A Unified Approach to Optimisation on Manifolds

Jonathan H. Manton

Department of Information Engineering
Research School of Information Sciences and Engineering
The Australian National University

Engineering Mathematics and Applications Conference
September 2005

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

Global Parametrisations

Converting a Constrained Optimisation Problem to an Unconstrained One

- Let $M = \{(x, y) \in \mathbb{R}^2 | x + y = 0\}$ denote a line in $\mathbb{R}^2$.
- Note $\phi : \mathbb{R} \to M$, $\phi(t) = (t, -t)$, is a global parametrisation.

Global Parametrisations

Converting a Constrained Optimisation Problem to an Unconstrained One

- Let $M = \{(x, y) \in \mathbb{R}^2 | x + y = 0\}$ denote a line in $\mathbb{R}^2$.
- Note $\phi : \mathbb{R} \to M$, $\phi(t) = (t, -t)$, is a global parametrisation.
- Therefore, the constrained problem

$$\min_{(x,y)\in M} f(x, y)$$

Global Parametrisations

Converting a Constrained Optimisation Problem to an Unconstrained One

- Let $M = \{(x, y) \in \mathbb{R}^2 | x + y = 0\}$ denote a line in $\mathbb{R}^2$.
- Note $\phi : \mathbb{R} \to M$, $\phi(t) = (t, -t)$, is a global parametrisation.
- Therefore, the constrained problem

$$\min_{(x,y)\in M} f(x, y)$$

can be recast as the unconstrained problem

$$\min_{t\in\mathbb{R}} f \circ \phi(t) = \min_{t\in\mathbb{R}} f(t, -t).$$

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

Local Parametrisations of a Surface (or a Space)

- Global parametrisations do not exist in general; there is no continuous bijection from $\mathbb{R}$ to the circle.

Local Parametrisations of a Surface (or a Space)

- Global parametrisations do not exist in general; there is no continuous bijection from $\mathbb{R}$ to the circle.(If you allow the parametrisation to wrap around, you can use $\phi(\theta) = (\cos\theta, \sin\theta)$, and the general theory of optimisation on manifolds does not preclude this.)

Local Parametrisations of a Surface (or a Space)

- Global parametrisations do not exist in general; there is no continuous bijection from $\mathbb{R}$ to the circle.(If you allow the parametrisation to wrap around, you can use $\phi(\theta) = (\cos\theta, \sin\theta)$, and the general theory of optimisation on manifolds does not preclude this.)
- Let M be a set and, for some fixed n and for each p, let $\phi_p : \mathbb{R}^n \to M$ be a function such that

Local Parametrisations of a Surface (or a Space)

- Global parametrisations do not exist in general; there is no continuous bijection from $\mathbb{R}$ to the circle.(If you allow the parametrisation to wrap around, you can use $\phi(\theta) = (\cos\theta, \sin\theta)$, and the general theory of optimisation on manifolds does not preclude this.)
- Let M be a set and, for some fixed n and for each p, let $\phi_p : \mathbb{R}^n \to M$ be a function such that
  - $\phi_p(0) = p$ and

Local Parametrisations of a Surface (or a Space)

- Global parametrisations do not exist in general; there is no continuous bijection from $\mathbb{R}$ to the circle.(If you allow the parametrisation to wrap around, you can use $\phi(\theta) = (\cos\theta, \sin\theta)$, and the general theory of optimisation on manifolds does not preclude this.)
- Let M be a set and, for some fixed n and for each p, let $\phi_p : \mathbb{R}^n \to M$ be a function such that
  - $\phi_p(0) = p$ and
  - $\phi_p(\mathbb{R}^n)$ contains all points in a neighbourhood of p.

Local Parametrisations of a Surface (or a Space)

- Global parametrisations do not exist in general; there is no continuous bijection from $\mathbb{R}$ to the circle.(If you allow the parametrisation to wrap around, you can use $\phi(\theta) = (\cos\theta, \sin\theta)$, and the general theory of optimisation on manifolds does not preclude this.)
- Let M be a set and, for some fixed n and for each p, let $\phi_p : \mathbb{R}^n \to M$ be a function such that
  - $\phi_p(0) = p$ and
  - $\phi_p(\mathbb{R}^n)$ contains all points in a neighbourhood of p. (M has a topology; ϕ_p is a local homeomorphism.)

