Learning transformations between directed subspaces ... online!

Adam Smith
amsmith@cs.ucsc.edu

University of California - Santa Cruz

June 7, 2006 - CMPS 290C
Last update - June 14, 2007

Outline

Motherspace

The motherspace $\mathcal{M}$ ...

- is an infinite dimensional vector space
- includes every subspace of instances we might consider
- includes every subspace of labels we might consider

Transformation

Some linear function $f(\cdot)$

- maps instances vectors to label vectors ($\mathbf{y}_t = f(\mathbf{x}_t)$)
- may really be from a restricted class of functions
- may shift over time (need to adapt online!)

Homogeneous Transformation Problem

Definition

The homogeneous tranformation problem

- Receive an instance vector $\mathbf{x}_t$ in instance space.
- Predict a label vecor $\hat{\mathbf{y}}_t$ in motherspace.
- Receive true label vector $\mathbf{y}_t$ in label space.
- Incur a loss $L_{\mathbf{y}_t}(\hat{\mathbf{y}}_t)$.

Solution

- pick some class of linear functions and find a parameterized form
- initialize parameter $\mathbf{P}$ to something reasonable
- predict $\mathbf{y}_t = f_{\mathbf{P}}(\mathbf{x}_t)$
- update parameter to minimize tradeoff of divergence from last value and loss
- exactly $\mathbf{P}_{t_1} = \inf_{\mathbf{P} \in \mathcal{P}} \{\Delta(\mathbf{P}, \mathbf{P}_t) + \eta L_{\mathbf{y}_t}(f_{\mathbf{P}}(\mathbf{x}_t))\}$

Outline

Universal Geometric Algebra $\mathcal{G}$

- set: an infinite dimensional vector space
- operators: geometric sum and the (non-commutative) geometric product
- the closure of operations on the space is called $\mathcal{G}$ (the algebra of the multivectors)
- all real vector algebras are subalgebras of $\mathcal{G}$ (as is the algebra of the real numbers)
- <u>contains a subalgebra just for our problem!</u>

Some classes of linear functions

Focus on classes of linear functions with a simple, interpretable parameter.

- reflection (useful in embeddings)
  - parameter is a vector $\mathbf{n}$, the normal to the hyperplane of reflection
  - $f_{\mathbf{n}}(\mathbf{x}) = -\mathbf{nxn}$
- rotation (we will focus here shortly)
  - parameter is a rotor $\mathbf{R}$, the two-dimensional plane of rotation and angle
  - a quaternion-like package of a scalar and a bivector (in general, the product of two unit vectors)
  - $f_{\mathbf{R}}(\mathbf{x}) = \mathbf{Rx}$
- projection onto a subspace (good for PCA?)
  - a multivector $\mathbf{S}$ that is the geometric product of vectors spanning that space
  - $f_{\mathbf{S}}(\mathbf{x}) = (\mathbf{x} \cdot \mathbf{S})\mathbf{S}^{-1}$
- scale (not very interesting, but still in same setting)
  - a scalar λ
  - $f_{\lambda}(\mathbf{x}) = \lambda\mathbf{x}$

Divergences

Identify the divergence $\Delta(.,.)$

- $|A-B|^2$, the "norm squared of difference" is a natural choice
  - defined the same way for all multivectors
  - for multivectors of the same grade (just scalars, just vectors, just bivectors, etc) it is exactly the euclidean distance
- ???, a "geometric relative entropy"
  - $\Delta(A)=\frac{1}{2}\langle < AlogA - A + 1\rangle >$ is an entropy-like measure defined for all multivectors
  - for rotors, it yields a 'distance from identiy rotation' (proportional to $-\theta\sin\theta - 2\cos\theta$)
  - only convex on hemisphere of rotations near identity (not unexpected)
  - no clean Bregman divergence derived yet

Losses

Identify the loss $L_{\mathbf{y}_t}(.)$

- norm squared of difference is suitable, $L_{\mathbf{y}_t}(\hat{\mathbf{y}}_t) = |\mathbf{y}_t - \hat{\mathbf{y}}_t|^2$
- geometric relative entropy may be suitable too (someday) $(L_{\mathbf{y}_t}(\hat{\mathbf{y}}_t) = \Delta(\mathbf{y}_t \mathbf{y}_t))$
- we just need *some* notion of a distance between vectors in motherspace

Mapping into Motherspace

Some coordinate-free and domain-indepdendent methods of mapping application-level vectors into the motherspace:

- identity transformation - consider only homogeneous subspace of original space
- projective transformation - consider non-homogeneous spaces in ororiginalig space
- conformal transformation - consider generalized circles in original space
- conic transformaion - consider generalized conics in original space

General strategy: make homogeneous subspaces more powerful!
Domain-specific transformations should be applied outside of this discussion.

