Optimization Algorithms

Gradient Descent, Coordinate Descent

Dr. Elli Angelopoulou
Lehrstuhl für Mustererkennung (Informatik 5)
Friedrich-Alexander-Universität Erlangen-Nürnberg

Optimization Algorithms

- Solving optimization problems is a key component of pattern recognition.
- Many of the optimization problems are quite complex. Deriving an analytic solution is not trivial.
- An alternative is to use an algorithm to (iteratively) compute an (approximate) solution to the optimization problem.
- A widely used optimization algorithm is **gradient descent** (also known as steepest descent).
- A closely related algorithm for simultaneous solution of multiple parameters is **coordinate descent**.

Main Idea of Gradient Descent

- In order to find a local minimum of a function one can take steps proportional to the *negative of the gradient* of the function at the current point.
- Given a real valued function $f(\vec{x}) \in R$, which is differentiable at a point $\vec{x}_j \in R^n$, then at point $\vec{x}_j$, the function $f(\vec{x})$ decreases the fastest in the direction of the negative gradient $-\nabla f\left(\vec{x}_j\right)$ at $\vec{x}_j$, where

$$-\nabla f\left(\vec{x}\right) = \left(\frac{\partial f(\vec{x})}{\partial x_1}, \frac{\partial f(\vec{x})}{\partial x_2}, \ldots, \frac{\partial f(\vec{x})}{\partial x_n}\right)$$

Gradient Descent

- Thus if one "takes a small step *s* " on $f(\vec{x})$ at point $\vec{x}_j$ in the direction of the negative gradient $-\nabla f\left(\vec{x}_j\right)$, (s)he moves closer to the local minimum of the function $f(\vec{x})$.

$$s = -\eta \nabla f\left(\vec{x}_j\right)$$

$$\vec{x}_{j+1} = \vec{x}_j - \eta \nabla f\left(\vec{x}_j\right)$$

- Hence, one can start with an initial guess $\vec{x}_0$ for a local minimum of a function and follow a sequence of such steps $\vec{x}_0, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_j, \vec{x}_{j+1}, \ldots$ to gradually reach the local minimum.

