Stochastic gradient descent on Riemannian manifolds

Silvère Bonnabel[1]
Centre de Robotique - Mathématiques et systèmes
Mines ParisTech

SMILE Seminar
Mines ParisTech
Novembre 14th, 2013

[1] silvere.bonnabel@mines-paristech

Introduction

- We proposed a stochastic gradient algorithm on a specific manifold for matrix regression in:
- *Regression on fixed-rank positive semidefinite matrices: a Riemannian approach*, Meyer, Bonnabel and Sepulchre, Journal of Machine Learning Research, 2011.
- Compete(ed) with (then) state of the art for low-rank Mahalanobis distance and kernel learning
- Convergence then left as an open question
- The material of today's presentation is the paper *Stochastic gradient descent on Riemannian manifolds*, IEEE Trans. on Automatic Control, September 2013.

Outline

1. Stochastic gradient descent
    - Introduction and examples
    - SGD and machine learning
    - Standard convergence analysis (due to L. Bottou)
2. Stochastic gradient descent on Riemannian manifolds
    - Introduction
    - Results
3. Examples

Classical example

Linear regression: Consider the linear model

$$y = x^T w + \nu$$

where $x, w \in \mathbb{R}^d$ and $y \in \mathbb{R}$ and $\nu \in \mathbb{R}$ a noise.

- examples: $z = (x, y)$
- loss (prediction error):

$$Q(z, w) = (y - \hat{y})^2 = (y - x^T w)^2$$

- cannot minimize expected risk $C(w) = \int Q(z, w) dP(z)$
- minimize empirical risk instead $\hat{C}_n(w) = \frac{1}{n} \sum_{i=1}^{n} Q(z_i, w)$.

Gradient descent

Batch gradient descent : process all examples together

$$w_{t+1} = w_t - \gamma_t \nabla_w \left(\frac{1}{n} \sum_{i=1}^{n} Q(z_i, w_t) \right)$$

Stochastic gradient descent: process examples one by one

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t)$$

for some random example $z_t = (x_t, y_t)$.

$\Rightarrow$ well known identification algorithm for Wiener systems, ARMAX systems etc.

Stochastic versus online

Stochastic: examples drawn randomly from a finite set

- SGD minimizes the empirical risk

Online: examples drawn with unknown $dP(z)$

- SGD minimizes the expected risk (+ tracking property)

Stochastic approximation: approximate a sum by a stream of single elements

Stochastic versus batch

SGD can converge very slowly: for a long sequence

$$\nabla_w Q(z_t, w_t)$$

may be a very bad approximation of

$$\nabla_w \hat{C}_n(w_t) = \nabla_w \left(\frac{1}{n} \sum_{i=1}^{n} Q(z_i, w_t) \right)$$

SGD can converge very fast when there is redundancy

- extreme case $z_1 = z_2 = \cdots$

Some examples

Least mean squares: Widrow-Hoff algorithm (1960)

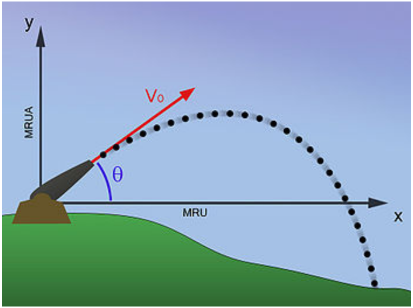
- Loss: $Q(z, w) = (y - \hat{y})^2$
- Update: $w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t) = w_t - \gamma_t (y_t - \hat{y}_t) x_t$

Robbins-Monro algorithm (1951): C smooth with a unique minimum $\Rightarrow$ the algorithm converges in L^2

k-means: McQueen (1967)

- Procedure: pick z_t, attribute it to w^k
- Update: $w_{t+1}^k = w_t^k + \gamma_t (z_t - w_t^k)$

