

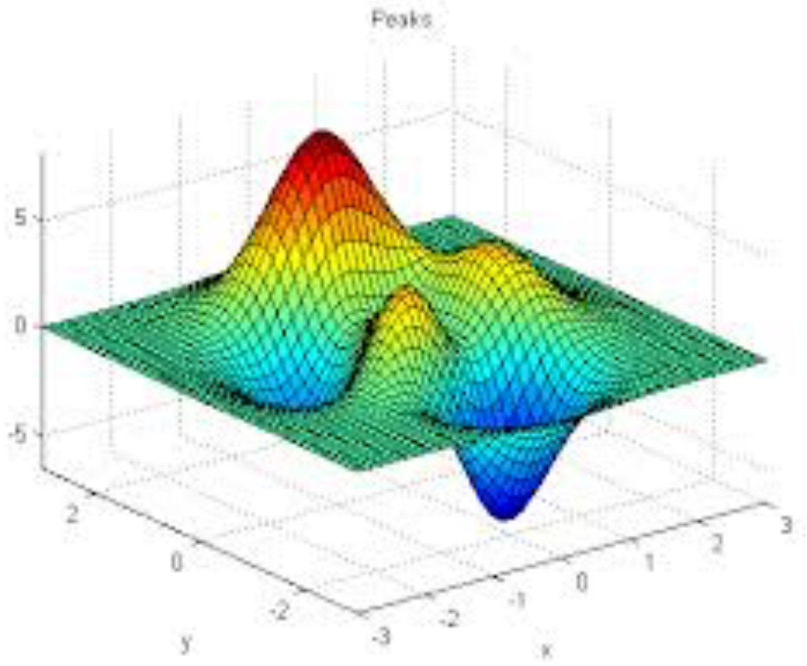
Agenda

- convex relaxations
- convex optimization
- Gradient descent

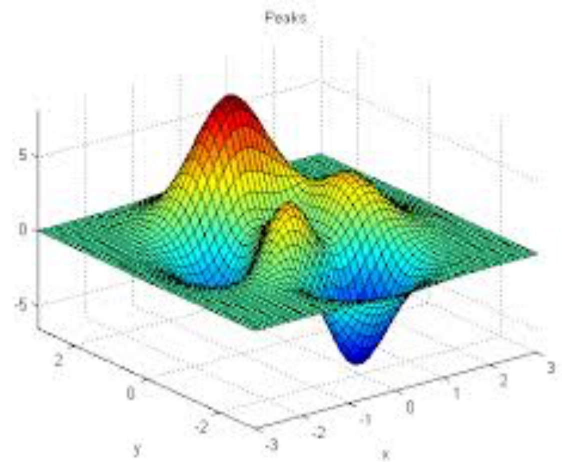
Mathematical optimization

Input: function $f: K \mapsto R$, for $K \subseteq R^d$

Output: point $x \in K$, such that $f(x) \leq f(y) \; \forall \; y \in K$



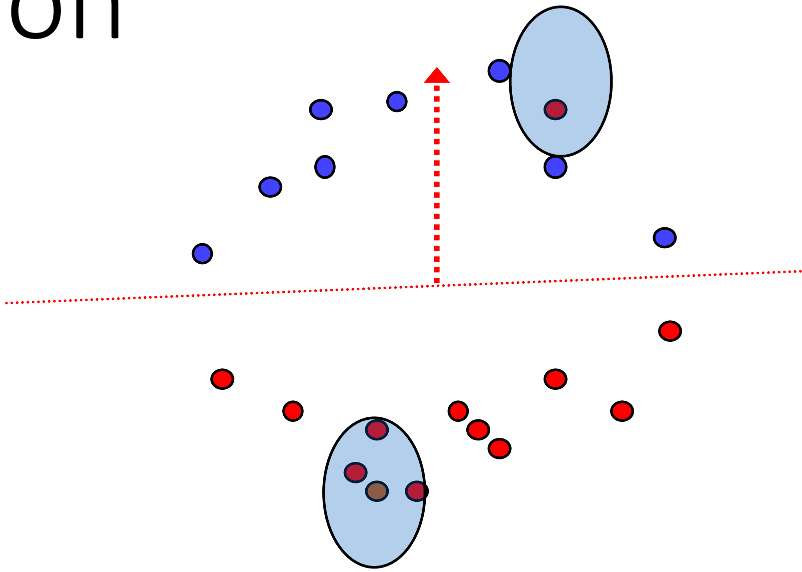
Mathematical optimization



- Continuous functions (back to calculus, derivatives, differentiability, …)
- Vs. combinatorial optimization as in graph algorithms (strong connection)
- Studied since early 1900's , lots of work in soviet union (central optimization, resource allocation, military applications, etc.)
- Special cases: linear programming, convex optimization, max flow in graphs