Illustration of Gradient Descent

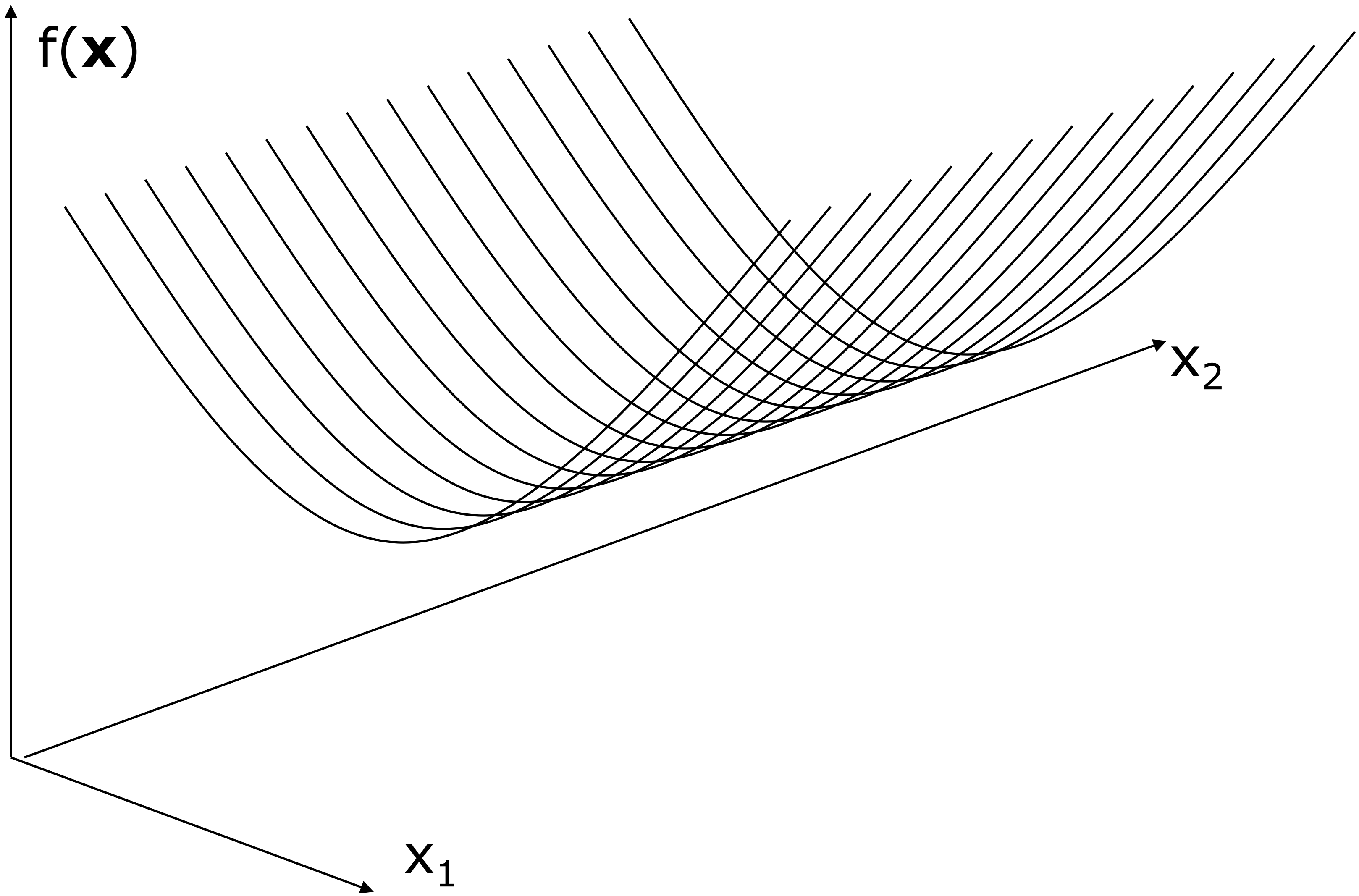


