Document Number: D2245R0
Date: 2020-10-25
Authors: Michael Wong
Project: Programming Language C++, SG19 Machine Learning
Reply to: Michael Wong <michael@codeplay.com>

SG19: Machine Learning Meeting Minutes 2020/04/09-2020/10/08

Contents

Minutes for 2020/04/09 SG19 Conference Call 2
Minutes for 2020/05/14 SG19 Conference Call 6
Minutes for 2020/06/11 SG19 Conference Call 11
Minutes for 2020/07/09 SG19 Conference Call 21
Minutes for 2020/08/13 SG19 Conference Call 38
Minutes for 2020/10/08 SG19 Conference Call 42

Minutes for 2020/04/09 SG19 Conference Call

> 1.1 Roll call of participants
>

Jens Maurer, Phil Ratzloff, Richard Dosselmann, Vincent Reverdy, Michael Wong, Eugenio

> 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
>
> Prague summary
>
> mailing deadlien every month on 15th
>
> all ISO meetings online until Nov
>

Apr 9, 2020 02:00 PM: stats paper
May 14, 2020 02:00 PM: Differential calculus
Jun 11, 2020 02:00 PM: Graph paper
Jul 9, 2020 02:00 PM: Stats paper
Aug 13, 2020 02:00 PM: Differential calculus
Sep 10, 2020 02:00 PM: Graph paper
Oct 8, 2020 02:00 PM: stats paper

> 2.2 Paper reviews
>
> 2.2.1: ML topics
>
> Richard Dosselman et al
>

> P1708R1: Math proposal for Machine Learning
>
>
https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit
>
std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
above is the stats paper that was reviewed in Prague
http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19

Revie Jolanta Polish feedback.
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html

VR gave report from SG6 and noted their feedback along with Lisa's report
above
one class for moments and one for quantiles
this could be logical and mathematical

>
> Phil Ratsloff et al
>
> P1709R1: Graph Proposal for Machine Learning
>
>
>
https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing
> <
>
>
https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE%2Fedit%3Fusp%3Dsharing&data=02%7C01%7CPhil.Ratzloff%40sas.com%7C729b2cf8502641e4ae5e08d749064578%7Cb1c14d5c362545b3a4309552373a0c2f%7C0%7C0%7C637058163592253027&sdata=4UQm8tqrcUbiZsr200UMrOaEModJYGNgP1oNot9PbAg%3D&reserved=0>
>
>
P1709R3:
https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing

working on implementation MSVC, BGL 17/lemon , met Intel people on Graph
Blas
compare and contrast what is in the paper and what is being one in the
state of the art library

graph used with simds
VR has used tachniqeus to flatten trees related to simd with graphs

Differentiable Programing by Marco Foco
>
Aimed for next call.

> P1416R1: 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
>
> 2.2.2 SG14 Linear Algebra progress:
> Different layers of proposal
>
>
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_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
>
> -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
>
> May 14, 2020 02:00 PM
> Jun 11, 2020 02:00 PM
> Jul 9, 2020 02:00 PM
> Aug 13, 2020 02:00 PM
> Sep 10, 2020 02:00 PM
> Oct 8, 2020 02:00 PM

Minutes for 2020/05/14 SG19 Conference Call

> 1.1 Roll call of participants
>

Andrew Lumsdaine, Phil Ratzloff, luke D'alessandro, Marco Foco, Richard Dosselmann, Scott McMillan, Michael Wong, Michael Chiu, Kevin Deweese

1.2 Adopt agenda
>
Approve

> 1.3 Approve minutes from previous meeting, and approve publishing
> previously approved minutes to ISOCPP.org
>
> 1.4 Action items from previous meetings
>
> 2. Main issues (125 min)
>
> 2.1 General logistics
>
> Meeting plan, focus on one paper per meeting but does not preclude other
> paper updates:
>
> Apr 9, 2020 02:00 PM: stats paper- DONE
> May 14, 2020 02:00 PM: Stats paper replaces Differential calculus
> Jun 11, 2020 02:00 PM: Graph paper
> Jul 9, 2020 02:00 PM: Stats paper
> Aug 13, 2020 02:00 PM: Differential calculus
> Sep 10, 2020 02:00 PM: Graph paper
> Oct 8, 2020 02:00 PM: stats paper
>
>
>
> all ISO meetings online untilend of Aug
>

monthly 15th is a deadline

2.2 Paper reviews
>
> 2.2.1: ML topics
>
> Richard Dosselman et al

>
> P1708R1: Math proposal for Machine Learning
>
>
https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit
>
> > std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
> > above is the stats paper that was reviewed in Prague
> > http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
> >
> > Revie Jolanta Polish feedback.
> > http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html
>