Efficient (poly-time) algorithms

Optimization for linear classification

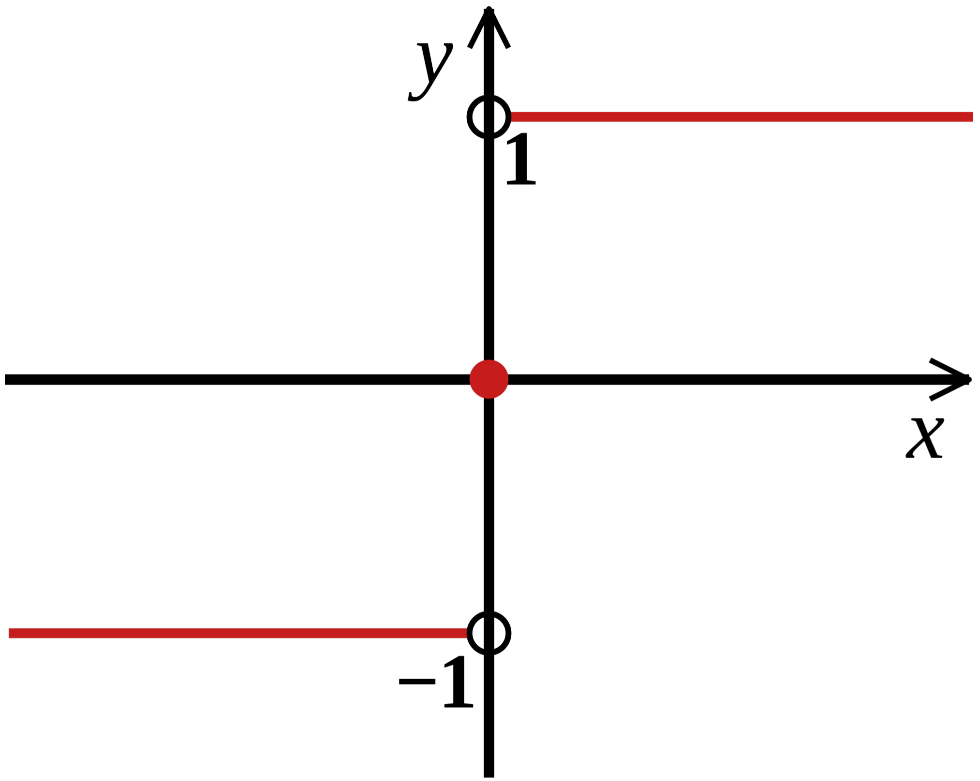


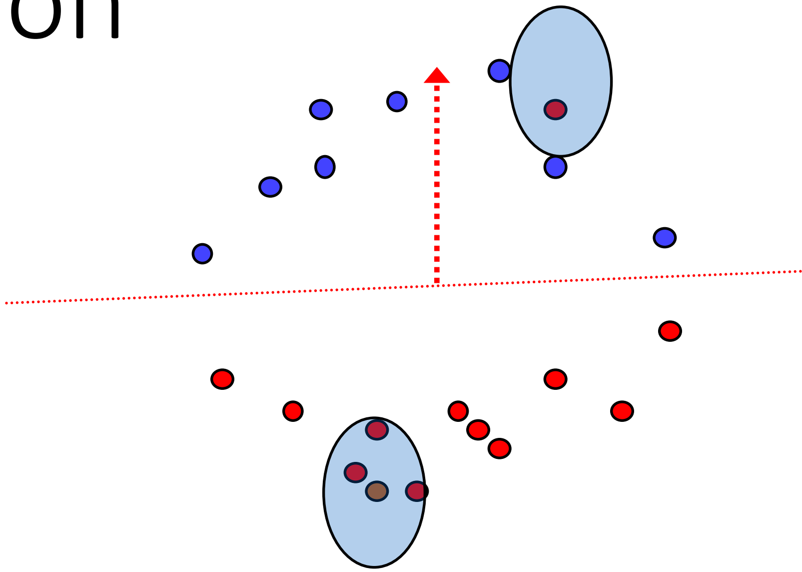
Given a sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, find hyperplane (through the origin w.l.o.g) such that:

$$w = \arg\min_{|w| \leq 1} \# \text{ of mistakes}$$

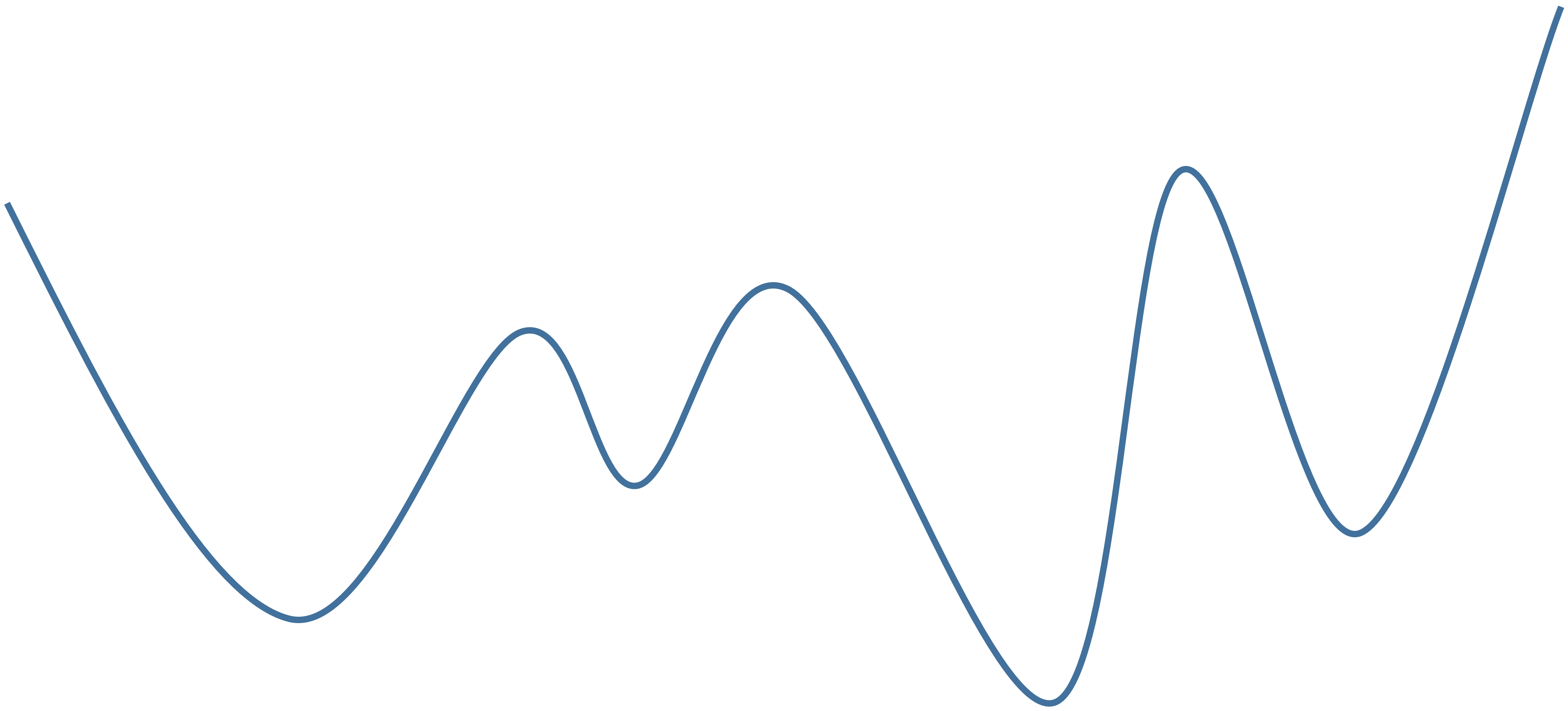
Optimization for linear classification

$$w = \arg\min_{|w|\leq 1} |\{i \;\; \text{s.t.}\; sign\,(w^T x_i) \neq y_i\}|$$

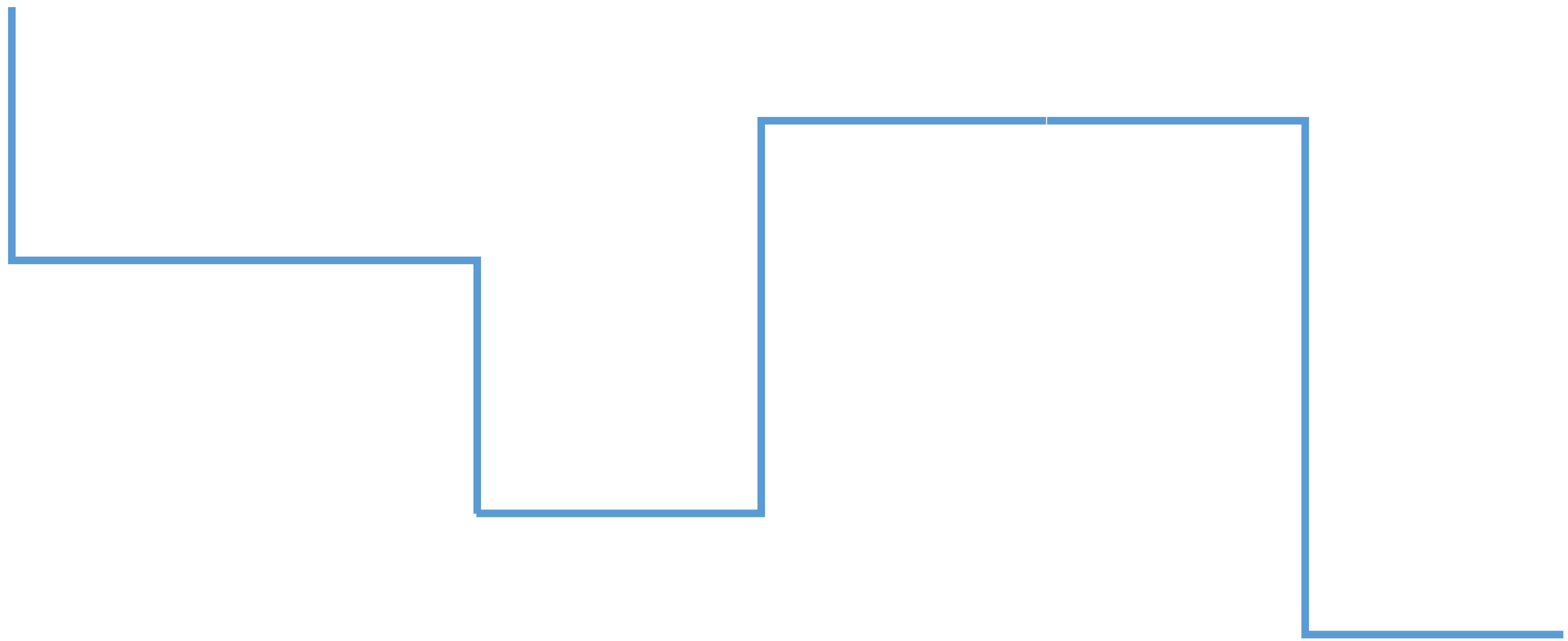




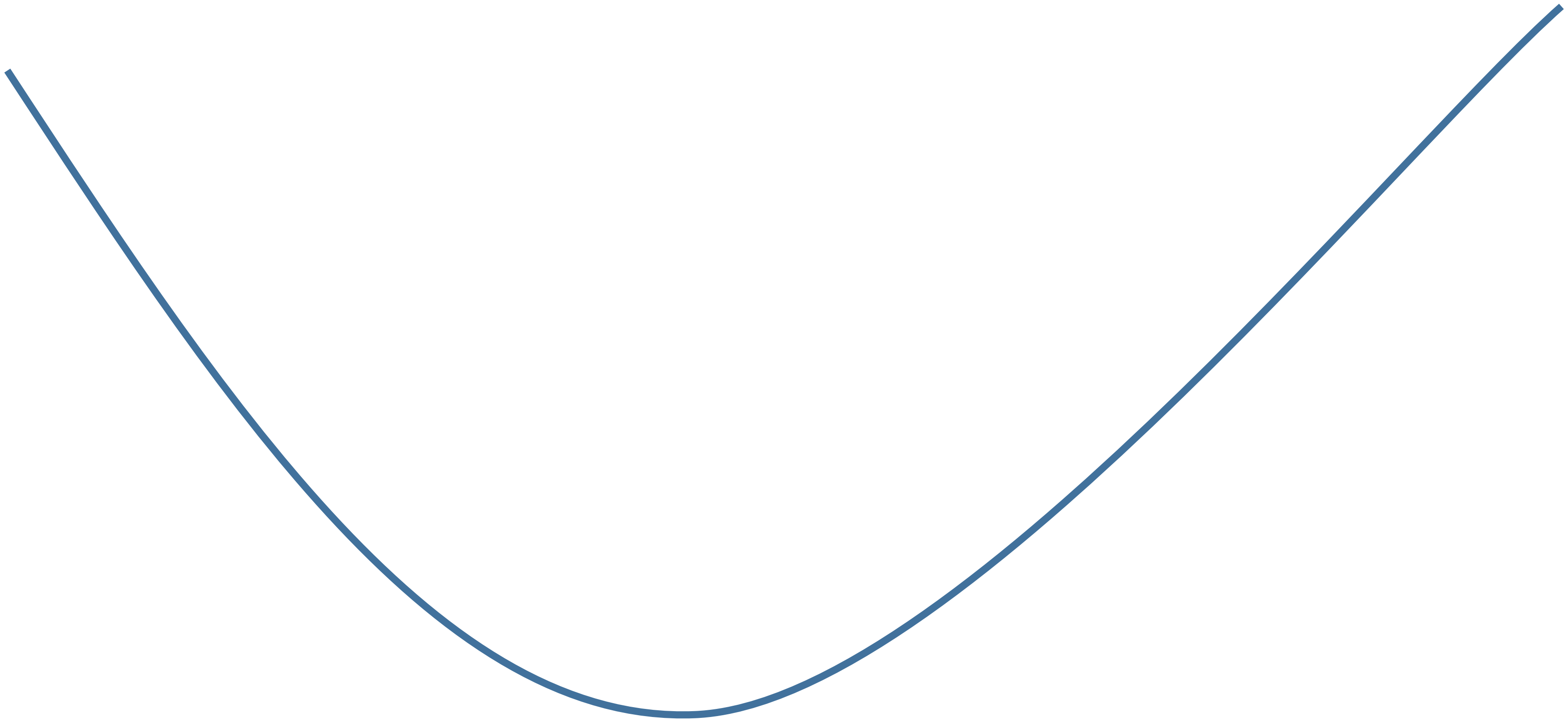
Minimization can be hard



Sum of signs → hard

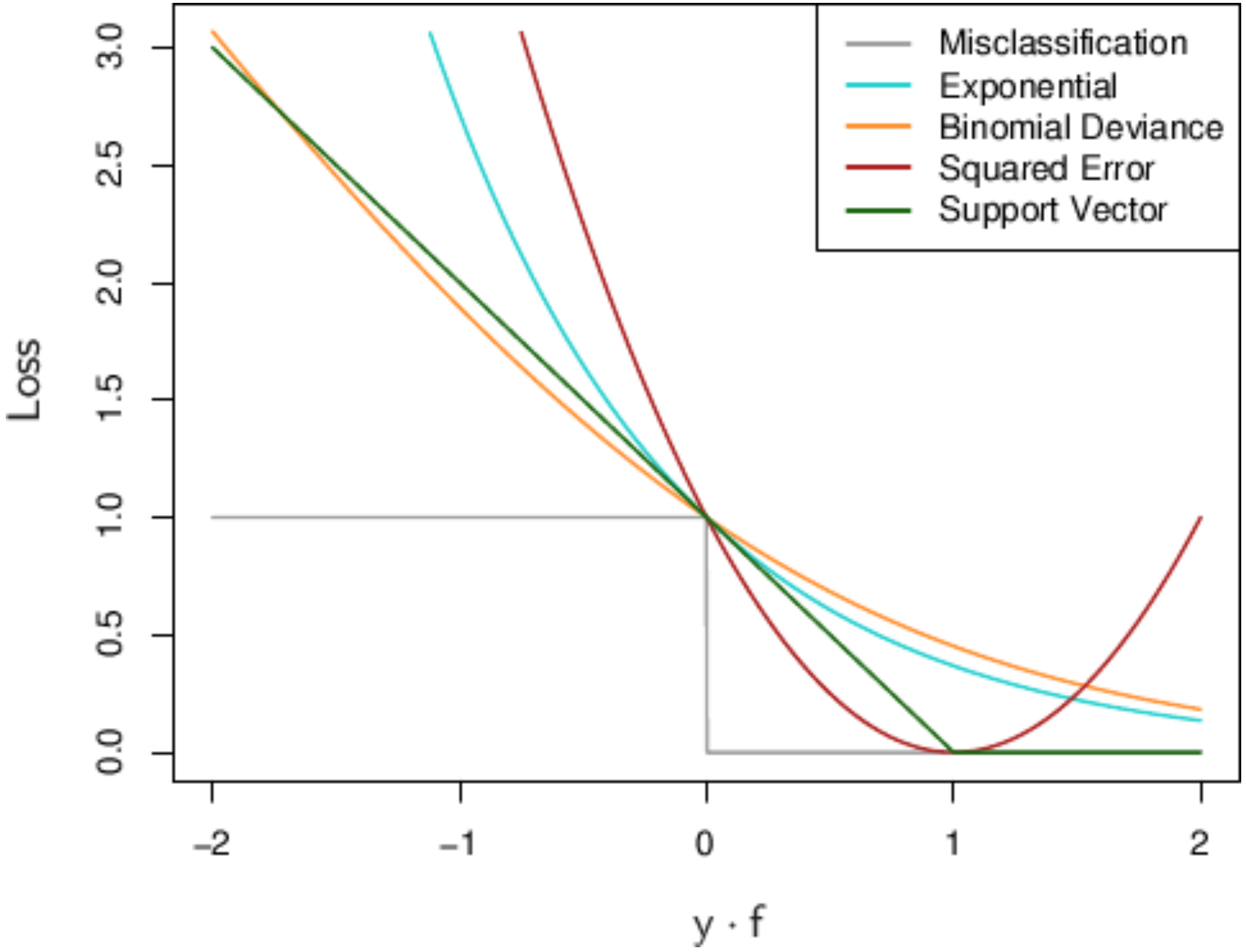


Convex functions: local → global

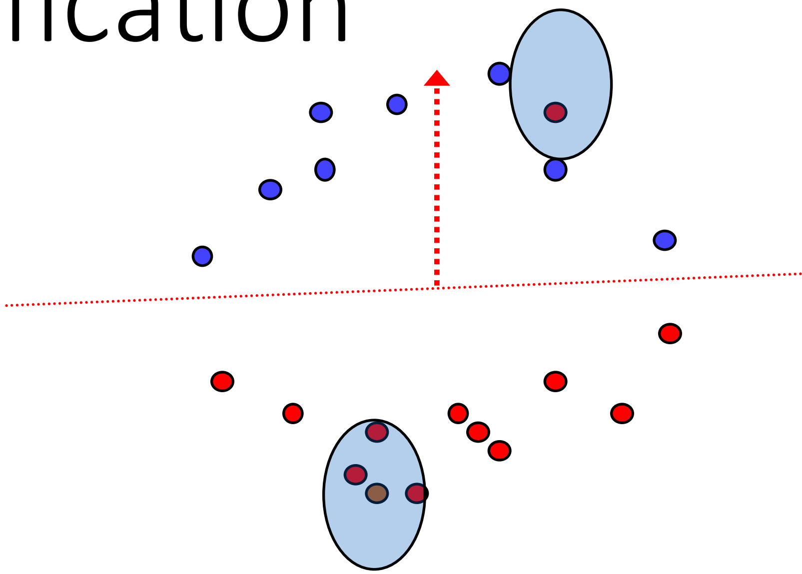


Sum of convex functions → also convex