Local Parametrisations of a Surface (or a Space)

- Global parametrisations do not exist in general; there is no continuous bijection from $\mathbb{R}$ to the circle.(If you allow the parametrisation to wrap around, you can use $\phi(\theta) = (\cos\theta, \sin\theta)$, and the general theory of optimisation on manifolds does not preclude this.)
- Let M be a set and, for some fixed n and for each p, let $\phi_p : \mathbb{R}^n \to M$ be a function such that
  - $\phi_p(0) = p$ and
  - $\phi_p(\mathbb{R}^n)$ contains all points in a neighbourhood of p. (M has a topology; ϕ_p is a local homeomorphism.)
- How can we use this structure to solve $\min_{x \in M} f(x)$, where $f : M \to \mathbb{R}$?

Optimisation on a Locally Parametrisable Space

- Let $N_g : \mathbb{R}^n \to \mathbb{R}^n$ be an optimisation algorithm, e.g. Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$, for minimising a cost function $g : \mathbb{R}^n \to \mathbb{R}$.

Optimisation on a Locally Parametrisable Space

- Let $N_g : \mathbb{R}^n \to \mathbb{R}^n$ be an optimisation algorithm, e.g. Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$, for minimising a cost function $g : \mathbb{R}^n \to \mathbb{R}$.
- We are currently at the point p_k on M.

Optimisation on a Locally Parametrisable Space

- Let $N_g : \mathbb{R}^n \to \mathbb{R}^n$ be an optimisation algorithm, e.g. Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$, for minimising a cost function $g : \mathbb{R}^n \to \mathbb{R}$.
- We are currently at the point p_k on M.
- If we are happy to restrict attention to a neighbourhood of p_k, it suffices to consider choosing p_{k+1} in the neighbourhood $\phi_{p_k}(\mathbb{R}^n) \subset M$ such that $f(p_{k+1}) < f(p_k)$.

Optimisation on a Locally Parametrisable Space

- Let $N_g : \mathbb{R}^n \to \mathbb{R}^n$ be an optimisation algorithm, e.g. Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$, for minimising a cost function $g : \mathbb{R}^n \to \mathbb{R}$.
- We are currently at the point p_k on M.
- If we are happy to restrict attention to a neighbourhood of p_k, it suffices to consider choosing p_{k+1} in the neighbourhood $\phi_{p_k}(\mathbb{R}^n) \subset M$ such that $f(p_{k+1}) < f(p_k)$.
- This is equivalent to finding a point $x \in \mathbb{R}^n$ such that $f \circ \phi_{p_k}(x) < f(p_k) = f \circ \phi_{p_k}(0)$.

Optimisation on a Locally Parametrisable Space

- Let $N_g : \mathbb{R}^n \to \mathbb{R}^n$ be an optimisation algorithm, e.g. Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$, for minimising a cost function $g : \mathbb{R}^n \to \mathbb{R}$.
- We are currently at the point p_k on M.
- If we are happy to restrict attention to a neighbourhood of p_k, it suffices to consider choosing p_{k+1} in the neighbourhood $\phi_{p_k}(\mathbb{R}^n) \subset M$ such that $f(p_{k+1}) < f(p_k)$.
- This is equivalent to finding a point $x \in \mathbb{R}^n$ such that $f \circ \phi_{p_k}(x) < f(p_k) = f \circ \phi_{p_k}(0)$.
- Thus, we propose $p_{k+1} = \phi_{p_k} \circ N_{f \circ \phi_{p_k}}(0)$.