Some examples



Ballistics example (old). Early adaptive control

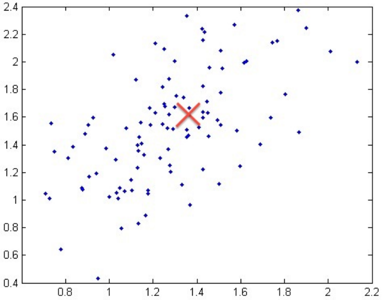
- optimize the trajectory of a projectile in fluctuating wind
- successive gradient corrections on the launching angle
- with $\gamma_t \to 0$ it will stabilize to an optimal value

Another example: mean

Computing a mean: Total loss $\frac{1}{n}\sum_i \|z_i - w\|^2$

Minimum: $w - \frac{1}{n}\sum_i z_i = 0$ i.e. w is the mean of the points z_i

Stochastic gradient: $w_{t+1} = w_t - \gamma_t(w_t - z_i)$ where z_i randomly picked[2]



[2] what if $\|\|$ is replaced with some more exotic distance ?

Outline

1. Stochastic gradient descent
   - Introduction and examples
   - SGD and machine learning
   - Standard convergence analysis (Bottou)
2. Stochastic gradient descent on Riemannian manifolds
   - Introduction
   - Results
3. Examples

Learning on large datasets

Supervised learning problems: infer an input to output function $h : x \mapsto y$ from a training set

Large scale problems: randomly picking the data is a way to handle ever-increasing datasets

Bottou and Bousquet helped popularize SGD for large scale machine learning[3]

---

[3]pointing out there is no need to optimize below approximation and estimation errors (for large but finite number of examples)

Outline

Notation

Expected risk:

$$C(w) := E_z(Q(z,w)) = \int Q(z,w)dP(z)$$

Approximated gradient under the event z denoted by $H(z,w)$

$$E_z H(z,w) = \nabla(\int Q(z,w)dP(z)) = \nabla C(w)$$

Stochastic gradient update: $w_{t+1} \leftarrow w_t - \gamma_t H(z_t, w_t)$

Convergence results

Convex case: known as Robbins-Monro algorithm.
Convergence to the global minimum of $C(w)$ in mean, and almost surely.

Nonconvex case. $C(w)$ is generally not convex. We are interested in proving

- almost sure convergence
- a.s. convergence of $C(w_t)$
- ... to a local minimum
- $\nabla C(w_t) \to 0$ a.s.

Provable under a set of reasonable assumptions

Assumptions

Learning rates: the steps must decrease. Classically

$$\sum \gamma_t^2 < \infty \quad \text{and} \quad \sum \gamma_t = +\infty$$

The sequence $\gamma_t = t^{-\alpha}$, provides examples for $\frac{1}{2} < \alpha \leq 1$.

Cost regularity: averaged loss $C(w)$ 3 times differentiable (relaxable).

Sketch of the proof

1. confinement: w_t remains a.s. in a compact.
2. convergence: $\nabla C(w_t) \to 0$ a.s.

Confinement

Main difficulties:

1. Only an approximation of the cost is available
2. We are in discrete time

Approximation: the noise can generate unbounded trajectories with small but nonzero probability.

Discrete time: even without noise yields difficulties as there is no line search.

SO ? : confinement to a compact holds under a set of assumptions: well, see the paper[4] ...

[4] L. Bottou: Online Algorithms and Stochastic Approximations. 1998.

Convergence (simplified)

Confinement

- All trajectories can be assumed to remain in a compact set
- All continuous functions of w_t are bounded

Convergence

Letting $h_t = C(w_t) > 0$, second order Taylor expansion:

$$h_{t+1} - h_t \leq -2\gamma_t H(z_t, w_t)\nabla C(w_t) + \gamma_t^2 \|H(z_t, w_t)\|^2 K_1$$

with K_1 upper bound on $\nabla^2 C$.