Freestanding, or accumulator object? Have done both
series of weight suggested by Michael C
moved to ranges projection as suggested by Jens M
this now looks more compact

latest versoin of python has geometric + harmonic mean following geometric mean
now we can compact all calculation of avg down to 6 , now we are on par with with python has

Yolanta's paper pointed out notion of trimmed mean, implies it is sorted
nice compromise so we don't have to take on the trimmed stats
Luke A: have you considered less strenous restriction of input range as long as you already have forward range?
yes I can change that as I thought foward range was more restricted, I will change that

PR: selection between population and sample, better to use enumeration population vs sample? OK
MF: or do both for when you need to switch to a projection? OK so switch order of projection and population

Quantile as a replacement for median, so you can ask for 0.25 instead of strictly median

have a random_access_range for the sorted case to find the middle position

for mode we can customize with your own allocator, customize what it means to be equal, makes use of output iterator so you can return multiple mode
LA: may want to commit all into ranges, so return mode as a range vs output iterator (view, allowing take_n), so return all of them and not just first
strange that python decided to only return one

PhilR: i was advised too to use output iterator
LA: range's model is composability

PhilR: Also Macro to implementation

accumulator object was more difficult to get right
liek a vehicle that stops every mile, and accumulate it at every mile, user just say they want mean or variance, then pass the mean to this function then you can pull the stats out of these objects; idea is to fuse multipel operations into a single path
so in the end you can do it all in one pass

also have sig variance for a weighted mean

LA: use forwardign references && vs & aggressively when dealing with ranges

Invite Yolanta, Walter, Eric, and Lisa for July meeting.

>
> Phil Ratsloff et al
>
> P1709R1: Graph Proposal for Machine Learning
>
>
> P1709R3:
>
>
https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing
>
>
>
>
https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing
> <
>
>
https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE%2Fedit%3Fusp%3Dsharing&data=02%7C01%7CPhil.Ratzloff%40sas.com%7C729b2cf8502641e4ae5e08d749064578%7Cb1c14d5c362545b3a430
>
9552373a0c2f%7C0%7C0%7C637058163592253027&sdata=4UQm8tqrcUbiZsr200UMrOaEModJYGNgP1oNot9PbAg%3D&reserved=0>
>

> Differentiable Programing by Marco Foco
>
> P1416R1: 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
>
> 2.2.2 SG14 Linear Algebra progress:
> Different layers of proposal
>
>
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_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
>
> -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
>
>
> Jun 11, 2020 02:00 PM
> Jul 9, 2020 02:00 PM
> Aug 13, 2020 02:00 PM
> Sep 10, 2020 02:00 PM
> Oct 8, 2020 02:00 PM

Minutes for 2020/06/11 SG19 Conference Call

Meeting notes.

Hi all, I am sorry my power died. Phil, I heard at the end that you might
still have a few things to do on the paper but it seems largely done.
I like to see if we can get that done for the next call in July, and
possibly vote on it in SG19 to pass it to LEWG.

I like to correct that the next call in July will be on Stats paper
review. The Aug call will be on RL and AD.
So assuming that is possible, I have adjusted the schedule but that too can
be changed.

Thanks all.

On Wed, Jun 10, 2020 at 12:50 PM Michael Wong <fraggamuffin_at_[hidden]>
wrote:

> SG19 Machine Learning 2 hours
> Hi,
>
> Michael Wong is inviting you to a scheduled Zoom meeting.
>
> Topic: SG19 monthly Apr 2020-Oct 2020
> Time: 02:00 PM Eastern Time (US and Canada) 18:00 UTC
> Every month on the Second Thu, until Oct 8, 2020, 7 occurrence(s)
> Apr 9, 2020 02:00 PM 18:00 UTC
> May 14, 2020 02:00 PM 18:00 UTC
> Jun 11, 2020 02:00 PM 18:00 UTC
> Jul 9, 2020 02:00 PM 18:00 UTC
> Aug 13, 2020 02:00 PM 18:00 UTC
> Sep 10, 2020 02:00 PM 18:00 UTC
> Oct 8, 2020 02:00 PM 18:00 UTC
> Please download and import the following iCalendar (.ics) files to
> your
> calendar system.
> Monthly:
>
>
https://iso.zoom.us/meeting/v50sceqopj4pyLdu5Mx1orYgnZZUj0RNqw/ics?icsToken=98tyKuuhrz0pGtyQs1-CArUqE53ibvG1kmhirrYIsQe0DDJqZQ3MDNdIYoBRAc-B