Optimisation on a Locally Parametrisable Space

- Let $N_g : \mathbb{R}^n \to \mathbb{R}^n$ be an optimisation algorithm, e.g. Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$, for minimising a cost function $g : \mathbb{R}^n \to \mathbb{R}$.
- We are currently at the point p_k on M.
- If we are happy to restrict attention to a neighbourhood of p_k, it suffices to consider choosing p_{k+1} in the neighbourhood $\phi_{p_k}(\mathbb{R}^n) \subset M$ such that $f(p_{k+1}) < f(p_k)$.
- This is equivalent to finding a point $x \in \mathbb{R}^n$ such that $f \circ \phi_{p_k}(x) < f(p_k) = f \circ \phi_{p_k}(0)$.
- Thus, we propose $p_{k+1} = \phi_{p_k} \circ N_{f \circ \phi_{p_k}}(0)$. It works!

Optimisation on a Locally Parametrisable Space

- Let $N_g : \mathbb{R}^n \to \mathbb{R}^n$ be an optimisation algorithm, e.g. Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$, for minimising a cost function $g : \mathbb{R}^n \to \mathbb{R}$.
- We are currently at the point p_k on M.
- If we are happy to restrict attention to a neighbourhood of p_k, it suffices to consider choosing p_{k+1} in the neighbourhood $\phi_{p_k}(\mathbb{R}^n) \subset M$ such that $f(p_{k+1}) < f(p_k)$.
- This is equivalent to finding a point $x \in \mathbb{R}^n$ such that $f \circ \phi_{p_k}(x) < f(p_k) = f \circ \phi_{p_k}(0)$.
- Thus, we propose $p_{k+1} = \phi_{p_k} \circ N_{f \circ \phi_{p_k}}(0)$. It works!
- (For Newton method, require $f \circ \phi_p$ to be twice differentiable at the origin for all $p \in M$. This is a smoothness requirement on the ϕ_p, motivating M being a smooth manifold.)

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

Manifolds

- Roughly, “locally parametrisable spaces” are manifolds.

Manifolds

- Roughly, “locally parametrisable spaces” are manifolds.
- An *n*-dimensional (smooth) manifold is a nice topological space (Hausdorff, paracompact)

Manifolds

- Roughly, “locally parametrisable spaces” are manifolds.
- An *n*-dimensional (smooth) manifold is a nice topological space (Hausdorff, paracompact) which locally looks like $\mathbb{R}^n$ in a topological sense (homeomorphic)

Manifolds

- Roughly, “locally parametrisable spaces” are manifolds.
- An *n*-dimensional (smooth) manifold is a nice topological space (Hausdorff, paracompact) which locally looks like $\mathbb{R}^n$ in a topological sense (homeomorphic) and for which there exists a differentiable structure (so we can make sense of the idea of a smooth function defined on the space).

Manifolds

- Roughly, “locally parametrisable spaces” are manifolds.
- An n-dimensional (smooth) manifold is a nice topological space (Hausdorff, paracompact) which locally looks like $\mathbb{R}^n$ in a topological sense (homeomorphic) and for which there exists a differentiable structure (so we can make sense of the idea of a smooth function defined on the space).
- Simplest examples are “smooth” subsets of $\mathbb{R}^m$, i.e. $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \to \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.

Manifolds

- Roughly, “locally parametrisable spaces” are manifolds.
- An *n*-dimensional (smooth) manifold is a nice topological space (Hausdorff, paracompact) which locally looks like $\mathbb{R}^n$ in a topological sense (homeomorphic) and for which there exists a differentiable structure (so we can make sense of the idea of a smooth function defined on the space).
- Simplest examples are “smooth” subsets of $\mathbb{R}^m$, i.e. $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \to \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.
- In this case, a function $f : M \to \mathbb{R}$ is smooth if there exists an open set $U \subset \mathbb{R}^m$ containing M and a smooth function $\tilde{f} : U \to \mathbb{R}$ such that $f = \tilde{f}|_M$.

Manifolds in Signal Processing

- Manifolds can arise in signal processing

Manifolds in Signal Processing

- Manifolds can arise in signal processing
  - as a constraint set $M = \{x \in \mathbb{R}^m | F(x) = 0\}$

Manifolds in Signal Processing

- Manifolds can arise in signal processing
  - as a constraint set $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \rightarrow \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.

Manifolds in Signal Processing

- Manifolds can arise in signal processing
  - as a constraint set $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \to \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.
  - by quotienting out an ambiguity.

