Quantum Machine Group Learning

Carlos Tavares

University of Minho
High-Assurance Software Laboratory-INESC TEC

ctavares@inesctec.pt

November 13, 2017

Overview

Groups

A symmetry of an object is a transformation that leaves certain properties of that object intact (invariants) [GD14]

Group algebra

- Associativity

$$\forall_{a,b,c \in G} : (a \circ b) \circ c = a \circ (b \circ c) \tag{1}$$

- Identity element

$$\exists_{e \in G} \forall_{a \in G} : e \circ a = a \circ e = a \tag{2}$$

- Closure

$$\forall_{a,b \in G} : (a \circ b) \in G \tag{3}$$

- Inverse element

$$\forall_{a \in G} \exists_{b \in G} : a^{-1} = b, a \circ b = b \in a = e \tag{4}$$

Groups are the natural mathematical tool to model symmetries!

Groups

Example

Common groups (infinite group)

- $\mathbb{Z}$, under the $+$ operation

$$\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots \tag{5}$$

  $\mathbb{Q}, \mathbb{R}, \mathbb{C}$

- Cyclic group $\mathbb{Z}/n\mathbb{Z}$, integers modulo n (finite group)
  Example: x mod 6

$$0, 1, 2, 3, 4, 5 \tag{6}$$

Abelian and Non-Abelian groups

Abelian groups:$\forall_{g,h \in G} : g \circ h = h \circ g$; Non-Abelian groups: $\exists_{g,h \in G} : g \circ h \neq h \circ g$

Groups

Example

Non-Abelian groups

- Dihedral group

- Symmetry group: group of the permutations of a set $\pi : X \to X$

$$1,2,3 \xrightarrow{f_1} 2,1,3 \xrightarrow{f_2} 2,3,1 \xrightarrow{f_3} 1,3,2 \xrightarrow{f_1} 3,1,2 \xrightarrow{f_2} 3,2,1 \qquad (7)$$

f_1, f_2, f_3 plus the identity transformation form a group!

Example

Groups in physics Symmetries in physics: conservation laws

- Lorentz group, Poincare group, Lie groups in field theory

Groups in Machine learning

Can groups be on any use in machine learning: yes! [GD14], [Kon08]

Definition

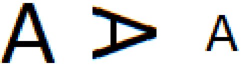
Objective A classification function invariant to group actions

$$\forall_{g \in G} f(x) = f(T_g(x)) \tag{8}$$

Example

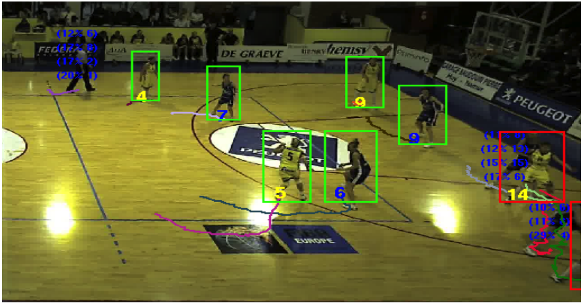
Computer Vision: Images are invariant to a large set of transformations. Many can be modeled by groups of increasing complexity.

- Shifts, Rotations, Scalings, Distortions (changes of coordinates)



Groups in Machine learning

Object tracking using the symmetry group



Picture available in:
https://sites.uclouvain.be/ispgroup/uploads//Research/IdentityAssignment.png

Other examples: Permutation learning using the symmetry group

- More examples?

The approach of using groups quickly gets intractable, depending on the groups involved!