Illustration of Gradient Descent 2

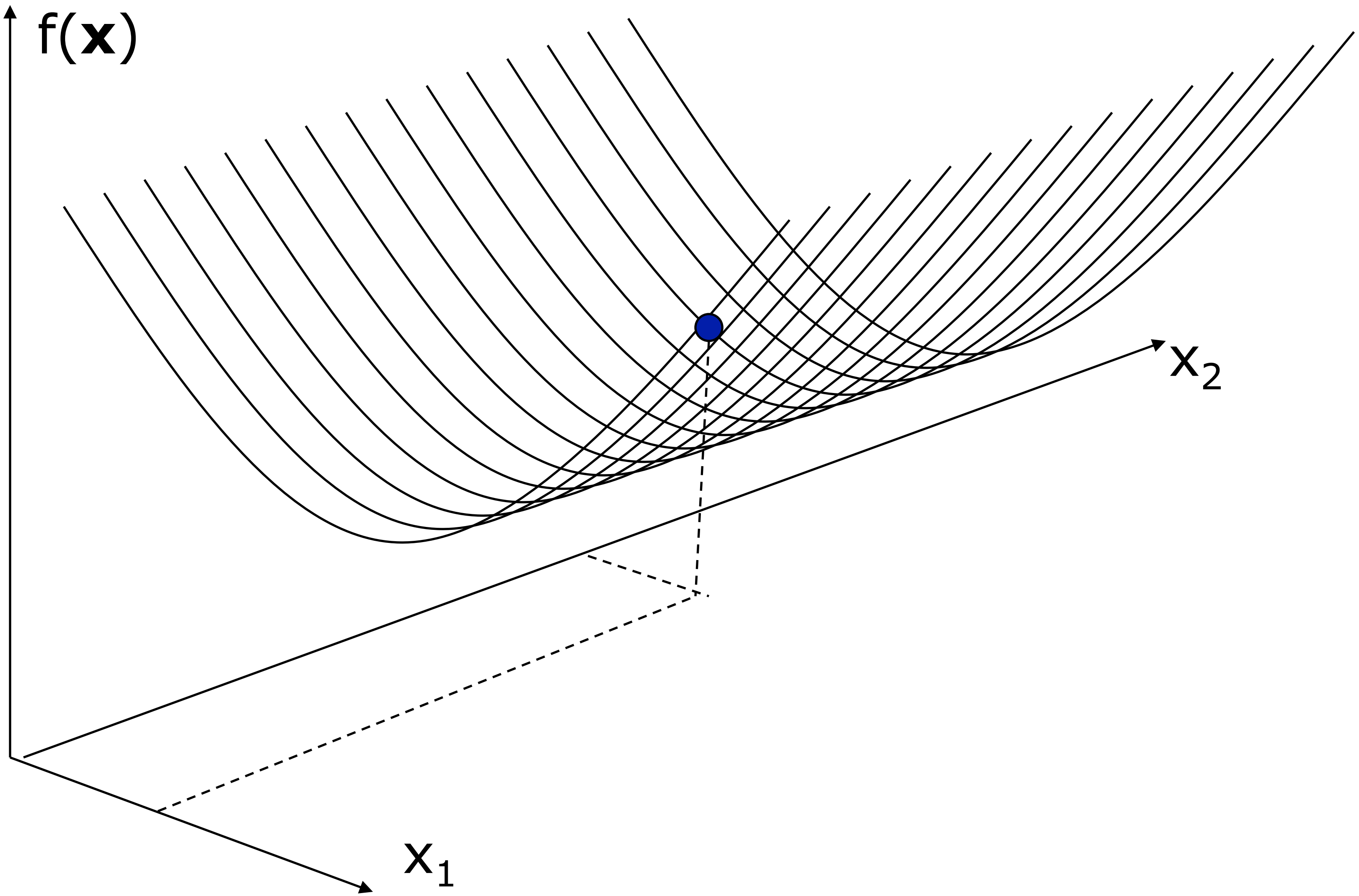
