

Document Number: P1911R0  
Date: 2019-10-07  
Authors: Michael Wong  
Project: Programming Language C++, SG19 Machine Learning  
Reply to: Michael Wong <michael@codeplay.com>

## **SG19: Machine Learning 2019/08/08-2019/09/11**

### **Contents**

Minutes for 2019/08/08 SG19 Conference Call .....	2
Minutes for 2019/09/11 SG19 Conference Call .....	7

## Minutes for 2019/08/08 SG19 Conference Call

### 1.1 Roll call of participants

>

Matthew Galati, Phil Ratzloff, Richard Dosselmann, Michael Wong, Andrew Lumsdaine, Javier Cabezas, Frank Seide, William Tambellini, Marco Foco, Vincent Reverdy, Mgirkar, fultz

### 1.2 Adopt agenda

>

> 1.3 Approve minutes from previous meeting, and approve publishing  
> previously approved minutes to ISOCPP.org <<http://isocpp.org>>

>

> 1.4 Action items from previous meetings

>

> 2. Main issues (125 min)

>

> 2.1 General logistics

> All C++ reflector are now moved to listserv

>

> <https://lists.isocpp.org/mailman/listinfo.cgi/sg19>

>

> 2.2 Paper reviews

>

> Review Cologne results

>

> 2.2.1: ML topics

> Differentiable Programing by Marco Foco

>

implemented library version of differentiable programming  
symbolic differentiation from a templated ast, now workign on forward and  
backward automatic differentiation

these are not good solutions because of the loss of the type system,  
because to inspect the ast, you have to declare functions as taking all  
auto parameters

for SG6, forward differentiatin needsimaginary numbers

Richard Dosselman

> Math proposal for Machine Learning

>

>

<https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit>

>

>

> Generally positive

use optional for return value to know if there was an empty set given  
on Median: hard to get consensus, I still want it to return a single value  
instead of a pair

iterators to start and end incase there is an even length and there are 2  
values to the median, Jens wanted a pair

more people seem to normally expect a value

LEWGI also seems to want a value

mean of photos may not be meaningful and this is about numerics

but it could be useful

also may want to mean and std dev of vectors

does opencv library have this for images? for mean and std dev

also means of tensors along any axis in cmtk with generic reduction  
operators

would be surprising if they are not there in opencv.

Should we use overloads to support beyond numerics

how an image different from a matrix? Its not, colour may have 3 or 4  
planes of numbers

how close are we getting to numpy? cmtk I try to put somethign close to  
numpy

similar interfaces on to pytorch, tensorflow

should mode allocate memory internally, should input be pre-sorted or not?

previously also want it to be pre-sorted liek other in stl library

also add to passing in the allocator

stddev has complex numbers and higher moments

also use less then comparator for sorts

from LA, if someone median mean mode, that require 4 passes over, can we  
have a combine function that would make a single pass

will this always give you numerically stable way when you use any floats

look at anything from std similar for guidance

boost accumulate allows you to combine and compose multiple ones

Graph Proposal for Machine Learning

>

>

<https://docs.google.com/document/d/13rdk1Xq8ZshUiTL5QASK1N2yD5bLwK3lQjbDs5yIF6o/edit>

>

>

> what is an adjacency array, which is a new thing, all the edges stored in a single vector,

note Sebastien Messmer's email post meeting on the definition open library vs closed framework,

successor of BGL I will share it and open it after cleaning it up

std library should just have a reference impl for adjacency array and list STL does really nicely in that container with interface and that may be important for graphs

yes that is a perfect example

taking input and next revision will focus more on the functions using iterators and ranges

and I like it better

also looking at concepts and ranges. What other description needed for range compliant and range friendly

descriptors are needed, but how is that different than an iterator? Good question, once you have properties, the descriptor is a way to retrieve a property on the edge or vertex