Convex relaxation for 0-1 loss



Convex relaxation for linear classification



$$w = \arg\min_{|w| \le 1} |\{i \;\; s.t. \; sign\,(w^T x_i) \neq y_i\}|$$

$$w = \arg\min_{|w| \le 1} \ell(w^\top x_i, y_i)$$ such as:

1. Ridge / linear regression $\ell(w^\top x_i, y_i) = (w^\top x_i - y_i)^2$
2. SVM $\ell(w^\top x_i, y_i) = \max\{0, 1 - y_i \; w^\top x_i\}$
3. Logistic regression $\ell(w^\top x_i, y_i) = \log(1 + e^{w^\top x_i})$

Small recap

- Finding linear classifiers: formulated as mathematical optimization
- Convexity: property that allows local greedy algorithms
- Formulate convex relaxations to linear classification

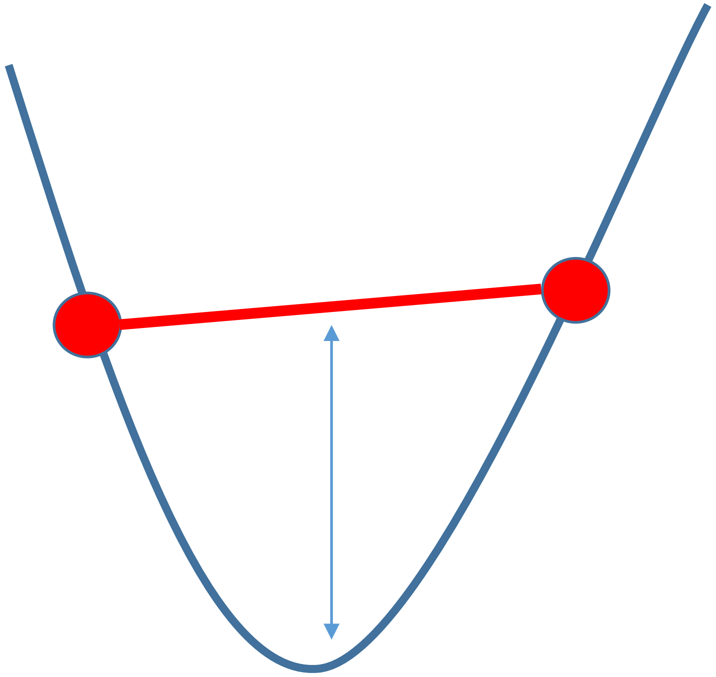
Next:

- Algorithms for convex optimization

Convexity

A function $f: R^d \mapsto R$ is convex if and only if:

$$f\left(\frac{1}{2}x + \frac{1}{2}y\right) \leq \frac{1}{2}f(x) + \frac{1}{2}f(y)$$

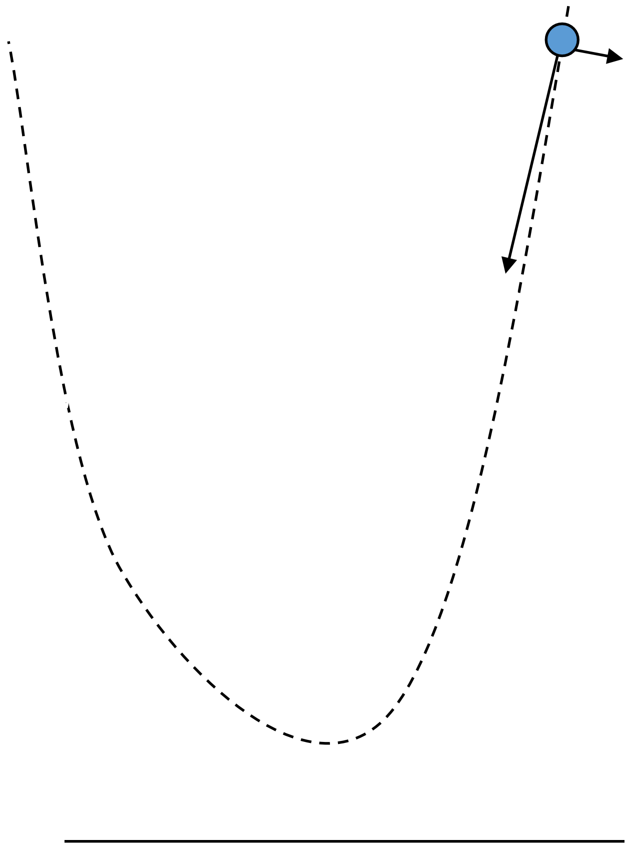


- Informally: smiley ☺

Calculus reminder: gradient

- Gradient = the direction of steepest descent, which is the derivative in each coordinate:

$$-[\nabla f(x)]_i = -\frac{\partial}{\partial x_i} f(x)$$

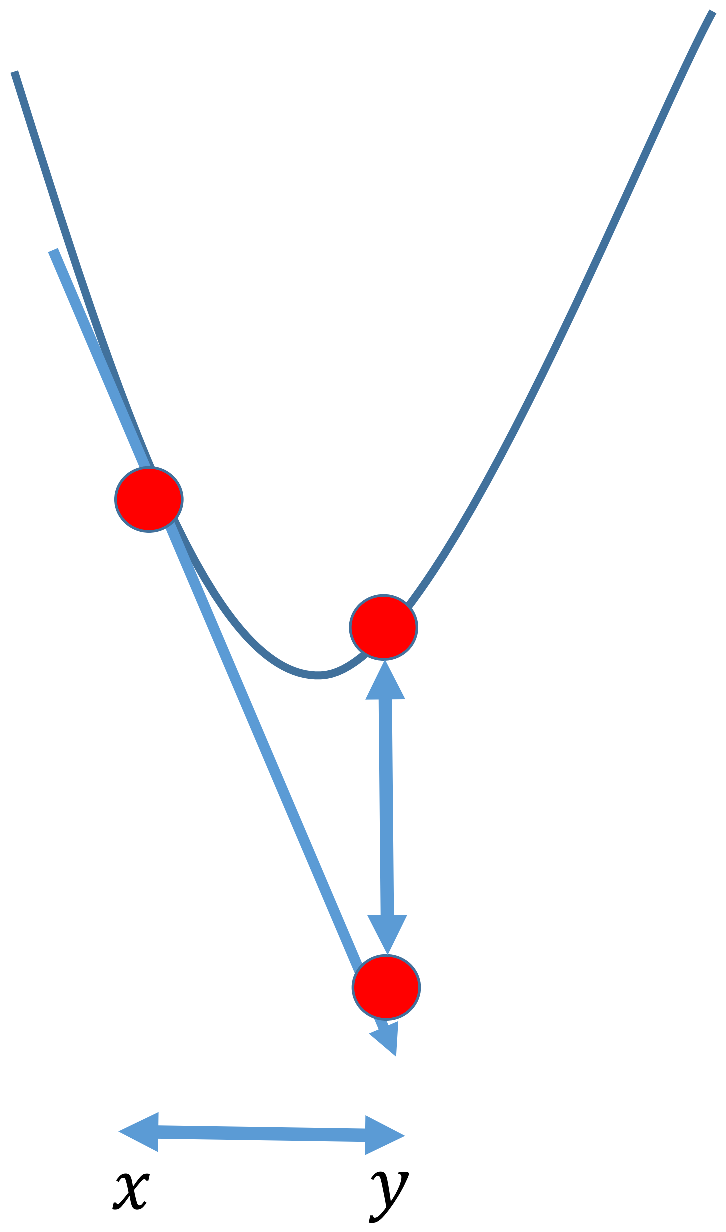


Convexity

- Alternative definition:

$$f(y) \geq f(x) + \nabla f(x)^{\top}(y - x)$$

(assumes differentiability, o/w subgradient)
(another alternative: second derivative is non-negative in 1D)



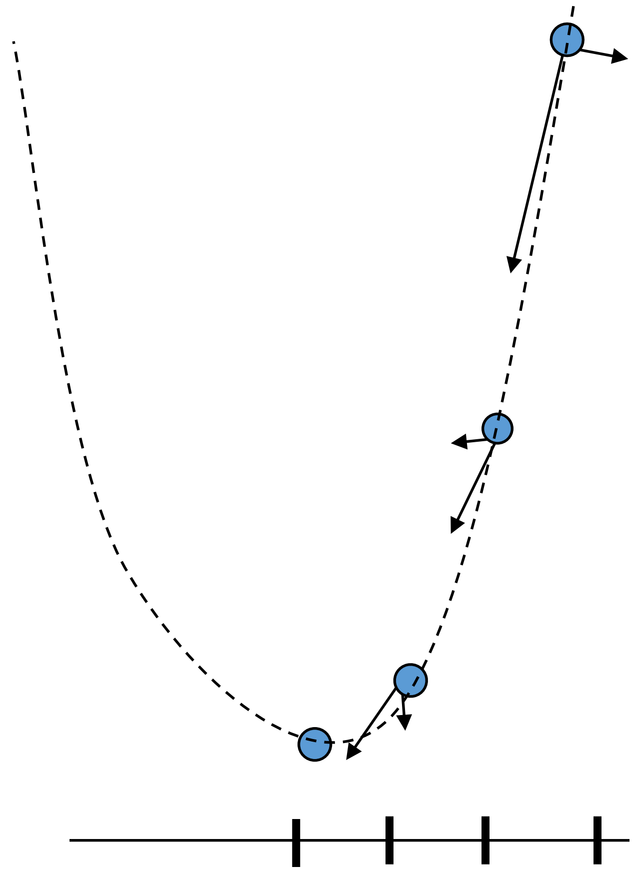
Greedy optimization: gradient descent

- Move in the direction of steepest descent, which is:

$$-[\nabla f(x)]_i = -\frac{\partial}{\partial x_i} f(x)$$

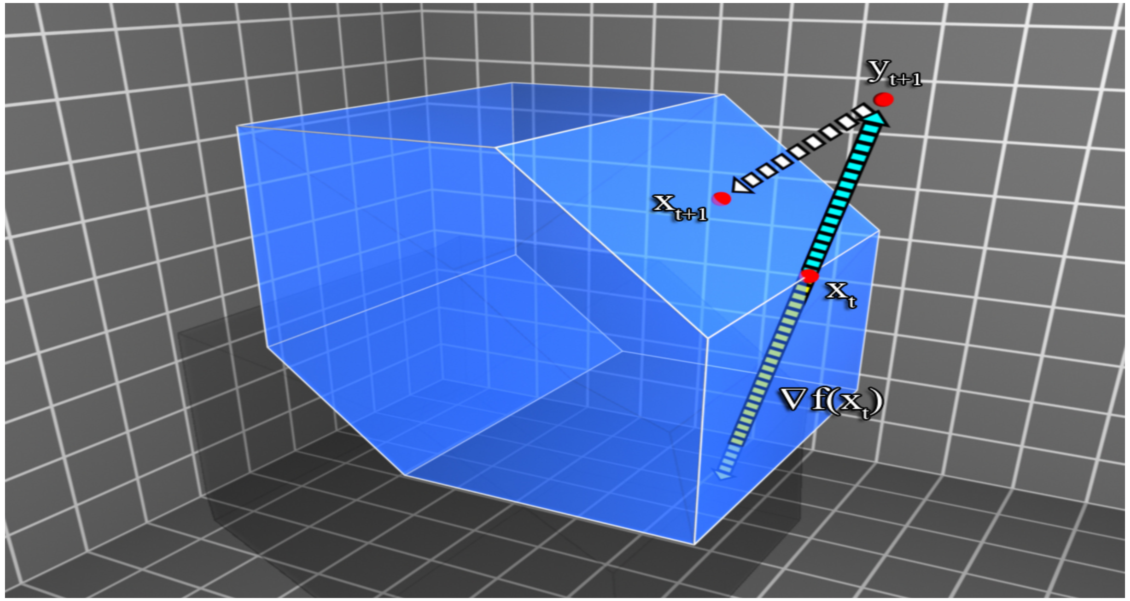
$$x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$$

