MACHINE LEARNING

WEEK 3: OPTIMIZATION

LECTURE 1: LAGRANGE MULTIPLIERS, DYN. PROG.

Lagrange Multipliers

Minimization is often complicated by the addition of constraints. The simplest kind of constraint is an equality constraint, for example, $G(x) = 0$. Formally this addition is stated as

$$\min_{x} \tilde{F}(x) \text{ subject to } G(x) = 0$$

The method of Lagrange reduces the constrained problem to a new, unconstrained minimization problem with additional variables. The additional variables are known as Lagrange multipliers. To handle this problem, append $G(x)$ to the function $\tilde{F}(x)$ using a Lagrange multiplier λ:

$$F(x, \lambda) = \tilde{F}(x) + \lambda G(x)$$

The Lagrange multiplier is an extra scalar variable, so the number of degrees of freedom of the problem has increased, but the plus side is that now simple, unconstrained minimization techniques can be applied to the composite function. The problem becomes

$$\min_{x,\lambda} F(x, \lambda)$$

Matchbox Example

Max $V = xyz$ STC: $x + 2y + z = L$

Interpretation of Lagrange Multipliers

$$\min_{x,y} F(x,y) \text{ subject to } G(x,y) = 0$$

and suppose for a moment that the constraint equation could be solved so that

$$y = h(x)$$

In that case y can be eliminated from $F(x,y)$, reducing the problem to

$$\min_{x} F[x, h(x)]$$

which can be differentiated using the chain rule to obtain

$$F_x + F_y \frac{dy}{dx} = 0 \tag{1}$$

Now consider moving along the curve $G(x,y) = 0$. Let s be a parameter that varies with arc length so that $\frac{dG}{ds} = 0$, or alternatively

