Escaping *Undesired* Stationary Points in Local Saddle Point Optimization
A Curvature Exploitation Approach

Leonard Adolphs Hadi Daneshmand Aurelien Lucchi Thomas Hofmann

Institute of Machine Learning, ETH Zürich, Switzerland

Saddle Point Problem

$$\min_{x \ R^k} \max_{y \ R^d} f(x, y)$$

Assumptions

- f smooth but non-convex (non-concave) in x (y)
- $^2_x f(x, y)$, $^2_y f(x, y)$ non-degenerate

Relaxed Objective

Finding a **global** saddle point of the above form is generally **infeasible**. Therefore, we aim for a solution in a local neighbourhood, i.e., a point (x , y) s.t.

$$f(x , y) \leq f(x , y) \leq f(x, y) \quad (x, y) \quad K_y$$

where K_y is a local neighbourhood around the saddle.

Local Saddle Point Conditions

1. $f(x , y) = 0$
2. $^2_x f(x , y) \quad 0$
3. $^2_y f(x , y) \quad 0$

Gradient-Based Optimization

Simultaneously applying Gradient Descent on x and Gradient Ascent on y:

$$\begin{bmatrix} x^+ \\ y^+ \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \eta \begin{bmatrix} - \ _x f(x, y) \\ _y f(x, y) \end{bmatrix}$$

If convergent, it almost surely finds a **stable** stationary point of the gradient dynamics. But not necessarily a solution to the local saddle point problem . . .

Gradient-Based Optimization Does not Solve for Local Saddles

Even if gradient-based optimization converges, we have no (approximate) guarantee of obtaining a solution to the local saddle point problem.

Stability versus Optimality

	local optimality condition	stability condition
Minimization	$^2_x f(x, y) \quad 0$ (=)	$^2_x f(x, y) \quad 0$
Saddle Point Optimization	$^2_x f(x, y) \quad 0$; $^2_y f(x, y) \quad 0$ (≠)	λ $\begin{bmatrix} - \ ^2_x f & - \ _{xy} f \\ _{yx} f & ^2_y f \end{bmatrix}$ negative real part

Table 1: Stability versus optimality condition in minimization and saddle point optimization.

$$\min_{x \ R} \max_{y \ R} f(x, y) = 2x^2 + y^2 + 4xy + \frac{4}{3}y^3 - \frac{1}{4}y^4$$