>
> Join from PC, Mac, Linux, iOS or Android:
> https://iso.zoom.us/j/291630853?pwd=WUlKbS9SNFNRa0QyWXRWenlGSDhaQT09
> Password: 339768
>
> Or iPhone one-tap :
> US: +14086380968,,291630853# or +16468769923,,291630853#
> Or Telephone:
> Dial(for higher quality, dial a number based on your current
> location):
> US: +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1
> 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799
> or 877 853 5247 (Toll Free)
> Meeting ID: 291 630 853
> Password: 339768
> International numbers available: https://iso.zoom.us/u/abhaIjFKLZ
>
> Or Skype for Business (Lync):
> https://iso.zoom.us/skype/291630853
>
> Agenda:
>
> 1. Opening and introductions
>
> 1.1 Roll call of participants
>

Michael Wong, Richard Dosselman, Phil Ratzloff, Jorge Silva, Larry Lewis,
Kevin Dewessee, Scott McMllan, ANdrew Lumsdaie, Jesun Firoz, Marco Foco

> 1.2 Adopt agenda
>

Yes

> 1.3 Approve minutes from previous meeting, and approve publishing
> previously approved minutes to ISOCPP.org
>
> 1.4 Action items from previous meetings
>
> 2. Main issues (125 min)
>
> 2.1 General logistics
>
> Meeting plan, focus on one paper per meeting but does not preclude other
> paper updates:
>
> Apr 9, 2020 02:00 PM: stats paper- DONE

> May 14, 2020 02:00 PM: Stats paper replaces Differential calculus DONE
> Jun 11, 2020 02:00 PM: Graph paper-
> Jul 9, 2020 02:00 PM: Stats paper + Graph paper vote
> Aug 13, 2020 02:00 PM: Differential calculus + Reinforcement Learning
> Sep 10, 2020 02:00 PM: Graph paper +stats paper
> Oct 8, 2020 02:00 PM: Differential calculus + Reinforcement Learning
>
> ISO meeting status
>

No meeting until end of year, this year deep dive on each ML topics
Papers can still move through using the online meetings for EWG, LEWG, though there is no decision made online, just tentative decisions

> CPPCON status
>

Will happen in hybrid form
Phil Submitted proposal for Cppcon

> 2.2 Paper reviews
>
> 2.2.1: ML topics
>
> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html> Larry
> Lewis Jorge Silva
>
> Reinforcement Learning proposal:
>

RL within SAS
provide guidance and an API for RL, optimizers
build on top of generic machine learning
independent of vendors, but follow pytorch, tensorflow
dependent of the underlying tensor
core RL algorithms, a ton of research, it is newer then deep learning
some algos will be distributed, get feedback fro mthe group doign tensors and LA
will you need some of the underlying facilities like Supervised and unsupervised ML? Yes
will you base this on certain libraries? we are most familiar with pytorch, tensorflow when it becomes more functional, we will pick that up
will you use GPUs? yes pytorch is transparent in that respect
for deep learning, we don't use pytorch, just for RL, I don't want to be just dependent on pytorch
hard part can be design phase for ISO C++ which can take a lot more time then you think
as C++ moves, we have to adjust the design to match the new style C++20, 23, 23, we started with data structures, and switched to functions for stats

GPU is where performance come from; yes for future we will talk about
parallel ranges and SYCL

what about automatic differentiation? for a tensor we do need both GPU and
AD - this helps with back propagation
for AD: do you build a network of optimizers? yes plan to reuse work from
other teams; have implemented NN too many times
trying hard to make AD that works for everybody? library or language
SG7 reflection had a lot of polarization on this topic, one side say it
should be language, another side says it should be library; but we need
code introspection, and generate from the code
we dont want to standardize entire AST of C++

pytorch differentiates with AD and forward differentiation meant you could
not single step forward in the code, but pytorch can do that