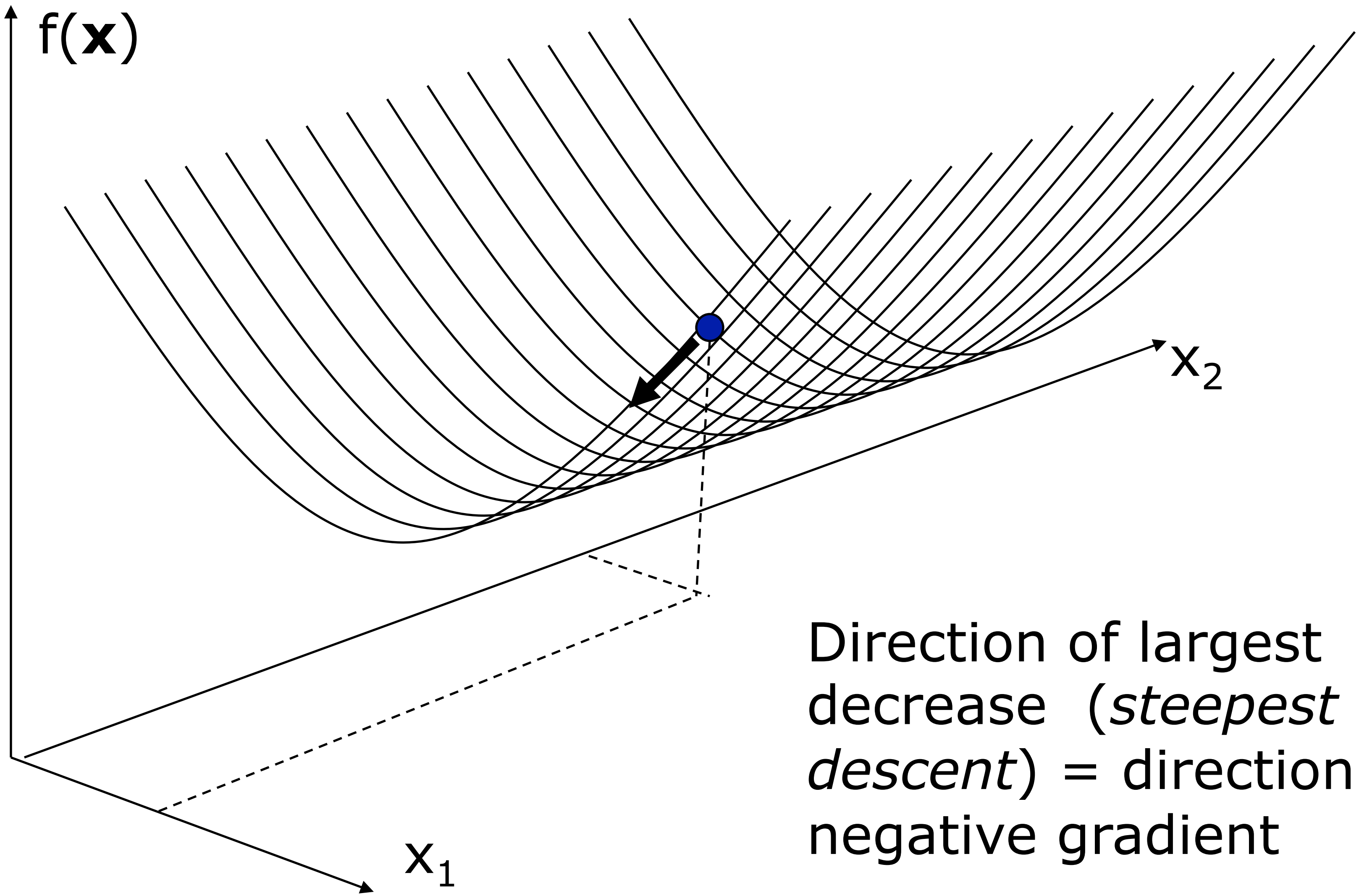
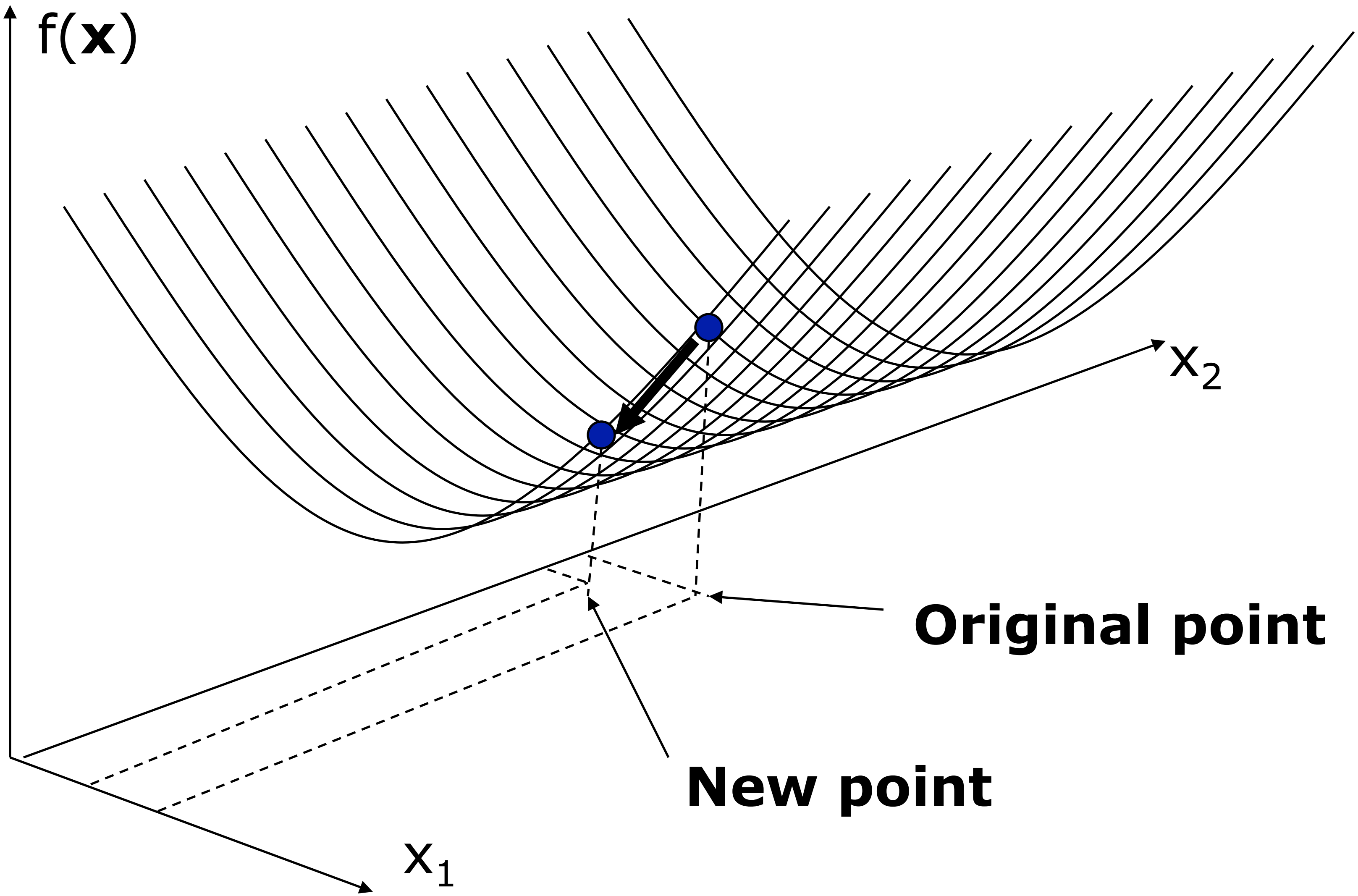