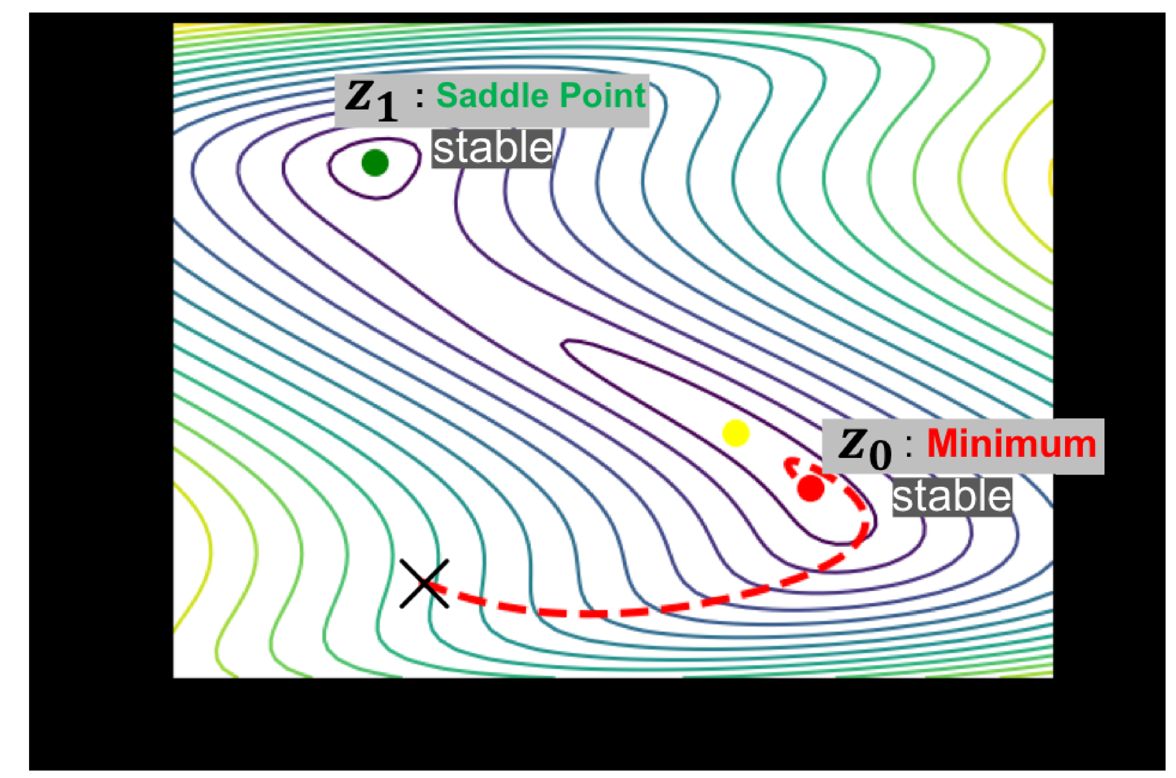


Figure 1: Gradient-based optimization converges to the minimum z_0 rather than the saddle point z_1.

CESP - Curvature Exploitation for the Saddle Point Problem

Intuition

Simple observation: If there is positive (negative) curvature in x-direction (y-direction) then the local saddle point conditions are not met, because $^2_x f(x, y) \quad 0$ ($\ ^2_y f(x, y) \quad 0$).

Following negative curvature in x and positive curvature in y helps us escape from undesired stable stationary points.

Algorithm

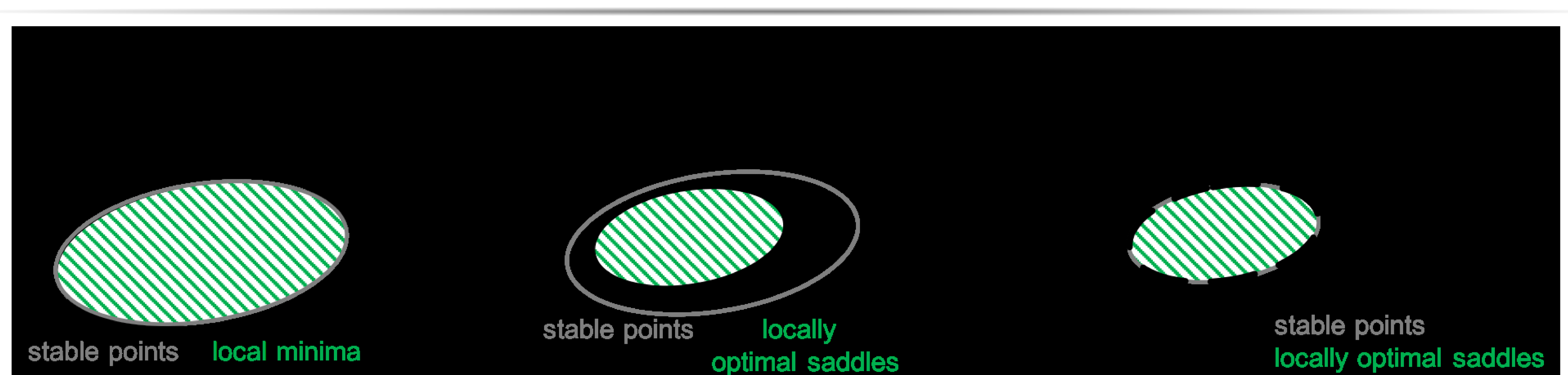
Let λ_x (λ_y) be the minimum (maximum) eigenvalue of $^2_x f$ ($^2_y f$) with its associated eigenvector v_x (v_y) and $\rho_x, \rho_y > 0$ some smoothness parameters.