> Phil Ratsloff et al
>
> P1709R1: Graph Proposal for Machine Learning
>
> P1709R3:
>
>
https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing
>
>
>
https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing
> <
>
>
https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE%2Fedit%3Fusp%3Dsharing&data=02%7C01%7CPhil.Ratzloff%40sas.com%7C729b2cf8502641e4ae5e08d749064578%7Cb1c14d5c362545b3a430
>
9552373a0c2f%7C0%7C0%7C637058163592253027&sdata=4UQm8tqrcUbiZsr200UMrOaEModJYGNgP1oNot9PbAg%3D&reserved=0>
>
>
> Iâ€™ve been working on the prototype implementation to get it building in
> both Windows & Linux, using CMake & the Conan package manager:
>
> 1. All unit tests complete successfully for both MSVC & gcc10

> 2. All bgl17 code has been removed from the repository. It uses a
> cloned bgl17 directory (ENABLE_BGL17 cmake option).
> 3. Catch2 is now being used instead of Google Test for unit testing
> 4. A simple unit test demonstrates the use of the libraryâ€™s
> *dfs_vertex_range* iteration *using bgl17â€™s vov graph*. This can be
> seen in test/test_vov_adaptor.cpp.
> 1. There were a few changes needed in bgl17 to accommodate this (I
> havenâ€™t pushed these changes)
>
> i. I
> added an inner_container type definition to vov
>
> ii. There
> were 3 places where I added #ifdef _*MSC*_VER to disable linux-specific
> code, far fewer than before.
>
> 1. Adapting vov requires the following
>
> i. An
> adaptor graph class to map the vov types to expected types
>
> ii. Function
> overloads that uses the adaptor graph class as a template argument
>
> 1. Added graph API functions to avoid name ambiguity with begin(g) &
> end(g) for vertices in the dfs & bfs range iterators.
> 1. vertex_begin(g), vertex_end(g)
> 2. edge_begin(g,u), edge_end(g,u)
>
>
>
> I havenâ€™t written the code to support value(uv) function to get edge
> properties for vov yet.
>
> These changes should bring the library much closer to a repeatable
> cross-platform build and youâ€™re welcome to try it.
>
> Iâ€™ve pushed the code to the master branch at
> https://github.com/pratzl/graph
>
>
>
> The next SG19 meeting is 6/11/20 (12d from now) and I have some things in
> mind to work on. Iâ€™ve been focused on the prototype to make it more
> accessible for all the authors and I need to switch back to the paper and
> give it more attention.

>
> 1. Paper
> 1. Complete algorithm descriptions & examples:
>
> i. Connected
> Components
>
> ii. Strongly
> Connected components
>
> iii. Bi-connected
> Components
>
> iv. Articulation
> Points
>
> 1. Data structures
>
> i. Add
> section on graph adaptors
>
> 1. algorithm implementations
> 1. connected & strongly connected components unit tests
> 2. [bi-connected components]
> 3. [articulation points]
> 2. bgl17 adaptors
> 1. vov adaptor: implement value(edge), add dfs_edge_range tests
> 2. implement a compressed adaptor
> 3. other prototype features
> 1. Support Clang10 using the range-v3 concepts macros
> 4. Documentation
> 1. Add explicit description of how to install and use the library
>
>
> 90-95% done, major sections there, need examples

prototype email library works on linux and windows
using cmake conan, unit test framework

all algo have iterators and can also take range
output_iterator concept added requires output iterator, might be Richard
want to use that

added vertex begin and end to allow me to want the graph based one

can iterate through graph, have begin and end, with starting vertex, can
also construct one with a range

a section on graph data structure has been rewritten, carried from the beginning, reflect what I have in my prototype
have classes with common template types, 3 types for user values
there is an index type which is either 32 or 16 bit value
default of 32 bit is most of the case
there is the allocator
a section on what kind of user-defined type for weighs for example in an adjacency list

compressed has been changed to direct_adjacency_array to compliments undirected

can define properties for a graph and edge/vertex
only user-defined property can be changed after ts been constructed
other constraint is source edges has to be ordered by vertex key
which is DAA graph, and is a template alias with various defaults
have various classes to implement this
access id defined by public function

possible to customize by overriding

Do I need all the constant types?
Think yes, both const and non-const
also think so, we did that withBGL17 as well, else its annoying
I need to prove it for my self though I also think so
a lot of boilerplate stuff to make all this work; yes can shortcut with enable_if as well