Kernel methods

Kernel methods are a classification/regression technique with very solid foundations in statistical learning theory[HSS08]

Classification/Regression problems

Given a set of training examples, associating inputs (x) to outputs (y)

$$(x_1, y_1), \ldots, (x_n, y_n) \in X \times Y \tag{9}$$

Estimate the function that given an x, outputs the correspondent y.

$$f : X \rightarrow Y \tag{10}$$

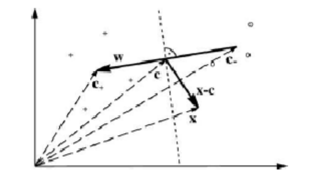
Find the function f that minimizes a loss function L.

$$R_{reg}[f] = \frac{1}{m} \sum_{i=1}^{m} L(f(x_i), y_i) \tag{11}$$

Kernel methods

Example

A very simple classifier



$+, -$ - classification classes, c_+, c_- - average points; $\vec{w} = \vec{c_+} - \vec{c_-}$; $\vec{c} = (\vec{c_+} - \vec{c_-})/2$;
$\vec{v} = \vec{x} - \vec{c}$; $h = \vec{w} \cdot \vec{v}$

A measurement of similarity was required!

Kernel methods

Kernel are a generalized notion of similarity between points in the higher dimensional space [HSS08], they also target generalization.

Kernels

$$k : X \times X \to \mathbb{R} \tag{12}$$

- Kernels are not universal!
  - "Famous" Kernels: Gaussian Kernel, ANOVA kernel, sparse vector kernel, Polynomial Kernel
- Some can be transported to a *feature space*: the kernel can be mapped product in an Hilbert space

$$k(x, x') \to \langle \Phi(x), \Phi(x') \rangle \tag{13}$$

- Positive definite kernels can induce Hilbert spaces: Regularized Kernel Hilbert Spaces (RKHS)
  - To avoid overfitting regularization is required!

Kernel methods

Regularized RKHS

RKHS, provides a natural regularized setting for functions based on kernel

$$R_{reg}[f] = \frac{1}{m}\sum_{i=1}^{m} L(f(x_i), y_i) + \lambda ||f||^2 \quad (14)$$

Representer theorem: An extensive set of functions can be reduced to linear expansions of the kernels of the training examples

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i) \quad (15)$$

The problem of optimization is reduced to finding the correct $\alpha_i \in \mathbb{R}$ coeficients

Groups in Machine learning

Invariant RKHS kernels [Kon08]

- Objective: induce a $RKHS_{inv}$ of functions invariant to an action of a group G

$$\forall_{g \in G} f(x) = f(T_g(x)) \quad (16)$$

- The kernel must be also invariant to the action of the group and positive definite
- An invariant kernel on the group can be defined by a positive definite function on the group, which, being positive definite has a Fourier transform.

FFTs can be of great use in machine learning by helping in build kernels invariant to certain groups. Several examples for non-trivial groups, such as the symmetry group exists [Kon08].

Groups and FFT's in Quantum computation

There is a close relationship between the hidden subgroup problem and Fourier transform

HSP and Fourier transform

- A few interesting results with non-Abelian groups [HRTS00] , however the most relevant groups are still out of reach: Dihedral group and Symmetry group [EH00]

Fourier transform:

Group family	Classical	Quantum
Abelian groups	$NlogN$ [CT65]	$(logN)$ [HH00]
...	...	...
Symmetry group	$N!$ [CB93]	$Polynomial$ [Bea97]

Can we still provide any type of advantage to kernel based methods?

Quantum Support Vector Machines

Many quantum SVM implementions are availale: [Wit14]

Quantum SVM [RML14]

Objective:

$$y(\vec{x}) = sign(\sum_{j=1}^{M} \alpha_j k(\vec{x}_j, \vec{x}) + b) \tag{17}$$

State preparation I

- Assume oracles capable of preparing states corresponding to each training samples x:

$$|\vec{x_j}\rangle = \frac{1}{\sqrt{|\vec{x_j}|}} \sum_{k=1}^{K} (\vec{x_j})_k |k\rangle \tag{18}$$

Quantum Support Vector Machines

State preparation II

- Apply the oracles over a superposition state

$$|\Psi\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |i\rangle \xrightarrow{O} |\chi\rangle = \frac{1}{\sqrt{N_\chi}} \sum_{i=1}^{M} |\vec{x}_i| \, |i\rangle \, |\vec{x}_i\rangle \tag{19}$$

- We obtain the Kernel matrix, as the density matrix of the state

$$\hat{K} = \frac{K}{trK} \tag{20}$$

Optimization I

- The optimization problem can be reduced to a minimum squares problem!