Illustration of Gradient Descent 3



Direction of largest decrease (*steepest descent*) = direction of negative gradient

Illustration of Gradient Descent 4



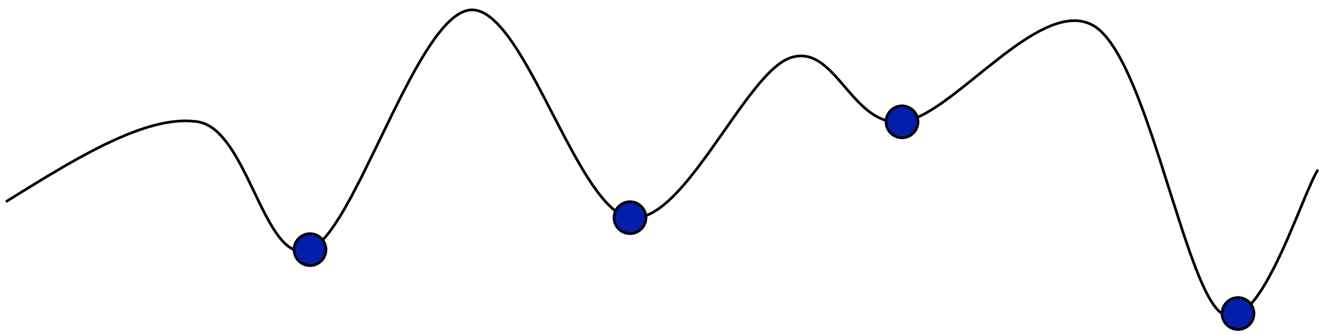
Gradient Descent Algorithm

k=0
Initialize x_k
while x_k is not a minimum
 compute gradient D_k at point x_k
 compute step s_k, $s_k = -\eta_k D_k$
 $x_{k+l} = x_k + s_k$
 k=k+1
end

- The size of the step depends on
  - The magnitude of the gradient
  - The value of the scalar η_k