Convergence (simplified)

We have just proved

$$h_{t+1} - h_t \leq -2\gamma_t H(z_t, w_t)\nabla C(w_t) + \gamma_t^2 \|H(z_t, w_t)\|^2 K_1$$

Conditioning w.r.t. $F_t = \{z_0, \cdots, z_{t-1}, w_0, \cdots, w_t\}$

$$E[h_{t+1} - h_t | F_t] \leq \underbrace{-2\gamma_t \|\nabla C(w_t)\|^2}_{\text{this term} \leq 0} + \gamma_t^2 E_z(\|H(z_t, w_t)\|^2) K_1$$

Assume for some $A > 0$ we have $E_z(\|H(z_t, w_t)\|^2) < A$. Using that $\sum \gamma_t^2 < \infty$ we have

$$\sum E[h_{t+1} - h_t | F_t] \leq \sum \gamma_t^2 A K_1 < \infty$$

As $h_t \geq 0$ from a theorem by Fisk (1965) h_t converges a.s. and $\sum |E[h_{t+1} - h_t | F_t]| < \infty$.

Convergence (simplified)

$$E[h_{t+1} - h_t | F_t] \leq -2\gamma_t \|\nabla C(w_t)\|^2 + \gamma_t^2 E_z(\|H(z_t, w_t)\|^2) K_1$$

Both red terms have convergent sums from Fisk's theorem.
Thus so does the blue term

$$0 \leq \sum_t 2\gamma_t \|\nabla C(w_t)\|^2 < \infty$$

Using the fact that $\sum \gamma_t = \infty$ we have[5]

$\nabla C(w_t)$ converges a.s. to 0.

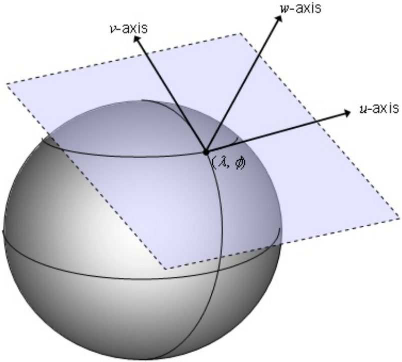
[5] as soon as $\|\nabla C(w_t)\|$ is proved to converge.

Outline

Connected Riemannian manifold

Riemannian manifold: local coordinates around any point

Tangent space:



Riemmanian metric: scalar product $\langle u, v \rangle_g$ on the tangent space

Riemannian manifolds

Riemannian manifold carries the structure of a metric space whose distance function is the arclength of a minimizing path between two points. Length of a curve $c(t) \in \mathcal{M}$

$$L = \int_a^b \sqrt{\langle \dot{c}(t), \dot{c}(t) \rangle_g} dt = \int_a^b \|\dot{c}(t)\| dt$$

Geodesic: curve of minimal length joining sufficiently close x and y.

Exponential map: $\exp_x(v)$ is the point $z \in \mathcal{M}$ situated on the geodesic with initial position-velocity (x, v) at distance $\|v\|$ of x.

Consider $f : \mathcal{M} \to \mathbb{R}$ twice differentiable.

Riemannian gradient: tangent vector at x satisfying

$$\frac{d}{dt}|_{t=0} f(\exp_x(tv)) = \langle v, \nabla f(x) \rangle_g$$

Hessian: operator $\nabla_x^2 f$ such that

$$\frac{d}{dt}|_{t=0} \langle \nabla f(\exp_x(tv)), \nabla f(\exp_x(tv)) \rangle_g = 2\langle \nabla f(x), (\nabla_x^2 f) v \rangle_g.$$

Second order Taylor expansion:

$$f(\exp_x(tv)) - f(x) \leq t\langle v, \nabla f(x) \rangle_g + \frac{t^2}{2} \|v\|_g^2 {\color{red}k}$$

where ${\color{red}k}$ is a bound on the hessian along the geodesic.

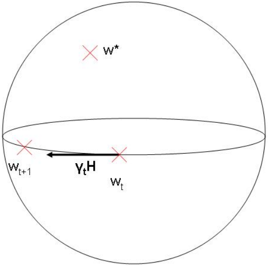
Riemannian SGD on $\mathcal{M}$

Riemannian approximated gradient: $E_z(H(z_t, w_t)) = \nabla C(w_t)$
a tangent vector !

Stochastic gradient descent on $\mathcal{M}$: update

$$w_{t+1} \leftarrow \exp_{w_t}(-\gamma_t H(z_t, w_t))$$

w_{t+1} must remain on $\mathcal{M}$!