$$F \begin{pmatrix} a \\ \vec{\alpha} \end{pmatrix} \equiv \begin{pmatrix} 0 & \vec{1}^T \\ \vec{1} & K + \gamma^{-1} \not{\mathbb{K}} \end{pmatrix} \begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{y} \end{pmatrix} \tag{21}$$

Quantum Support Vector Machines

Optimization II

- ... where $K_{ij} = \vec{x}_i^T \cdot x_j$, $\vec{y} = \{y_1, \ldots, y_n\}^T, \vec{1} = \{1, \ldots, 1\}^T$

The objective state $(b, \vec{\alpha}^T)^T$ can be obtained through the exponentiation and inverse of the F matrix

$$(b, \vec{\alpha}^T)^T = F^{-1}(0, y^{-T})^T \tag{22}$$

... which is expontially faster to obtain in a quantum computer. The end state after this process reads as follows

$$|b, \vec{\alpha}\rangle = \frac{1}{\sqrt{C}} \sum_{k=1}^{M} \alpha_k |k\rangle \tag{23}$$

Quantum Support Vector Machines

Classification

By constructing a few states and making measurements it is possible to classify probabilistically a *query state* $\vec{x}$

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N_{\tilde{x}}}} \sum_{k=1}^{M} |\vec{x}|\,|k\rangle\,|\vec{x}\rangle \tag{24}$$

$$|b, \vec{\alpha}\rangle \times |\tilde{x}\rangle \xrightarrow{f \circ m} +1, -1 \tag{25}$$

The idea!

Enhance the quantum SVM method so it can be used with symmetry-aware kernels, making use of Fourier transform algorithms

- The symmetry group will the desirable one.
- Most of the processing should be done during the data preparation time

Conclusions

- Kernel based methods have a solid mathematical foundation from statistical learning theory
- Machine learning methods can largely benefit from symmetries on data as it is observable in the classical world. Nonetheless, methods are complex, both from the computation and conceptual point of view.
- Quantum computation seems capable of providing the necessary ingredients to the application of these methods: efficient Fourier transforms and SVM methods.
- It is not well-understood the impact of such an algorithm in learnability, or classification/training performance, but it seems a promising path
- To do: Develop the algorithm, implement it in a quantum programming language and test its performance.

References I

Srinivasan Arunachalam and Ronald de Wolf.
A survey of quantum learning theory.
arXiv preprint arXiv:1701.06806, 2017.

Robert Beals.
Quantum computation of fourier transforms over symmetric groups.
In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 48–53. ACM, 1997.

Michael Clausen and Ulrich Baum.
Fast fourier transforms for symmetric groups: Theory and implementation.
Mathematics of Computation, 61(204):833–847, 1993.

James W Cooley and John W Tukey.
An algorithm for the machine calculation of complex fourier series.
Mathematics of computation, 19(90):297–301, 1965.

References II

Mark Ettinger and Peter Høyer.
On quantum algorithms for noncommutative hidden subgroups.
Advances in Applied Mathematics, 25(3):239–251, 2000.

Robert Gens and Pedro M Domingos.
Deep symmetry networks.
In *Advances in neural information processing systems*, pages 2537–2545, 2014.

Lisa Hales and Sean Hallgren.
An improved quantum fourier transform algorithm and applications.
In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 515–525. IEEE, 2000.

References III

Sean Hallgren, Alexander Russell, and Amnon Ta-Shma.
Normal subgroup reconstruction and quantum computation using group representations.
In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 627–635. ACM, 2000.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola.
Kernel methods in machine learning.
The annals of statistics, pages 1171–1220, 2008.

Imre Risi Kondor.
Group theoretical methods in machine learning.
Columbia University, 2008.

Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd.
Quantum support vector machine for big data classification.
Physical review letters, 113(13):130503, 2014.

References IV

Peter W Shor.
Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.
SIAM journal on computing, 26(5):1484–1509, 1997.

Peter Wittek.
Quantum machine learning: what quantum computing means to data mining.
Academic Press, 2014.

Questions ?