Gradient Descent and Global Minimum

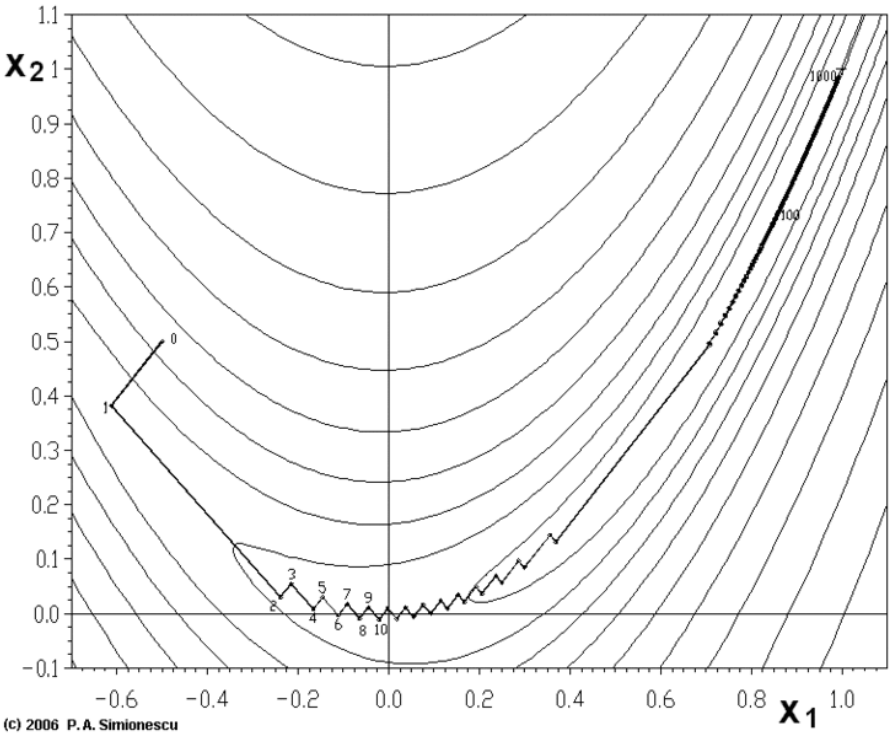
- Gradient descent converges to the closest local minimum.
- It computes the global minimum of a function only for unimodal functions.
- For functions with multiple minima, there is no guarantee that gradient descent will converge to the global minimum.
- A solution (still no guarantee): Run gradient descent multiple times starting from distinct initial points.