$$v_z^{(-)} = 1_{\{\lambda_x<0\}} \frac{\lambda_x}{2\rho_x} \operatorname{sgn}(v_x \ \ _x f(z)) v_x$$

$$v_z^{(+)} = 1_{\{\lambda_y>0\}} \frac{\lambda_y}{2\rho_y} \operatorname{sgn}(v_y \ \ _y f(z)) v_y$$

$$\begin{bmatrix} x^+ \\ y^+ \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \eta \begin{bmatrix} - \ _x f(x, y) \\ _y f(x, y) \end{bmatrix} + \begin{bmatrix} v_z^{(-)} \\ v_z^{(+)} \end{bmatrix}$$

Theoretical Guarantees



Toy Example

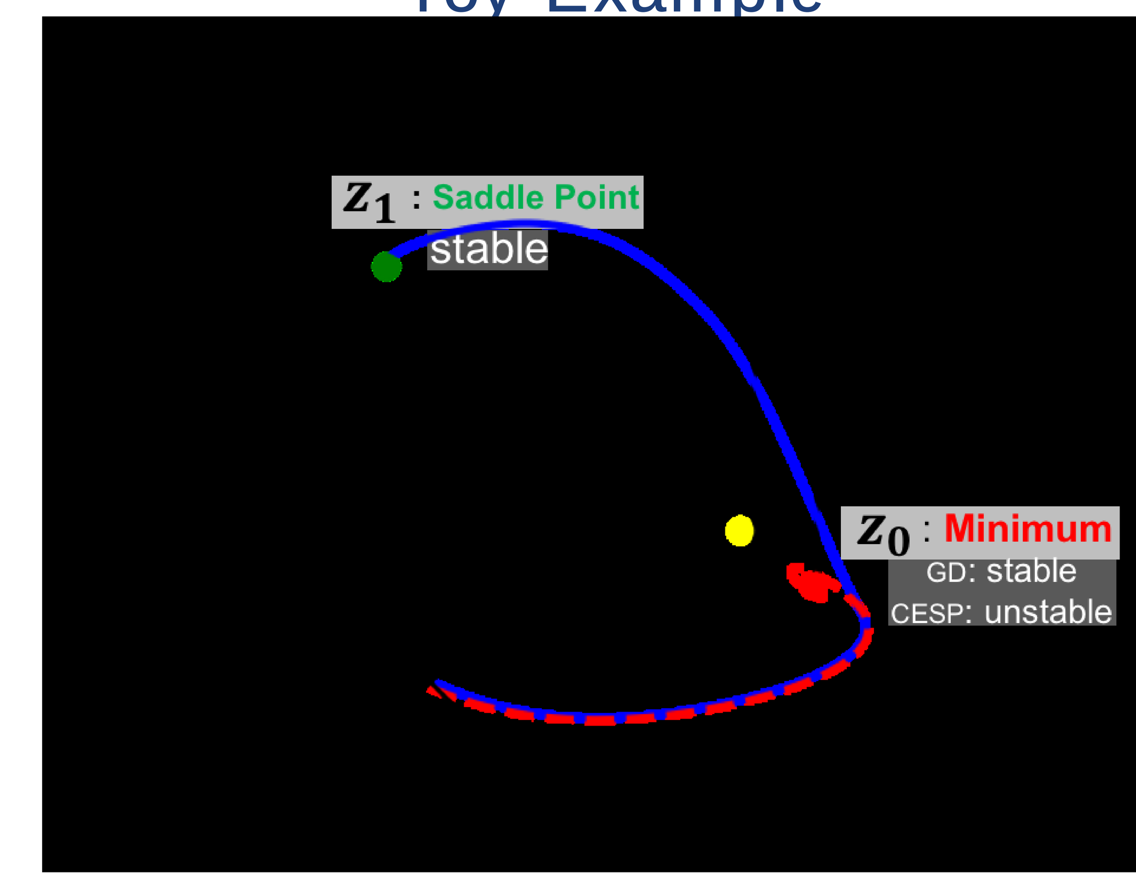


Figure 2: CESP (blue) converges to the saddle point as opposed to gradient-based optimization (red). The vector field shows the *extreme curvature vector* ($v_z^{(-)}$, $v_z^{(+)}$).