Manifolds in Signal Processing

- Manifolds can arise in signal processing
  - as a constraint set $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \to \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.
  - by quotienting out an ambiguity. For example, if we can identify the channel $h \in \mathbb{C}^m$ only up to scale, then the actual space we are interested in is $M = (\mathbb{C}^m - \{0\})/\sim$ where $h, h' \in \mathbb{C}^m$ are equivalent, $h \sim h'$, iff $\exists \lambda \in \mathbb{C}$, $h = \lambda h'$.

Manifolds in Signal Processing

- Manifolds can arise in signal processing
  - as a constraint set $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \rightarrow \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.
  - by quotienting out an ambiguity. For example, if we can identify the channel $h \in \mathbb{C}^m$ only up to scale, then the actual space we are interested in is $M = (\mathbb{C}^m - \{0\}) / \sim$ where $h, h' \in \mathbb{C}^m$ are equivalent, $h \sim h'$, iff $\exists \lambda \in \mathbb{C}$, $h = \lambda h'$. This is called complex projective space.

Manifolds in Signal Processing

- Manifolds can arise in signal processing
  - as a constraint set $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \to \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.
  - by quotienting out an ambiguity. For example, if we can identify the channel $h \in \mathbb{C}^m$ only up to scale, then the actual space we are interested in is $M = (\mathbb{C}^m - \{0\})/\sim$ where $h, h' \in \mathbb{C}^m$ are equivalent, $h \sim h'$, iff $\exists \lambda \in \mathbb{C}$, $h = \lambda h'$. This is called complex projective space.
  - naturally;

Manifolds in Signal Processing

- Manifolds can arise in signal processing
  - as a constraint set $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \to \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.
  - by quotienting out an ambiguity. For example, if we can identify the channel $h \in \mathbb{C}^m$ only up to scale, then the actual space we are interested in is $M = (\mathbb{C}^m - \{0\})/\sim$ where $h, h' \in \mathbb{C}^m$ are equivalent, $h \sim h'$, iff $\exists \lambda \in \mathbb{C}$, $h = \lambda h'$. This is called complex projective space.
  - naturally; the (n, p)-Grassmann manifold is the set of all p-dimensional linear subspaces of n-dimensional space and can be made naturally into a manifold.

Manifolds in Signal Processing

- Manifolds can arise in signal processing
  - as a constraint set $M = \{x \in \mathbb{R}^m | F(x) = 0\}$ where $F : \mathbb{R}^m \to \mathbb{R}^{m-n}$ is a smooth function whose Jacobian matrix has full row rank for all points $x \in M$.
  - by quotienting out an ambiguity. For example, if we can identify the channel $h \in \mathbb{C}^m$ only up to scale, then the actual space we are interested in is $M = (\mathbb{C}^m - \{0\})/\sim$ where $h, h' \in \mathbb{C}^m$ are equivalent, $h \sim h'$, iff $\exists \lambda \in \mathbb{C}$, $h = \lambda h'$. This is called complex projective space.
  - naturally; the (n, p)-Grassmann manifold is the set of all p-dimensional linear subspaces of n-dimensional space and can be made naturally into a manifold.
- There are corresponding optimisation, tracking and parameter estimation problems on manifolds.

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

The Optimisation Problem

- Not interested in minimising a single function

The Optimisation Problem

- Not interested in minimising a single function — we do it once and that's it —

The Optimisation Problem

- Not interested in minimising a single function — we do it once and that's it — nor in minimising every function

The Optimisation Problem

- Not interested in minimising a single function — we do it once and that's it — nor in minimising every function — no algorithm can perform well for all functions.

The Optimisation Problem

- Not interested in minimising a single function — we do it once and that's it — nor in minimising every function — no algorithm can perform well for all functions.
- Rather, given a class of cost functions (the smaller the better),

The Optimisation Problem

- Not interested in minimising a single function — we do it once and that's it — nor in minimising every function — no algorithm can perform well for all functions.
- Rather, given a class of cost functions (the smaller the better), devise an algorithm taking as input a cost function belonging to this class

The Optimisation Problem

- Not interested in minimising a single function — we do it once and that's it — nor in minimising every function — no algorithm can perform well for all functions.
- Rather, given a class of cost functions (the smaller the better), devise an algorithm taking as input a cost function belonging to this class and computing, in an efficient manner, the minimum of this function.