Outline

Rotation Problem

Definition

Rotation Problem

- instances come from a specific n-dimensional vector space $\mathcal{V}$
- labels also come from $\mathcal{V}$
- an hidden rotation $\mathbf{R_C}$ correctly maps the data
- hidden rotation may shift over time (adapt online!)

Attack

- use rotations as the class of linear functions
- use the norm squared of difference as the divergencne
- use the norm squared of difference as the loss
- use the identity transformation to map vectors (no changes)

Details

- $\mathbf{R}$ is the parameter, has $1 + n(n-1)/2$ degrees of freedom, one less if we keep it normalized
- $\mathbf{R}_1 = 1$ (just the scalar), the identity rotation
- predict with $f_{\mathbf{R}}(\mathbf{x}) = \mathbf{R}\mathbf{x}\mathbf{R}^{-\mathbf{1}}$ *
- update with $\mathbf{R}_{t+1} = \inf_{|\mathbf{R}|^2=1} \left\{ |\mathbf{R} - \mathbf{R}_t|^2 \; + \; \eta |\mathbf{R}\mathbf{x} - \mathbf{y}_t|^2 \right\}$
- at every step we may interpret the state of the algorithm by decomposing $\mathbf{R} = e^{-\mathbf{B}\theta}$ where $\mathbf{B}$ is the plane of rotation (a bivector) and θ is the half angle of rotation in radians

Deriving the Update

Actual ugly math ommitted for the talk, general idea:

- form Lagrangian of minimization problem with dual parameter α to enforce normalization contraint
- differentiate with respect to $\mathbf{R}$ (using geometric calculus!)
- set differential equal to zero
- solve for $\mathbf{R}$
- solve for α to enforce normalization

And the result is not pretty in component form, but is equivalent to ...

Geometric Update

$$\boxed{\mathbf{R}_{t+1} = \frac{\mathbf{R_t} \ + \ \eta \mathbf{y_t x_t}}{\alpha}}$$

Interpretation

"The best new rotor is simply the normalized sum of the previous rotor and the shortest rotor that explains the mapping from **x** to **y** (weighted by the tradeoff factor)."

Comments

- *the math is devastatingly elegant (IMHO)*
- can be loosely connected back to expert framework: "There is an expert for (half) rotation in each of the principle *planes* plus one extra for the identity rotation."
- this algorithm was derived completely mechanically (setting + problem $\rightsquigarrow$ algorithm)
- actually indepdentent of the dimensionality and signature of instance and label spaces (may even be orthogonal!)
- it has clear advantages over other formulations that the problem might suggests without adopting geometric algebra (oh really?)

Comments - Rotation Matrix

What if parameter is a rotation matrix?

- update is not coordinate-free
- no clear interpretation of matrix after an update (but an un-normalized rotor performs the same rotation as the normalized one! – the set is closed!!!)
- requires orthonormalization to rebuild rotation matrix after update, might undo progess (rotor only requires a divide of all components in parallel)
- has lots of extra degrees of freedom

Comments - Euler Angles

What if parameter is vector of Euler angles (rotation about each principle *axis*)?

- update is not coordinate-free
- update is makes heavy use of trigonometric functions
- sensitivity to noise is not uniform (gimbal lock, singularties)

Comments - Quaternions

What if parameter a quaternion?

- limited to only 3D case
- rotor algorthm reduces to quaternion case in 3d

Outline

Future Work Directions

- get formal bounds (already known to converge for batch case)
- explore "geometric relative entropy" more (for other parameter classes as well)
- try to derive online PCA using subspace projections
- find interpretable parameter(s) (not a matrix) for other important linear function classes
- plug into some real-world thingy

It's a $\mathcal{G}$ thing.