one interesting constructor that takes a range of edges and vertexes, extracts key from edge from edge range, just a pair of vertex keys, another one extract fn property and fn property
now I see I also need a way to specify the graph

need to revisit to see if I need to reimplement this for the test

then there is undirected adjacency list

assert there is one object per edge and is part of 2 linked lists
edges are in doubly linked list, stored in a vector, after construction
can't add vertexes or edges
inedges are ordered by vertex key

everythign else is similar

finally a section to adapt to external graph, adapt algo here to their data structures

we can define our own graph type to overeride the graph type to do the right thing, but also do that with types as well
I tested my own BFS algo with BGL17 data structure; yes have small things in there that can allow that to happen starting with a graph so I think this is the right approach

Jesun asked Is it a hard requirement that the vertices will be in a vector and edges in a linked list for undirected graph (and probably for directed graph type)? Any implication on iterating over them in terms of performance as well as mutability of the graph?
Good question, this is something I like to explore; have a key that i want to access in a different way to enable conditional algo, where you dont have that requirement
I like to relax that area more so I have not changed any concepts from what we have before
concept used reflects that algorithm; should still have something less restrictive; yes probably right should be able to do iteration, on neighbors, randomly access container of forward terable containers

Walking through code TDD style
as I am doing development, it is outputing results
at top of file I set the Test option using the German routes

I like what you did with BGL17 testing ... yes it was with tuples of ranges

dont have topological sort done yet

interface is stable
strongly connected components not tested

what about initializers? I have a few classes that will generate graph for me but can be improved as it is not repeatable; yes we have file i/o for matrix, graph =(){} convenient thingfor testing
could I do it with what I have now or do I need a constructor but will look at whatBGL17 have too
should be doable but have to go back to see where we put that in things to do:
1. fns I have not implemented
2. BGL17 compressed graph
3. range support sentinels
4. reverse filter for a graph?
creating a NN with weighs is a common thing and eliminates an edge; yes that kind of filter is useful for NN
comparison with other libraries - put out a separate paper
can we store a graph in some constexpr arrays

marshalling,
relax constraints on algos to make them more flexible

< my power died at this point>

aiming for guidance on moving this paper forward.

Richard Dosselman et al
>
> P1708R1: Math proposal for Machine Learning
>
>
https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit
>
> > std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
> > above is the stats paper that was reviewed in Prague
> > http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
> >
> > Review Jolanta Polish feedback.
> > http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html
>
>
>
> Richard persents
large number of revision now replace iterator pairs with ranges
free standing function presents linear pass over the data
for large data sets, then we have accumulator objects to make one combined pass and compute final stats at the end
alternate predicate, if you want to retrieve one value out of array of structures

for each mean, have overloads
MC suggested we have a weighted mean

Added execution policy for parallelization
gives you 4 variations of each of the means

we also have geometric and harmonic means with 4 flavors each

variance also follows but makes clear working with population vs sample

passing 2 ranges - 1 from value and one from range
can you pass just one (zipping of 2 ranges together and extracting the projection) if the 2 coincide with each other
OK, I might move in that direction, will think about it

replace median with general quantile

mode has a comparator for equality

python only returns 1st mode, but I will return all the modes

makes one linear pass through each data structure,
but can also allow single pass to compute it all
using accumulated weights which also have weighted and unweighted version
of each of the mean median and mode
this allows one single linear pass for all these data structures

mode can return a series of values and can handle non-numerical data

moving to documentation now

for normal distribution, can you have a parameter that defaults to normal?
for a statistician, Poisson distribution, arrival time,
whether a mean is a good moment to calculate, sample mean are good
estimators,
continue this on reflector

> Differentiable Programing by Marco Foco
>
> P1416R1: 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
>
> 2.2.2 SG14 Linear Algebra progress:
> Different layers of proposal
>
>
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_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
>
> -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
>
> Jul 9, 2020 02:00 PM
> Aug 13, 2020 02:00 PM
> Sep 10, 2020 02:00 PM
> Oct 8, 2020 02:00 PM

Minutes for 2020/07/09 SG19 Conference Call

Meeting notes.