Outline

Convergence

Using the same maths but on manifolds, we have proved:

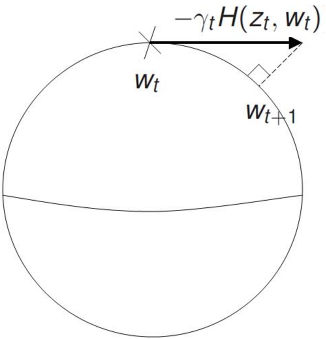
Theorem 1: confinement and a.s. convergence hold under hard to check assumptions linked to curvature.

Theorem 2: if the manifold is compact, the algorithm is proved to a.s. converge under painless conditions.

Theorem 3: same as Theorem 2, where a first order approximation of the exponential map is used.

Theorem 3

Example of first-order approximation of the exponential map:



The theory is still valid ! (as the step $\to 0$)

Outline

General method

Four steps:

1. identify the manifold and the cost function involved
2. endow the manifold with a Riemannian metric and an approximation of the exponential map
3. derive the stochastic gradient algorithm
4. analyze the set defined by $\nabla C(w) = 0$.

Considered examples

- Oja algorithm and dominant subspace tracking
- Matrix geometric means
- Amari's natural gradient
- Learning of low-rank matrices
- Consensus and gossip on manifolds

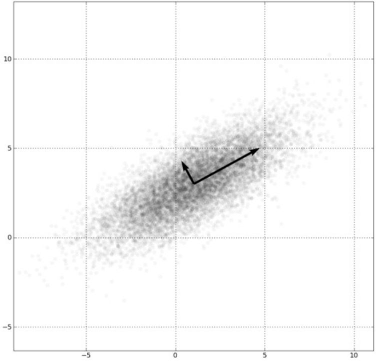
Oja's flow and online PCA

Online principal component analysis (PCA): given a stream of vectors $z_1, z_2, \cdots$ with covariance matrix

$$E(z_t z_t^T) = \Sigma$$

identify online the r-dominant subspace of Σ.

Goal: reduce online the dimension of input data entering a processing system to discard linear combination with small variances. Applications in data compression etc.



Oja's flow and online PCA

Search space: $V \in \mathbb{R}^{r\times d}$ with orthonormal columns. VV^T is a projector identified with an element of the Grassman manifold possessing a natural metric.

Cost: $C(V) = -\mathrm{Tr}(V^T\Sigma V) = E_z\|VV^Tz - z\|^2 + cst$

Riemannian gradient: $(I - V_tV_t^T)z_tz_t^TV_t$

Exponential approx: $R_V(\Delta) = V + \Delta$ plus orthonormalisation

Oja flow for subspace tracking is recovered

$V_{t+1} = V_t - \gamma_t(I - V_tV_t^T)z_tz_t^TV_t$ plus orthonormalisation.

Convergence is recovered within our framework (Theorem 3).

Considered examples

- Oja algorithm and dominant subspace tracking
- Positive definite matrix geometric means
- Amari's natural gradient
- Learning of low-rank matrices
- Decentralized covariance matrix estimation

Filtering in the cone $P^+(n)$

Vector-valued image and tensor computing

Results of several filtering methods on a 3D DTI of the brain[6]:

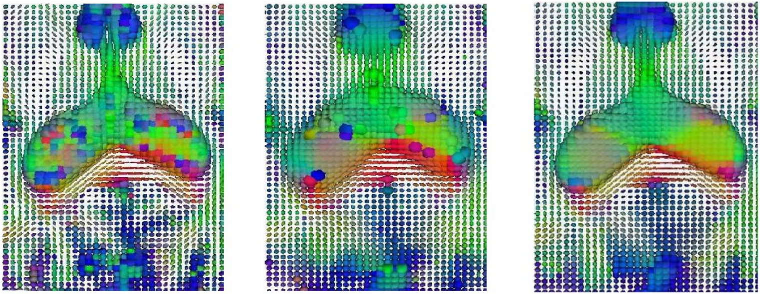


Figure: Original image “Vectorial" filtering “Riemannian" filtering

[6] Courtesy from Xavier Pennec (INRIA Sophia Antipolis)

Matrix geometric means

Natural geodesic distance d in $P_+(n)$.