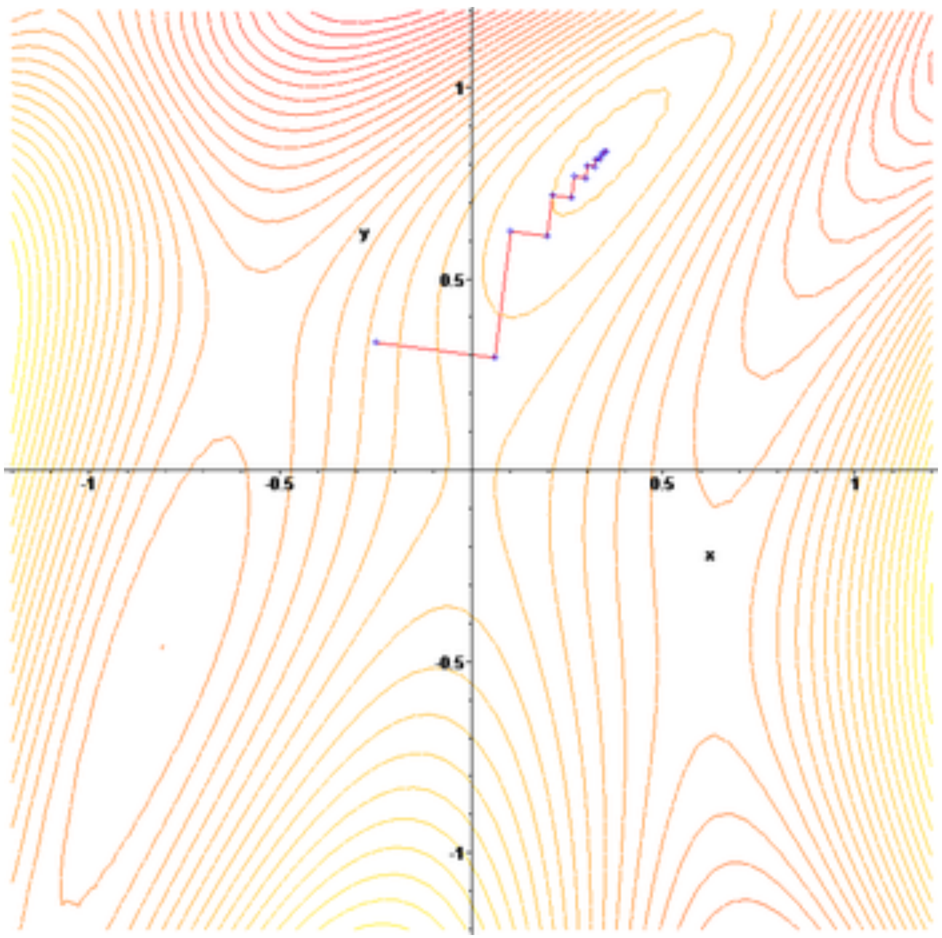
Remarks on Gradient Descent

- Picking an appropriate $\mathbf{x_o}$ is crucial, but also problem-dependent.
- The stopping criteria are not clearly defined.
- For solving maximization problems, one can simply step in the direction of the gradient $\nabla f\left(\vec{x}_j\right)$.
- A well-known problematic behavior of gradient descent is its “zig-zagging” track in functions with very flat local minima (maxima), that approximate plateaus.

Examples of Zig-Zagging Behavior



Plot of the Rosenbrock function, which has a very narrow and flat valley that contains the minimum. It takes many small steps, with localized zig-zagging behavior to eventually converge to the minimum.



Plots courtesy of Wikipedia, http://en.wikipedia.org/wiki/Gradient_descent

Coordinate Descent

- It is closely related to gradient descent.
- It is designed for optimization problems where multiple parameters of the same optimization function must be simultaneously searched for the optimal solution.

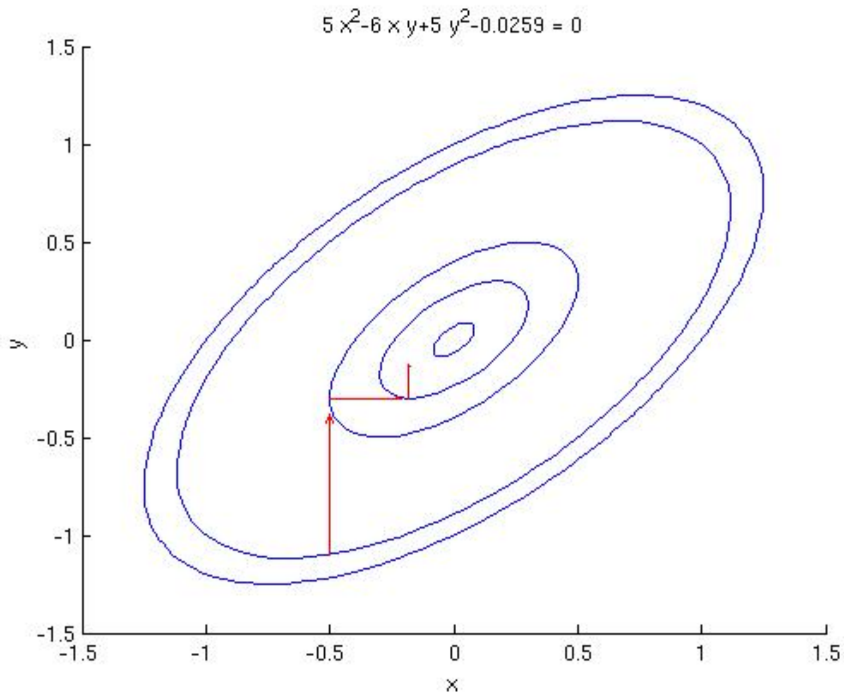
$$\hat{\vec{x}} = \arg\min_{x_1, x_2, \ldots x_n} f(\vec{x})$$

- Main idea: Apply gradient descent in one coordinate axis at a time. In other words, first search for x_1, then search for x_2, then for x_3 and so on. For example , during the (k+1)th iteration:

$$x_i^{k+1} = \arg\min_{y} f(x_1^{k+1}, x_2^{k+1}, \ldots, x_{i-1}^{k+1}, y, x_{i+1}^{k}, x_{i+2}^{k}, \ldots x_n^{k})$$

Coordinate Descent - continued

- In coordinate descent, unlike gradient descent, instead of descending along the direction of the gradient, one moves along a coordinate direction.
- In coordinate descent one cycles through the different coordinate directions.
- At each iteration one descents once through each coordinate direction.