Peter Gottsclig mentioned that MTL has this idea of cursors as a property map which is similar to what is in Boost graph. This is different than what has been traditionally been done. there might be multiple properties per node. Have separate arrays for each property and this led us to the idea of vertex descriptor where a single cursor can dereference and access one single property

might offer solution in new BGL

D1416R1: SG19 - Linear Algebra for Data Science and Machine Learning

>

> <https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr>

>

>

> P1415: Machine Learning Layered list

>

>

> [https://docs.google.com/document/d/1elNFdIXWoetbxjO1OKol\\_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr](https://docs.google.com/document/d/1elNFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr)

>

>

Vincent working on tree extensions to graphs  
graph linear algebra

There is a property tree in boost if needed :

[https://www.boost.org/doc/libs/1\\_65\\_1/doc/html/property\\_tree.html](https://www.boost.org/doc/libs/1_65_1/doc/html/property_tree.html)

2.2.2 SG14 Linear Algebra progress:

> Different layers of proposal

>

>

[https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM\\_1Nb6oYAXIK\\_d0ljdUAtSQ/edit](https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_d0ljdUAtSQ/edit)

>

>

> 2.2.3 any other proposal for reviews?

>

> 2.3 Other Papers and proposals

>

> 2.5 Future F2F meetings:

>

> 2.6 future C++ Standard meetings:

> <https://isocpp.org/std/meetings-and-participation/upcoming-meetings>

>

>

> - \*2019-11-04 to 09: Belfast, Northern Ireland;\* Archer Yates

>

> -2020-02-10 to 15: Prague, Czech Republic

>

> - 2020-06-01 to 06: Bulgaria

> - 2020-11: (New York, tentative)

> - 2021-02-22 to 27: Kona, HI, USA

>

> 3. Any other business

>

> New reflector

>

> <http://lists.isocpp.org/mailman/listinfo.cgi/sg19>

>

> Old Reflector

> <https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19>

> <<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>>

>

> Code and proposal Staging area

>

> 4. Review

>

> 4.1 Review and approve resolutions and issues [e.g., changes to SG's

- > working draft]
- >
- > 4.2 Review action items (5 min)
- >
- > 5. Closing process
- >
- > 5.1 Establish next agenda
- >
- > TBD
- >
- > 5.2 Future meeting
- >
- > Aug 8
- > Sep 12
- > Oct 10
- > Nov 14 - cancelled due to DST change and switching to a new cycle.

## Minutes for 2019/09/11 SG19 Conference Call

### 1.1 Roll call of participants

>

Michael Wong, Matthew Galati, Phil Ratzloff, Richard Dosselmann, Marco Foco, Frank Seide, Ronan Keryell, Sebastian Messmer, Andrew Lumsdaine, William Tambellini

### > 1.2 Adopt agenda

>

Approve.

### > 1.3 Approve minutes from previous meeting, and approve publishing

> previously approved minutes to ISOCPP.org

>

Approve.

### > 1.4 Action items from previous meetings

>

### > 2. Main issues (125 min)

>

#### > 2.1 General logistics

> All C++ reflector are now moved to listserv

>

> <https://lists.isocpp.org/mailman/listinfo.cgi/sg19>

>

Belfast Mailing Oct 7. After our next call Oct 10.

>

### > 2.2 Paper reviews

>

> Review Belfast paper submission strategy

>

#### > 2.2.1: ML topics

> Differentiable Programing by Marco Foco

>

differentiable programming (library solution, + other things)

<https://github.com/mfoco/pan>

Looking at:

[https://github.com/mfoco/pan/blob/master/examples/forward\\_differentiation/main.cpp](https://github.com/mfoco/pan/blob/master/examples/forward_differentiation/main.cpp)

Richard Dosselman

> Math proposal for Machine Learning

>

>

<https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit>

>

> Compute mean/median mode in one pass over set

Boost.accumulate also does one pass over dataset, connected with Eric Niebler

Vincent also suggest the following:

I would suggest an option 4 relying on the range library and inspired by range adaptors (link

<view-source:[https://en.cppreference.com/w/cpp/ranges#Range\\_adaptors](https://en.cppreference.com/w/cpp/ranges#Range_adaptors)>):

provide composable function objects for stats so that one could write :

```
auto my_stats = mean | median | mode; // Composes a function object
std::vector v = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
auto results = mystats(v); // Runs my_stats on vector v
std::cout << results.mean << " " << results.median << " " << results.mode
<< std::endl; // Display results
```