$$G_x \frac{dx}{ds} + G_y \frac{dy}{ds} = 0$$

Solving for $\frac{dy}{dx}$ and substituting into Equation 1,

$$F_x G_y = F_y G_x \tag{2}$$

Constrained Case

$$F'(x, y, \lambda) = F(x, y) + \lambda G(x, y)$$

Differentiating with respect to x and y yields

$$F_x + \lambda G_x = 0$$

and

$$F_y + \lambda G_y = 0$$

Eliminating λ gives the desired result,

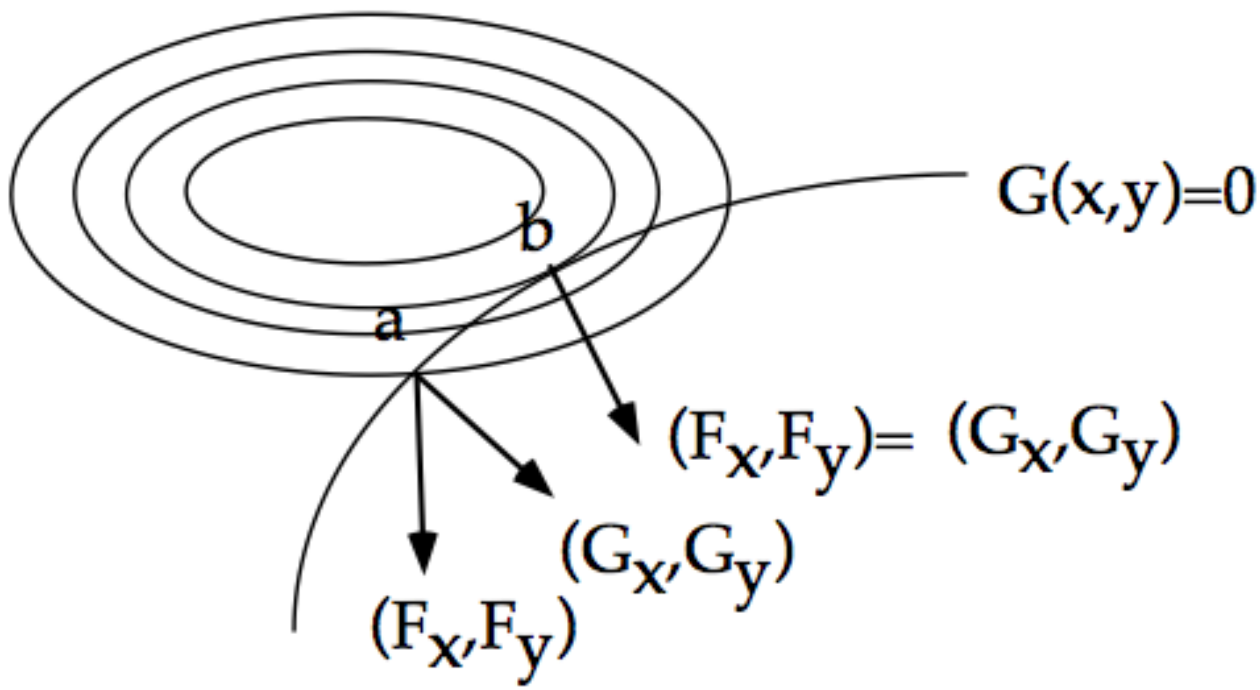
$$F_x G_y = F_y G_x$$

Thus the equation that defines the extrema in the unconstrained problem using Lagrange multipliers is the same as Equation 2, which was obtained by solving for the extrema in the constrained problem.

Geometric Interpretation

$$\frac{F_x}{F_y} = \frac{G_x}{G_y}$$

What this says is that at the extremum, the gradient must be in the same direction as the gradient of level curves of $G = constant$. If this were not true, then one could improve F by sliding along $G = 0$ in the appropriate direction.



Optimal Control

All physical systems will have a dynamics with associated parameters. So a ubiquitous problem is to pick those parameters to maximize some objective function.

Formally the dynamics can be described by

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$$

which starts at an initial state given by

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

and has controllable parameters $\boldsymbol{u}$

$$\boldsymbol{u} \in U$$

The objective function consists of a function of the final state $\psi[\boldsymbol{x}(T)]$ and a cost function (or *loss function*) ℓ that is integrated over time.

$$J = \psi[\boldsymbol{x}(T)] + \int_0^T \ell(\boldsymbol{u}, \boldsymbol{x}) dt$$

Dynamic Programming: Discretization step

First step: make all variables discrete:

The dynamics are now expressed by a difference equation,

$$\boldsymbol{x}(k+1) = f[\boldsymbol{x}(k), \boldsymbol{u}(k)]$$

The initial condition is:

$$\boldsymbol{x}(0) = \boldsymbol{x}_0$$

The allowable control is also discrete:

$$\boldsymbol{u}(k) \in U, k = 0, \ldots, N$$

Finally, the integral in the objective function is replaced by a sum:

$$J = \psi[\boldsymbol{x}(T)] + \sum_{0}^{N-1} \ell[\boldsymbol{u}(k), \boldsymbol{x}(k)]$$

Dynamic Programming: Recursive Eqns.

Working from back to front, the last stage is easy:

$$V(\boldsymbol{x}, N) = \psi[\boldsymbol{x}(N)]$$

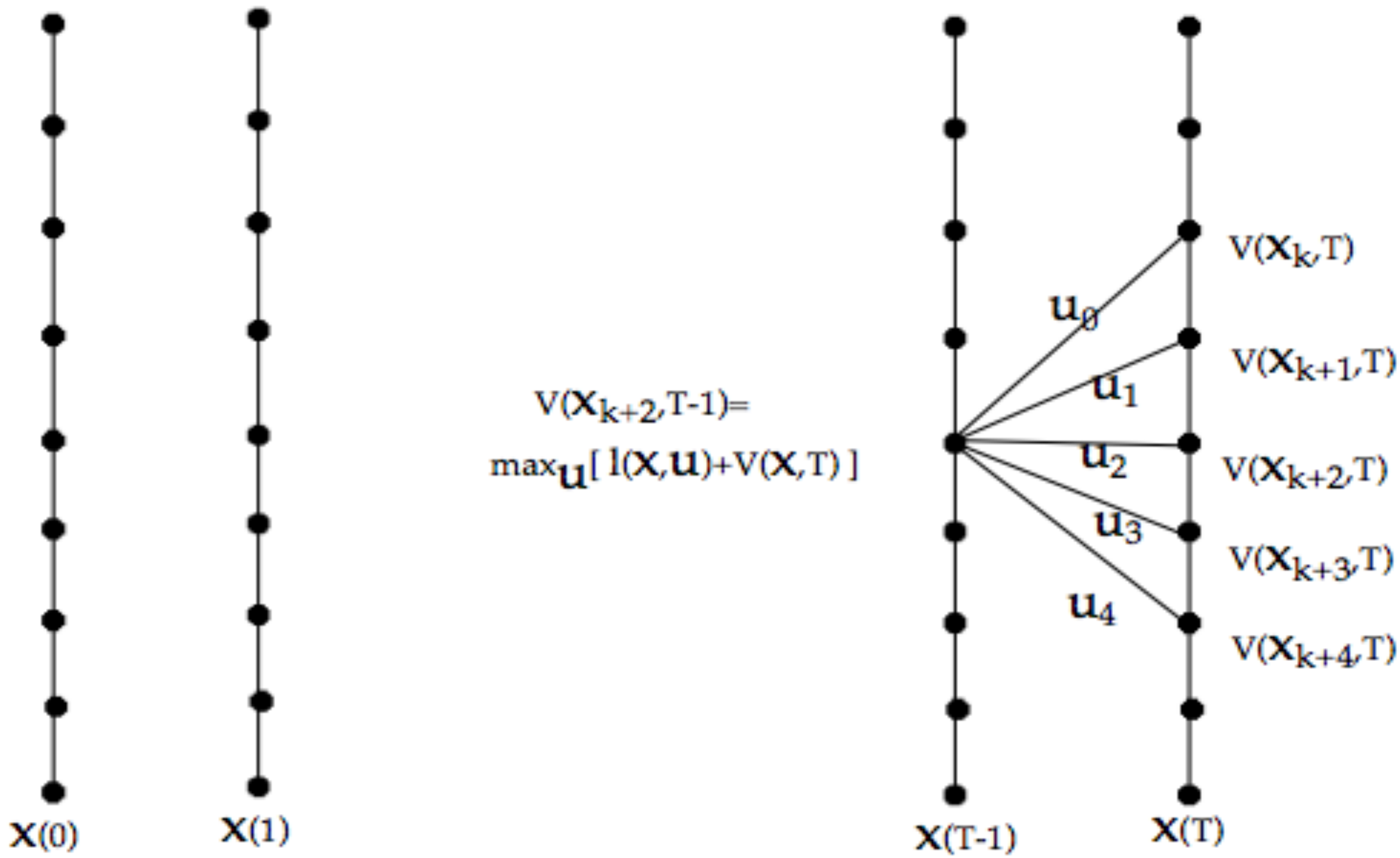
One step back,

$$V(\boldsymbol{x}, N-1) = \max_{\boldsymbol{u} \in U}\{\ell[\boldsymbol{u}(N-1), \boldsymbol{x}(N-1)] + \psi[\boldsymbol{x}(N)]\}$$