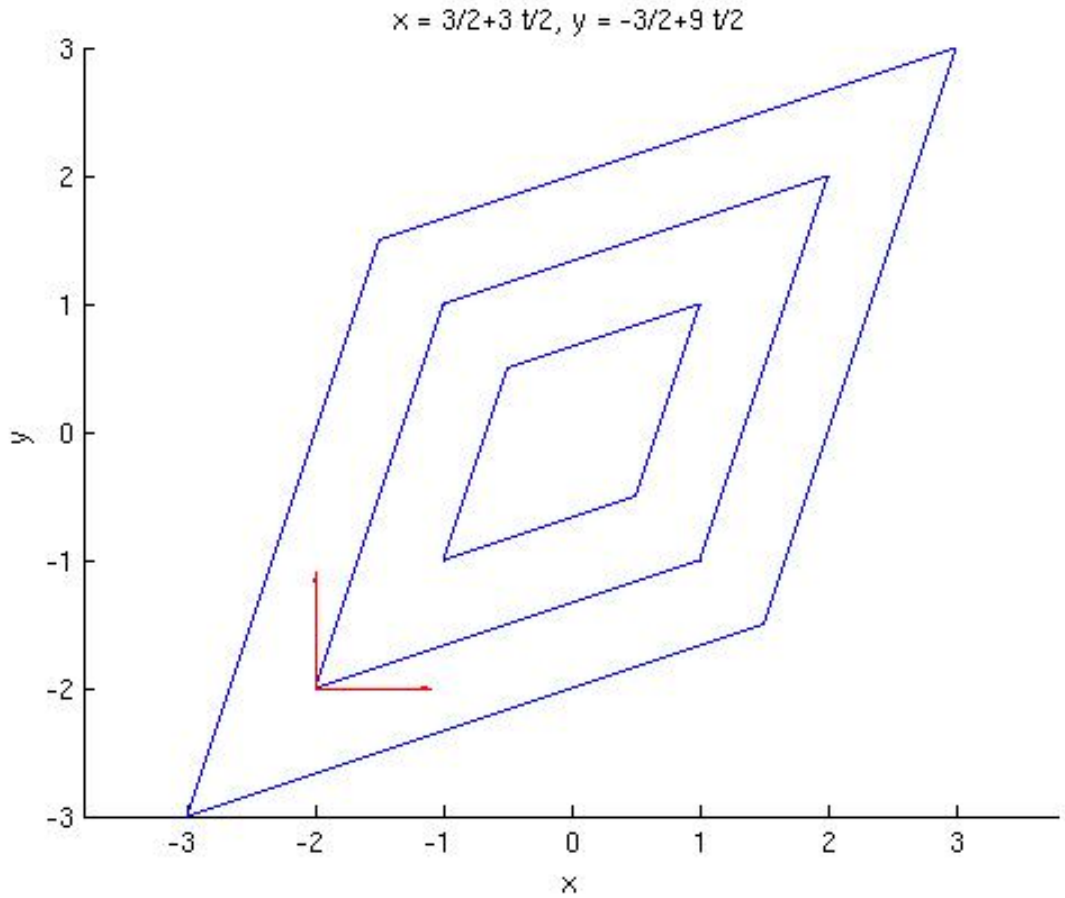


Plot courtesy of Wikipedia, http://en.wikipedia.org/wiki/Coordinate_descent

Coordinate Descent – continued 2

- Coordinate descent has similar convergence properties as gradient descent.
- It can also get stuck in local minima.
- However, it is easy to implement and sometimes faster to compute. No gradient computation.
- Drawback: No convergence proof.
- A well-known problem of coordinate descent is that it may stop descending for non-smooth functions.

Non-Smooth Functions and Coord. Descent



Plot courtesy of Wikipedia, http://en.wikipedia.org/wiki/Coordinate_descent

Resources

1. Some of the material on gradient descent is adapted from the slides by P. Smyth
   http://www.ics.uci.edu/~smyth/courses/cs175/slides5b_gradient_search.ppt