The Optimisation Problem

- Not interested in minimising a single function — we do it once and that's it — nor in minimising every function — no algorithm can perform well for all functions.
- Rather, given a class of cost functions (the smaller the better), devise an algorithm taking as input a cost function belonging to this class and computing, in an efficient manner, the minimum of this function. (If the class of cost functions contains a single element, compute beforehand the minimum of this function, then propose the algorithm which returns immediately this answer.)

The Optimisation Problem

- Not interested in minimising a single function — we do it once and that's it — nor in minimising every function — no algorithm can perform well for all functions.
- Rather, given a class of cost functions (the smaller the better), devise an algorithm taking as input a cost function belonging to this class and computing, in an efficient manner, the minimum of this function. (If the class of cost functions contains a single element, compute beforehand the minimum of this function, then propose the algorithm which returns immediately this answer.)
- The algorithm design usually needs to address computational complexity per iteration, domain of attraction, convergence rates etc.

Optimisation on Manifolds

- Given a manifold M and a class Ω of cost functions $f : M \to \mathbb{R}$, design an algorithm taking as input an element $f \in \Omega$ and returning $\arg\min_{p \in M} f(p)$.

Optimisation on Manifolds

- Given a manifold M and a class Ω of cost functions $f : M \to \mathbb{R}$, design an algorithm taking as input an element $f \in \Omega$ and returning $\arg\min_{p \in M} f(p)$.
- Our framework is to choose beforehand local parametrisations (local diffeomorphisms) $\phi_p : \mathbb{R}^n \to M$, where $n = \dim M$, and a minimisation scheme $N_g : \mathbb{R}^n \to \mathbb{R}^n$ in Euclidean space.

Optimisation on Manifolds

- Given a manifold M and a class Ω of cost functions $f : M \to \mathbb{R}$, design an algorithm taking as input an element $f \in \Omega$ and returning $\arg\min_{p \in M} f(p)$.
- Our framework is to choose beforehand local parametrisations (local diffeomorphisms) $\phi_p : \mathbb{R}^n \to M$, where $n = \dim M$, and a minimisation scheme $N_g : \mathbb{R}^n \to \mathbb{R}^n$ in Euclidean space.
- The algorithm is then $p_{k+1} = \phi_{p_k} \circ N_{f \circ \phi_{p_k}}(0)$.

Optimisation on Manifolds

- Given a manifold M and a class Ω of cost functions $f : M \to \mathbb{R}$, design an algorithm taking as input an element $f \in \Omega$ and returning $\arg\min_{p \in M} f(p)$.
- Our framework is to choose beforehand local parametrisations (local diffeomorphisms) $\phi_p : \mathbb{R}^n \to M$, where $n = \dim M$, and a minimisation scheme $N_g : \mathbb{R}^n \to \mathbb{R}^n$ in Euclidean space.
- The algorithm is then $p_{k+1} = \phi_{p_k} \circ N_{f \circ \phi_{p_k}}(0)$.
- The ϕ_p should be tailored to Ω

Optimisation on Manifolds

- Given a manifold M and a class Ω of cost functions $f : M \to \mathbb{R}$, design an algorithm taking as input an element $f \in \Omega$ and returning $\arg\min_{p \in M} f(p)$.
- Our framework is to choose beforehand local parametrisations (local diffeomorphisms) $\phi_p : \mathbb{R}^n \to M$, where $n = \dim M$, and a minimisation scheme $N_g : \mathbb{R}^n \to \mathbb{R}^n$ in Euclidean space.
- The algorithm is then $p_{k+1} = \phi_{p_k} \circ N_{f \circ \phi_{p_k}}(0)$.
- The ϕ_p should be tailored to Ω and not to M as such.