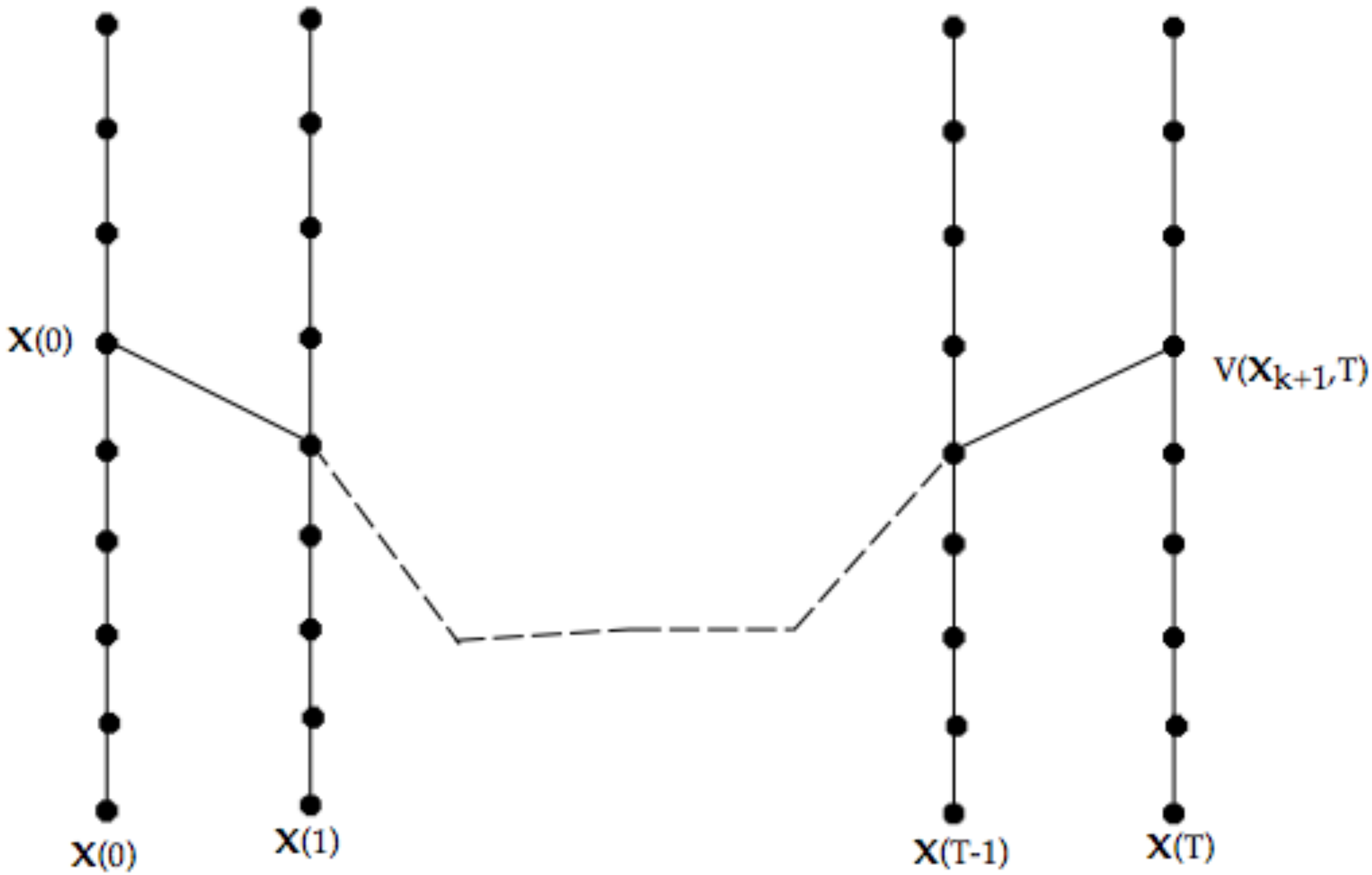
And in general, for $k < N - 1$,

$$V(\boldsymbol{x}, k-1) = \max_{\boldsymbol{u} \in U}\{\ell[\boldsymbol{u}(k-1), \boldsymbol{x}(k-1)] + V[\boldsymbol{x}(k), k]\}, k = 0, \ldots, N$$

Eqns: Illustration



Don't forget to save your path!



Cart Example

$$x(k+1) = x(k) + v(k)$$

$$v(k+1) = v(k) + u(k)$$

and the cost function is

$$J = x(T) - \sum_{0}^{T-1} \frac{u(k)^2}{2}$$

with $T = 9$ and $N = 9$.

Cart Results