"step size" or "Learning rate"



gradient descent – constrained set

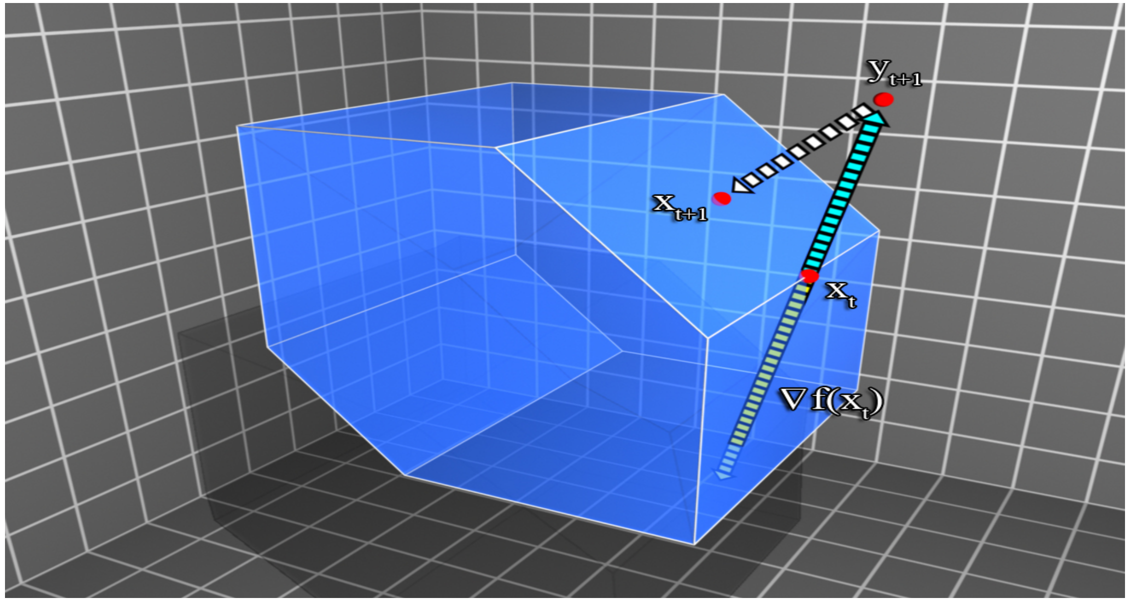
$$y_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$$
$$x_{t+1} = \arg\min_{x \in K} |y_{t+1} - x|$$



convex constraints

Set K is convex if and only if:

$$x, y \in K \;\; \Rightarrow (½x + ½y) \in K$$



gradient descent – constrained set

Let:

- G = upper bound on norm of gradients

$$|\nabla f(x_t)| \leq G$$

$$y_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$$
$$x_{t+1} = \arg\min_{x \in K} |y_{t+1} - x|$$

- D = diameter of constraint set

$$\forall x, y \in K \quad . \quad |x - y| \leq D$$

Theorem: for step size $\eta = \frac{D}{G\sqrt{T}}$

$$f\left(\frac{1}{T}\sum_t x_t\right) \leq \min_{x^* \in K} f(x^*) + \frac{DG}{\sqrt{T}}$$

Proof:

$$y_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$$
$$x_{t+1} = \arg\min_{x \in K} |y_{t+1} - x|$$

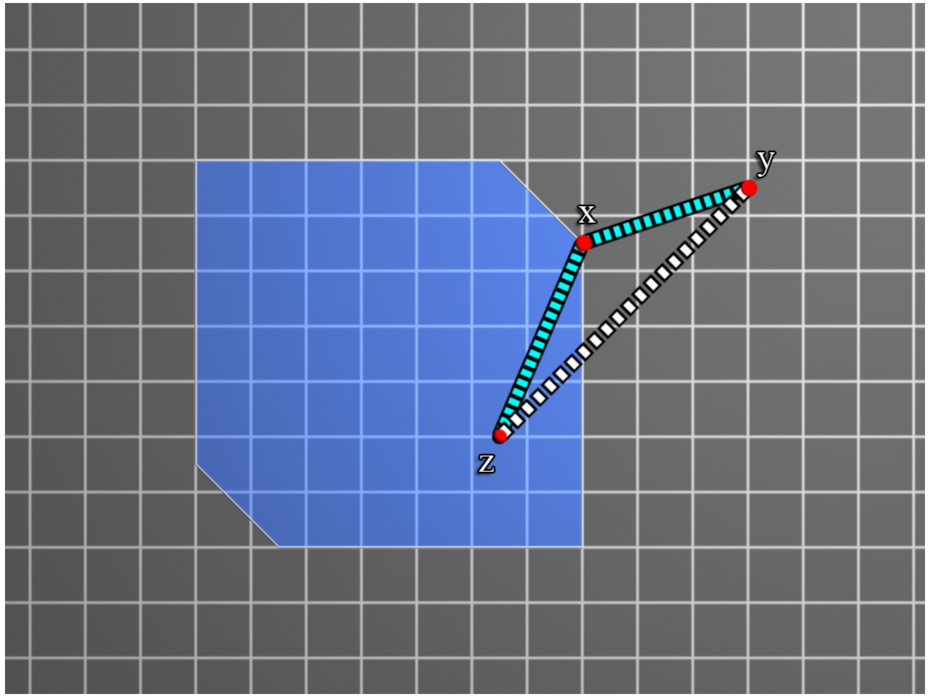
1. Observation 1:

$$|x^* - y_{t+1}|^2 = |x^* - x_t|^2 - 2\eta \nabla f(x_t)(x_t - x^*) + |\nabla f(x_t)|^2$$