Ranges compiler needed. Talk to Chris Di Bella

Range V3 may not require concepts and will work with GCC and clang, and VS recent versions

does not have composite object

<https://github.com/ericniebler/range-v3>

<https://en.cppreference.com/w/cpp/ranges>

How does it compose with multi dimensional tensors, like a multi dim iterators? Yes we want it to compose

Can it be generalized to generic reductions? Don't know

Unlikely to be at Belfast.

> Graph Proposal for Machine Learning

>

There is an updated Graph Proposal P1709R1 at

[https://docs.google.com/document/d/1QkfDzGyfNOKs86y053M0YHOLP6frzhTJqzg1Ug\\_vkkE/edit](https://docs.google.com/document/d/1QkfDzGyfNOKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit)

Rewritten for R1 as a functional interface, for accessing the graph  
added const steps as a starting point as well as type traits

Need some workign examples

New BGL17 could help with this

There are descriptors and property maps. Property is still new in LEWG, so  
it is being vetted.

[https://www.boost.org/doc/libs/1\\_71\\_0/libs/graph/doc/index.html](https://www.boost.org/doc/libs/1_71_0/libs/graph/doc/index.html)

Phil may be. Barry Hedquist who is HoD for US INCITS/ANSI

AI: andrew and Phil will get together

>

>

Also Tree Proposal for Machine Learning by Vincent

>

> D1416R1: SG19 - Linear Algebra for Data Science and Machine Learning

>

> <https://docs.google.com/document/d/1IKUNiUhBgRURW-UkspK7fAAyIhfXuMxjk7xKikK4Yp8/edit#heading=h.tj9hitg7dbtr>

>

> P1415: Machine Learning Layered list

>

>

[https://docs.google.com/document/d/1eINFdIXWoetbxjO1OKol\\_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr](https://docs.google.com/document/d/1eINFdIXWoetbxjO1OKol_Wj8fyi4Z4hogfj5tLVSj64/edit#heading=h.tj9hitg7dbtr)

>

> 2.2.2 SG14 Linear Algebra progress:

> Different layers of proposal

>

>

[https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM\\_1Nb6oYAXIK\\_d0ljdUAtSQ/edit](https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXIK_d0ljdUAtSQ/edit)

>

> 2.2.3 any other proposal for reviews?

>

> 2.3 Other Papers and proposals

>

> 2.5 Future F2F meetings:

>

Andrew Sebastian Michael at CPPCON next week.

Machine Learning at CPPCON 2020.

## 2.6 future C++ Standard meetings:

> <https://isocpp.org/std/meetings-and-participation/upcoming-meetings>

>

> - \*2019-11-04 to 09: Belfast, Northern Ireland;\* Archer Yates

>

> -2020-02-10 to 15: Prague, Czech Republic

>

> - 2020-06-01 to 06: Bulgaria

> - 2020-11: (New York, tentative)

> - 2021-02-22 to 27: Kona, HI, USA

>

> 3. Any other business

>

ISO on AI

<https://www.iso.org/committee/6794475.html>

> New reflector

>

> <http://lists.isocpp.org/mailman/listinfo.cgi/sg19>

>

> Old Reflector

> <https://groups.google.com/a/isocpp.org/forum/#!newtopic/sg19>

> <<https://groups.google.com/a/isocpp.org/forum/?fromgroups=#!forum/sg14>>

>

> Code and proposal Staging area

>

> 4. Review

>

> 4.1 Review and approve resolutions and issues [e.g., changes to SG's  
> working draft]

>

> 4.2 Review action items (5 min)

>

> 5. Closing process

>

> 5.1 Establish next agenda

>

> TBD

>

> 5.2 Future meeting

>

> Aug 8

> Sep 12

> Oct 10: Mailing deadline Oct 7, SC42 , tentative. Cancelled.

> Nov 14 - cancelled due to DST change and switching to a new cycle.