Optimisation on Manifolds

- Given a manifold M and a class Ω of cost functions $f : M \to \mathbb{R}$, design an algorithm taking as input an element $f \in \Omega$ and returning $\arg\min_{p \in M} f(p)$.
- Our framework is to choose beforehand local parametrisations (local diffeomorphisms) $\phi_p : \mathbb{R}^n \to M$, where $n = \dim M$, and a minimisation scheme $N_g : \mathbb{R}^n \to \mathbb{R}^n$ in Euclidean space.
- The algorithm is then $p_{k+1} = \phi_{p_k} \circ N_{f \circ \phi_{p_k}}(0)$.
- The ϕ_p should be tailored to Ω and not to M as such.
- We can generalise this framework slightly further. It encompasses all previously proposed optimisation on manifold type algorithms we know of.

Optimisation on Manifolds

- Given a manifold M and a class Ω of cost functions $f : M \to \mathbb{R}$, design an algorithm taking as input an element $f \in \Omega$ and returning $\arg\min_{p \in M} f(p)$.
- Our framework is to choose beforehand local parametrisations (local diffeomorphisms) $\phi_p : \mathbb{R}^n \to M$, where $n = \dim M$, and a minimisation scheme $N_g : \mathbb{R}^n \to \mathbb{R}^n$ in Euclidean space.
- The algorithm is then $p_{k+1} = \phi_{p_k} \circ N_{f \circ \phi_{p_k}}(0)$.
- The ϕ_p should be tailored to Ω and not to M as such.
- We can generalise this framework slightly further. It encompasses all previously proposed optimisation on manifold type algorithms we know of. This generalised framework is genuinely useful - better algorithms.

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

History

- Prior to our own work (2000), the approach to optimisation on manifolds (e.g. Gabay, 1982) always started by endowing the manifold with a metric structure

History

- Prior to our own work (2000), the approach to optimisation on manifolds (e.g. Gabay, 1982) always started by endowing the manifold with a metric structure, then generalising Euclidean algorithms by replacing
  - gradients with (Riemannian) gradients;

History

- Prior to our own work (2000), the approach to optimisation on manifolds (e.g. Gabay, 1982) always started by endowing the manifold with a metric structure, then generalising Euclidean algorithms by replacing
    - gradients with (Riemannian) gradients;
    - Hessians with (Riemannian) Hessians;

History

- Prior to our own work (2000), the approach to optimisation on manifolds (e.g. Gabay, 1982) always started by endowing the manifold with a metric structure, then generalising Euclidean algorithms by replacing
  - gradients with (Riemannian) gradients;
  - Hessians with (Riemannian) Hessians;
  - additions, e.g. $x_{k+1} = x_k + \Delta_k$, with moving along geodesics, e.g. $x_{k+1} = \mathrm{Exp}_{x_k}(\Delta_k)$;

History

- Prior to our own work (2000), the approach to optimisation on manifolds (e.g. Gabay, 1982) always started by endowing the manifold with a metric structure, then generalising Euclidean algorithms by replacing
  - gradients with (Riemannian) gradients;
  - Hessians with (Riemannian) Hessians;
  - additions, e.g. $x_{k+1} = x_k + \Delta_k$, with moving along geodesics, e.g. $x_{k+1} = \mathrm{Exp}_{x_k}(\Delta_k)$;
  - Hessian updates (e.g. in conjugate gradient methods) with tensor updates combined with parallel transport;

History

- Prior to our own work (2000), the approach to optimisation on manifolds (e.g. Gabay, 1982) always started by endowing the manifold with a metric structure, then generalising Euclidean algorithms by replacing
  - gradients with (Riemannian) gradients;
  - Hessians with (Riemannian) Hessians;
  - additions, e.g. $x_{k+1} = x_k + \Delta_k$, with moving along geodesics, e.g. $x_{k+1} = \mathrm{Exp}_{x_k}(\Delta_k)$;
  - Hessian updates (e.g. in conjugate gradient methods) with tensor updates combined with parallel transport;

  and finally, perhaps introducing some approximations to reduce computational complexity.