2. Observation 2:

$$|x^* - x_{t+1}|^2 \le |x^* - y_{t+1}|^2$$

This is the Pythagorean theorem:



Proof:

$$y_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$$
$$x_{t+1} = \arg\min_{x \in K} |y_{t+1} - x|$$

1. Observation 1:

$$|x^* - y_{t+1}|^2 = |x^* - x_t|^2 - 2\eta \nabla f(x_t)(x_t - x^*) + |\nabla f(x_t)|^2$$

2. Observation 2:

$$|x^* - x_{t+1}|^2 \leq |x^* - y_{t+1}|^2$$

Thus:

$$|x^* - x_{t+1}|^2 \leq |x^* - x_t|^2 - 2\eta \nabla f(x_t)(x_t - x^*) + G^2$$

And hence:

$$f\left(\frac{1}{T}\sum_t x_t\right) - f(x^*) \leq \frac{1}{T}\sum_t [f(x_t) - f(x^*)] \leq \frac{1}{T}\sum_t \nabla f(x_t)(x_t - x^*)$$

$$\leq \frac{1}{T}\sum_t \frac{1}{2\eta}(|x^* - x_{t+1}|^2 - |x^* - x_t|^2) + \frac{\eta}{2} G^2$$

$$\leq \frac{1}{T \cdot 2\eta} D^2 + \frac{\eta}{2} G^2 \leq \frac{DG}{\sqrt{T}}$$

gradient descent – constrained set

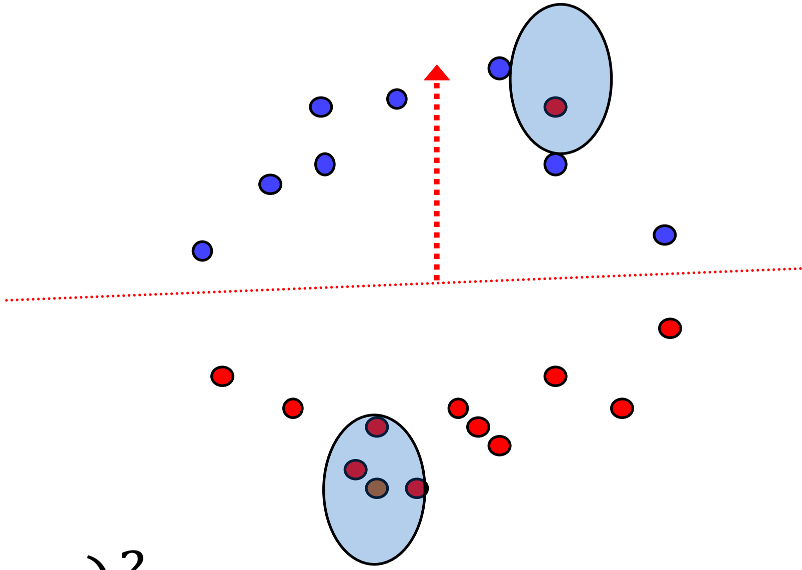
Theorem: for step size $\eta = \frac{D}{G\sqrt{T}}$

$$f\left(\frac{1}{T}\sum_t x_t\right) \leq \min_{x^* \in K} f(x^*) + \frac{DG}{\sqrt{T}}$$

Thus, to get ϵ-approximate solution, apply $\frac{\mathrm{D}^2 G^2}{\epsilon^2}$ gradient iterations.

GD for linear classification

$$w = \arg\min_{|w|\leq 1} \frac{1}{m} \sum_i \ell(w^\top x_i, y_i)$$

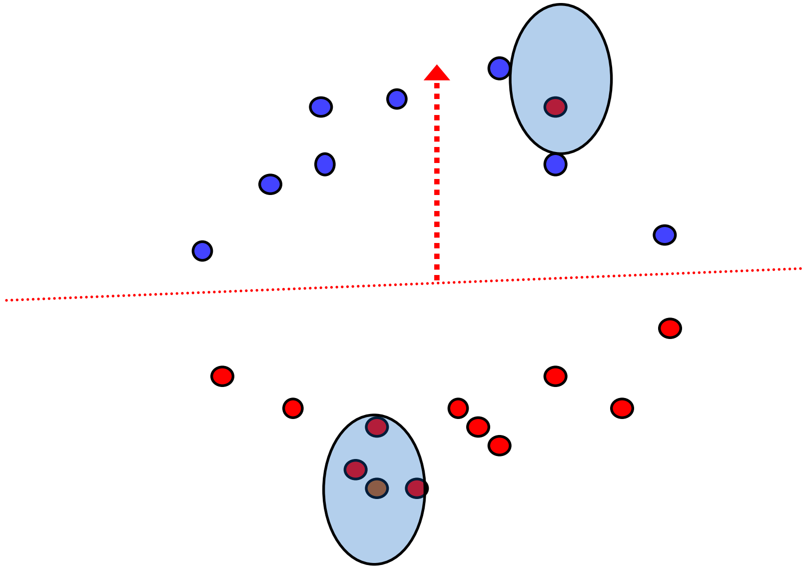


1. Ridge / linear regression $\ell(w^\top x_i, y_i) = (w^\top x_i - y_i)^2$
2. SVM $\ell(w^\top x_i, y_i) = \max\{0, 1 - y_i \; w^\top x_i\}$
3. Logistic regression $\ell(w^\top x_i, y_i) = \log(1 + e^{w^\top x_i})$

GD for linear classification

$$w = \arg\min_{|w|\leq 1} \frac{1}{m}\sum_i \ell(w^\top x_i, y_i)$$

$$w_{t+1} = \mathrm{w_t} - \eta \frac{1}{m}\sum_i \ell'(w_t^\top x_i, y_i) x_i$$



- Complexity? $\frac{1}{\epsilon^2}$ iterations, each taking ~ linear time in data set
- Overall $O\left(\frac{md}{\epsilon^2}\right)$ running time, m=# of examples in R^d
- Can we speed it up??

Summary

- Mathematical optimization for linear classification
- Convex relaxations
- Gradient descent algorithm
- GD applied to linear classification