Karcher mean: minimizer of $C(W) = \sum_{i=1}^N d^2(Z_i, W)$.

No closed form solution of the Karcher mean problem.

A Riemannian SGD algorithm was recently proposed[7].

SGD update: at each time pick Z_i and move along the geodesic with intensity $\gamma_t d(W, Z_i)$ towards Z_i

Convergence can be recovered within our framework.

[7] Arnaudon, Marc; Dombry, Clement; Phan, Anthony; Yang, Le *Stochastic algorithms for computing means of probability measures* Stochastic Processes and their Applications (2012)

Considered examples

- Oja algorithm and dominant subspace tracking
- Positive definite matrix geometric means
- Amari's natural gradient
- Learning of low-rank matrices
- Decentralized covariance matrix estimation

Amari's natural gradient

Natural gradient works efficiently in learning
SI **Amari** - Neural computation, 1998 - MIT Press
When a parameter space has a certain underlying structure, the ordinary **gradient** of a function does not represent its steepest direction, but the **natural gradient** does. Inform geometry is used for calculating the **natural** gradients in the parameter space of ...
Cité 1358 fois - Autres articles - Les 19 versions

Considered problem: z_t are realizations of a parametric model with parameter $w \in \mathbb{R}^n$ and pdf $p(z; w)$. Let

$$Q(z, w) = -l(z; w) = -\log(p(z; w))$$

Cramer-Rao bound: any unbiased estimator $\hat{w}$ of w based on the sample $z_1, \cdots, z_k$ satisfies

$$\mathrm{Var}(\hat{w}) \geq \frac{1}{k} G(w)^{-1}$$

with $G(w)$ the Fisher Information Matrix.

Amari's natural gradient

Fisher Information (Riemannian) Metric at w:

$$\langle u, v\rangle_w = u^T G(w) v$$

Riemannian gradient of $Q(z, w)$ **= natural gradient**

$$-G^{-1}(w)\nabla_w l(z, w)$$

Exponential approximation: simple addition $R_w(u) = w + u$. Taking $\gamma_t = 1/t$ we recover the celebrated

Amari's natural gradient: $w_{t+1} = w_t - \frac{1}{t} G^{-1}(w_t)\nabla_w l(z_t, w_t)$.

Fits in our framework and a.s. convergence is recovered

Considered examples

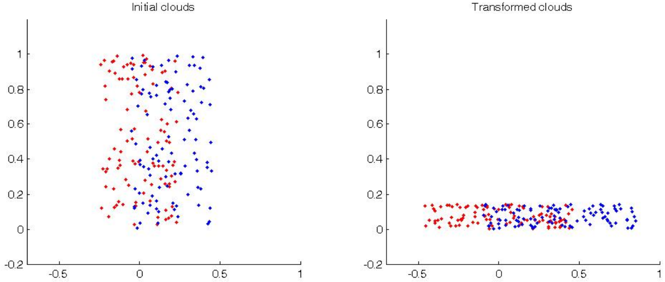
- Oja algorithm and dominant subspace tracking
- Positive definite matrix geometric means
- Amari's natural gradient
- Learning of low-rank matrices
- Decentralized covariance matrix estimation

Mahalanobis distance learning

Mahalanobis distance: parameterized by a positive semidefinite matrix W (inv. of cov. matrix)

$$d_W^2(x_i, x_j) = (x_i - x_j)^T W(x_i - x_j)$$

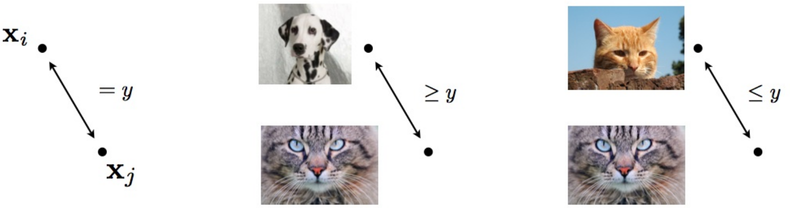
Learning: Let $W = GG^T$. Then d_W^2 simple Euclidian squared distance for transformed data $\tilde{x}_i = Gx_i$. Used for classification