History

- Prior to our own work (2000), the approach to optimisation on manifolds (e.g. Gabay, 1982) always started by endowing the manifold with a metric structure, then generalising Euclidean algorithms by replacing
  - gradients with (Riemannian) gradients;
  - Hessians with (Riemannian) Hessians;
  - additions, e.g. $x_{k+1} = x_k + \Delta_k$, with moving along geodesics, e.g. $x_{k+1} = \mathrm{Exp}_{x_k}(\Delta_k)$;
  - Hessian updates (e.g. in conjugate gradient methods) with tensor updates combined with parallel transport;

  and finally, perhaps introducing some approximations to reduce computational complexity.
- In general though, this Riemannian structure is artificial, related only to M and not to the class of cost functions Ω.

Comments

- Our framework includes the Riemannian framework as a special case.
- The extra generality allows for the algorithm to be tailored to the actual class of cost functions at hand.
- Universal convergence proofs show that under very mild conditions, any algorithm expressed in our framework will converge locally with the same asymptotic rate as the underlying Euclidean algorithm N_g. (Previously, convergence proofs had to be constructed on a case-by-case basis.)
- We conjecture our framework is sufficiently general such that it captures, in a certain sense, all possible algorithms.

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

Newton Method in a Different Coordinate System

- Recall the Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$.
- If $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is a change of coordinates (diffeomorphism) then we can form a new iteration function

$$E_g(x) = \phi \circ N_{g\circ\phi} \circ \phi^{-1}(x)$$

which is simply N_g implemented in a new coordinate system.

Newton Method in a Different Coordinate System

- Recall the Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$.
- If $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is a change of coordinates (diffeomorphism) then we can form a new iteration function

$$E_g(x) = \phi \circ N_{g \circ \phi} \circ \phi^{-1}(x)$$

  which is simply N_g implemented in a new coordinate system.
- This can alter the domain of convergence, computational complexity, even rate of convergence.

Newton Method in a Different Coordinate System

- Recall the Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$.
- If $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is a change of coordinates (diffeomorphism) then we can form a new iteration function

$$E_g(x) = \phi \circ N_{g\circ\phi} \circ \phi^{-1}(x)$$

  which is simply N_g implemented in a new coordinate system.
- This can alter the domain of convergence, computational complexity, even rate of convergence. (If ϕ can be chosen so that $g \circ \phi$ is quadratic then convergence in a single iteration takes place. Morse's Lemma ensures this can be done locally.)

Newton Method in a Different Coordinate System

- Recall the Newton method $N_g(x) = x - \mathcal{H}_g^{-1}(x)\nabla g(x)$.
- If $\phi : \mathbb{R}^n \to \mathbb{R}^n$ is a change of coordinates (diffeomorphism) then we can form a new iteration function

  $$E_g(x) = \phi \circ N_{g \circ \phi} \circ \phi^{-1}(x)$$

  which is simply N_g implemented in a new coordinate system.
- This can alter the domain of convergence, computational complexity, even rate of convergence. (If ϕ can be chosen so that $g \circ \phi$ is quadratic then convergence in a single iteration takes place. Morse's Lemma ensures this can be done locally.)
- However, E_g is still a "Newton method"; nothing special going on.

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

Coordinate Adapted Newton Method

- The best coordinate system ϕ to use depends on the cost function.

Coordinate Adapted Newton Method

- The best coordinate system ϕ to use depends on the cost function.
- If we know we are close to a minimum, and if we know the class of possible cost functions, then we might have a good idea what the function looks like.

Coordinate Adapted Newton Method

- The best coordinate system ϕ to use depends on the cost function.
- If we know we are close to a minimum, and if we know the class of possible cost functions, then we might have a good idea what the function looks like.
- Therefore, this motivates choosing ϕ to depend on the current iterate!

$$x_{k+1} = \phi_{x_k} \circ N_{g \circ \phi_{x_k}} \circ \phi_{x_k}^{-1}(x_k).$$

Coordinate Adapted Newton Method

- The best coordinate system ϕ to use depends on the cost function.
- If we know we are close to a minimum, and if we know the class of possible cost functions, then we might have a good idea what the function looks like.
- Therefore, this motivates choosing ϕ to depend on the current iterate!

$$x_{k+1} = \phi_{x_k} \circ N_{g \circ \phi_{x_k}} \circ \phi_{x_k}^{-1}(x_k).$$

- This is the algorithm proposed earlier when $M = \mathbb{R}^n$.