CESP in the Real World

- Theoretical guarantees hold also for *transformed* gradient steps, e.g. ADAGRAD.
- Cheap implementation with Power Iterations using only Hessian-vector Products.
- Tested on (small) Generative Adversarial Nets

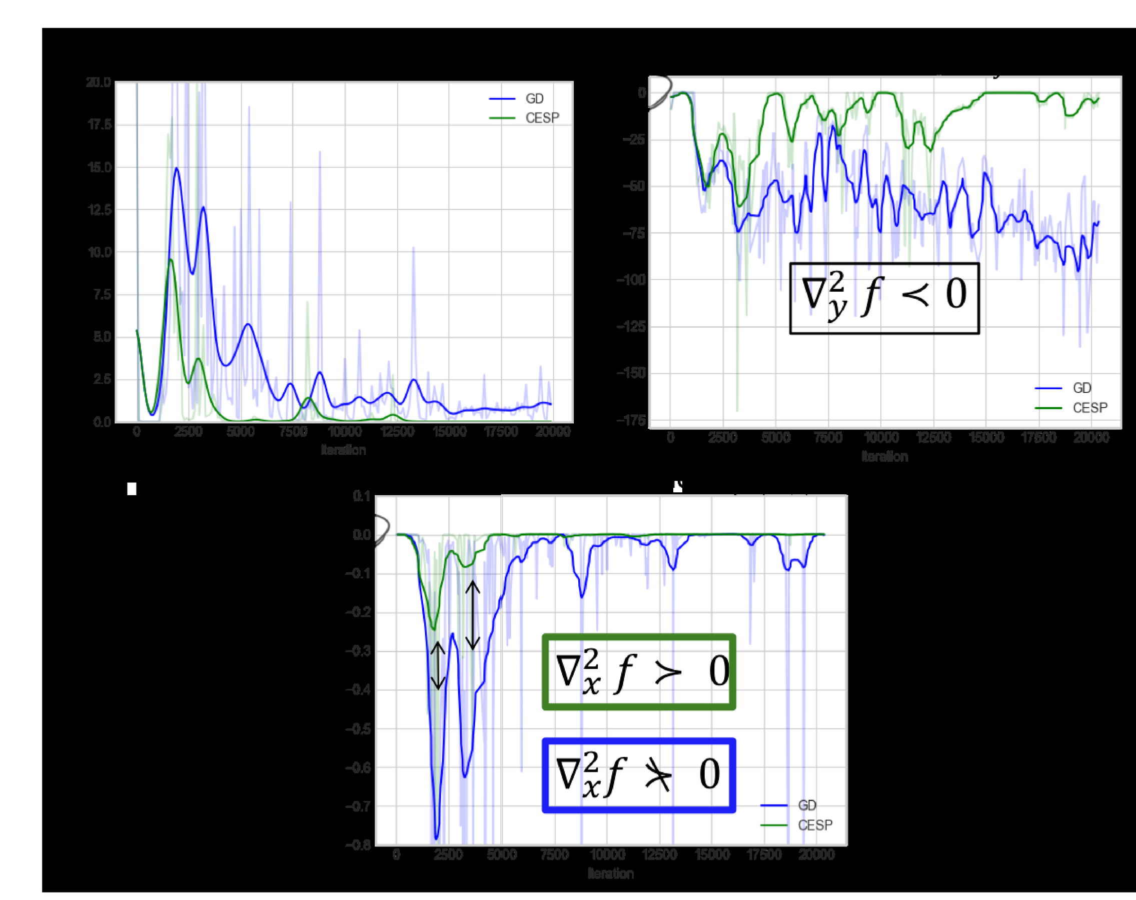


Figure 3: Cesp drives convergent solution to the desired *min-max* structure.

- Many more possible applications, e.g. Robust Optimization for empirical risk minimization