Mahalanobis distance learning

Goal: integrate new constraints to an existing W

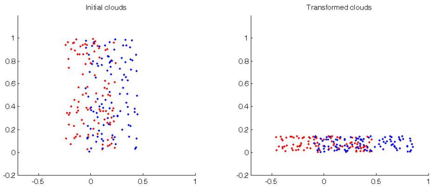
- equality constraints: $d_W(x_i, x_j) = y$
- similarity constraints: $d_W(x_i, x_j) \leq y$
- dissimilarity constraints: $d_W(x_i, x_j) \geq y$



Computational cost significantly reduced when W is low rank !

Interpretation and method

One could have projected everything on a horizontal axis ! For large datasets low rank allows to derive algorithm with linear complexity in the data space dimension d.



Four steps:

1. identify the manifold and the cost function involved
2. endow the manifold with a Riemannian metric and an approximation of the exponential map
3. derive the stochastic gradient algorithm
4. analyze the set defined by $\nabla C(w) = 0$.

Semi-definite positive matrices of fixed rank

$$S^+(d,r) = \{W \in \mathbb{R}^{d\times d}, W = W^T, W \succeq 0, \text{rank W} = r\}$$

Regression model: $\hat{y} = d_W(x_i, x_j) = (x_i - x_j)^T W (x_i - x_j)$,

Risk: $C(W) = E((\hat{y} - y)^2)$

Catch: $W_t - \gamma_t \nabla_{W_t}((\hat{y}_t - y_t)^2)$ has NOT same rank as W_t.

Remedy: work on the manifold !

Considered examples

- Oja algorithm and dominant subspace tracking
- Positive definite matrix geometric means
- Amari's natural gradient
- Learning of low-rank matrices
- Decentralized covariance matrix estimation

Decentralized covariance estimation

Set up: Consider a sensor network, each node i having computed its own empirical covariance matrix $W_{i,0}$ of a process.

Goal: Filter the fluctuations out by finding an average covariance matrix.

Constraints: limited communication, bandwith etc.

Gossip method: two random neighboring nodes communicate and set their values equal to the average of their current values.
$\Rightarrow$ should converge to a meaningful average.

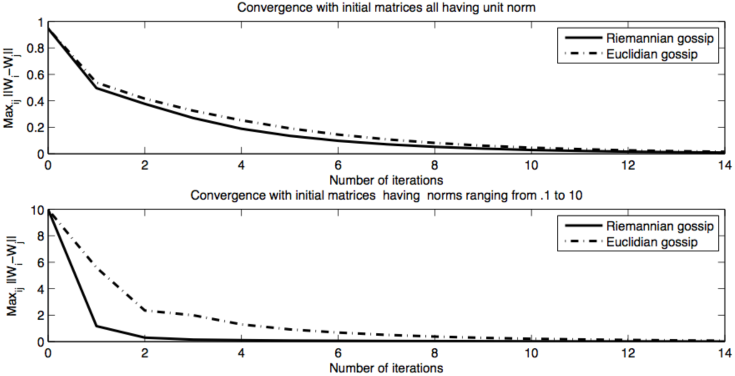
Alternative average why not the midpoint in the sense of Fisher-Rao distance (leading to Riemannian SGD)

$$d(\Sigma_1, \Sigma_2) \approx KL(\mathcal{N}(0, \Sigma_1) \,||\, \mathcal{N}(0, \Sigma_2))$$

Example: covariance estimation

Conventional gossip at each step the usual average $\frac{1}{2}(W_{i,t} + W_{j,t})$ is a covariance matrix, so the algorithms can be compared.

Results: the proposed algorithm converges much faster !



Conclusion

We proposed an intrinsic SGD algorithm. Convergence was proved under reasonable assumptions. The method has numerous applications.

Future work includes:

- better understand consensus on hyperbolic spaces
- speed up convergence via Polyak-Ruppert averaging $\overline{w}_t = \sum_{i=0}^{t-1} w_i$: generalization to manifolds non-trivial
- tackle new applications: online learning of rotations