Coordinate Adapted Newton Method

- The best coordinate system ϕ to use depends on the cost function.
- If we know we are close to a minimum, and if we know the class of possible cost functions, then we might have a good idea what the function looks like.
- Therefore, this motivates choosing ϕ to depend on the current iterate!

$$x_{k+1} = \phi_{x_k} \circ N_{g \circ \phi_{x_k}} \circ \phi_{x_k}^{-1}(x_k).$$

- This is the algorithm proposed earlier when $M = \mathbb{R}^n$.
- It is not a “Newton method” in any sense; new convergence proofs required etc.

Coordinate Adapted Newton Method

- The best coordinate system ϕ to use depends on the cost function.
- If we know we are close to a minimum, and if we know the class of possible cost functions, then we might have a good idea what the function looks like.
- Therefore, this motivates choosing ϕ to depend on the current iterate!

$$x_{k+1} = \phi_{x_k} \circ N_{g \circ \phi_{x_k}} \circ \phi_{x_k}^{-1}(x_k).$$

- This is the algorithm proposed earlier when $M = \mathbb{R}^n$.
- It is not a "Newton method" in any sense; new convergence proofs required etc.
- Optimisation on manifolds has led to new ideas in the Euclidean case too.

Outline

1. Optimisation using Parametrisations
   - Global Parametrisations
   - Local Parametrisations
2. Optimisation on Manifolds
   - What is a Manifold? When do they arise?
   - The Optimisation on Manifold Problem and our Solution
   - Previous Solutions
3. Coordinate Adapted Newton Method
   - Change of Coordinates
   - Different Change of Coordinates at Each Iteration
   - Example

Example of Coordinate Adapted Newton Method

- Family of cost functions: $f(x; z) = (x - z)^2 + 2(x - z)^3$.

Example of Coordinate Adapted Newton Method

- Family of cost functions: $f(x;z) = (x-z)^2 + 2(x-z)^3$.
- Define $\phi_x(y) = y - (y-x)^2$

Example of Coordinate Adapted Newton Method

- Family of cost functions: $f(x; z) = (x - z)^2 + 2(x - z)^3$.
- Define $\phi_x(y) = y - (y - x)^2$ and note

$$f \circ \phi_z(x) = (x - z)^2 - 5(x - z)^4 + \cdots .$$

Example of Coordinate Adapted Newton Method

- Family of cost functions: $f(x; z) = (x - z)^2 + 2(x - z)^3$.
- Define $\phi_x(y) = y - (y - x)^2$ and note

$$f \circ \phi_z(x) = (x - z)^2 - 5(x - z)^4 + \cdots .$$

- The coordinate adapted Newton method is

$$\begin{aligned} E_f(x; z) &= \phi_x \circ N_{f \circ \phi_x}(x) \\ &= z - 8(x - z)^3 + \cdots \end{aligned}$$

  and has cubic convergence.

Example of Coordinate Adapted Newton Method

- Family of cost functions: $f(x;z) = (x-z)^2 + 2(x-z)^3$.
- Define $\phi_x(y) = y - (y-x)^2$ and note

$$f \circ \phi_z(x) = (x-z)^2 - 5(x-z)^4 + \cdots .$$

- The coordinate adapted Newton method is

$$\begin{aligned} E_f(x;z) &= \phi_x \circ N_{f\circ\phi_x}(x) \\ &= z - 8(x-z)^3 + \cdots \end{aligned}$$

  and has cubic convergence.
- That is, changing coordinate systems at each point can alter significantly the properties of the algorithm.

Conclusion

- The traditional Riemannian approach to optimisation on manifolds does not take into account the class of cost functions at hand.
- We have proposed a more general framework, shown it can lead to better algorithms, and given universal convergence proofs.
- The framework can take any algorithm N_g in Euclidean space and extend it to an algorithm on an arbitrary manifold. Local convergence properties of N_g are preserved in the extended algorithm.
- The degrees of freedom in the extension allow for the domains of convergence, computational complexity etc, to be controlled.