> 1.1 Roll call of participants
>

Richard Dosselmann, Paul Preeny, Antony Peacock, Bran Erfani, Inbal Levi, Jessum Firoz, Jolanta Opara, Kevin Deweese, Larry Lewis, Michael Chiu, Phil Ratzloff, Rene Morell, Scott McMillan, Walter Brown, Michael Wong, Paul Preney

> 1.2 Adopt agenda
>

Approve. Phil and Michael Minute takers

> 1.3 Approve minutes from previous meeting, and approve publishing
> previously approved minutes to ISOCPP.org
>
> 1.4 Action items from previous meetings
>
> 2. Main issues (125 min)
>
> 2.1 General logistics
>
> Meeting plan, focus on one paper per meeting but does not preclude other
> paper updates:
>
> Apr 9, 2020 02:00 PM: stats paper- DONE
> May 14, 2020 02:00 PM: Stats paper replaces Differential calculus DONE
> Jun 11, 2020 02:00 PM: Graph paper- DONE
> Jul 9, 2020 02:00 PM: Stats paper
> Aug 13, 2020 02:00 PM: Differential calculus + Reinforcement Learning
> Sep 10, 2020 02:00 PM: Graph paper
> Oct 8, 2020 02:00 PM: stats paper
>
> ISO meeting status
>
> CPPCON status
>
> future C++ Std meetings
>
> 2.2 Paper reviews
>
> 2.2.1: ML topics
>
> 2.2.1.1 Stats review Richard Dosselman et al
>
> P1708R1: Math proposal for Machine Learning

>
>
https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit
>
> > std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
> > above is the stats paper that was reviewed in Prague
> > http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
> >
> > Review Jolanta Polish feedback.
> > http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html
>
> Preview attachment stats feedback.docx
> <https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671130344765626535&th=17310c31cd2198a7&view=att&disp=safe>
> stats feedback.docx
> 15 KB
>
> <https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671130344765626535&th=17310c31cd2198a7&view=att&disp=safe>
>
>
>
>
>
>
> Preview attachment stats.pdf
> <https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671397124497524079&th=1731fed44c48c56f&view=att&disp=safe>
> stats.pdf
> 224 KB
>
> <https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671397124497524079&th=1731fed44c48c56f&view=att&disp=safe>
>
>
>
>
>
>
>
> 2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva
>
> Reinforcement Learning proposal:
>
> 2.2.1.3 Graph Proposal Phil Ratsloff et al

>
> P1709R1: Graph Proposal for Machine Learning
>
> P1709R3:
>
>
https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing
>
>
>
https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing
> <
>
>
https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE%2Fedit%3Fusp%3Dsharing&data=02%7C01%7CPhil.Ratzloff%40sas.com%7C729b2cf8502641e4ae5e08d749064578%7Cb1c14d5c362545b3a430
>
9552373a0c2f%7C0%7C0%7C637058163592253027&sdata=4UQm8tqrcUbiZsr200UMrOaEModJYGNgP1oNot9PbAg%3D&reserved=0>
>
>
> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>
>
> 2.2.1.4: Differentiable Programing by Marco Foco
>
>
>
<https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit>
>
> 2.2.3 any other proposal for reviews?
>
> 2.3 Other Papers and proposals
>
> P1416R1: 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
>
> 2.2.2 SG14 Linear Algebra progress:
> Different layers of proposal
>
>
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit
>
> 2.5 Future F2F meetings:
>
> 2.6 future C++ Standard meetings:
> https://isocpp.org/std/meetings-and-participation/upcoming-meetings
>
> -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
>
> Jul 9, 2020 02:00 PM
> Aug 13, 2020 02:00 PM
> Sep 10, 2020 02:00 PM
> Oct 8, 2020 02:00 PM

Here are the rest of the minutes. Thanks to Phil Ratzloff.

ATTENDEES

Michael Wong, Richard Dosselmann, Antony Peacock, Baran Erfani, Inbal Levi, Jesun Firos, Jolanta Opara, Kevin Deweese, Larry Lewis, Michael Chiu, Paul Preney, Rene Morell, Scott McMillan, Walter Brown

LOGISTICS

ISO meetings canceled thru Dec

CppCon - 80% online, 20% physical

C++20 â€“ accepted in Prague; national ballots just received; waiting to work thru ISO processes

STATUS

Reinforcement Learning

LL: determining underlying technologies (e.g. tensor, neural nets)

Graph

PR: No significant change

Auto-Differentiation

(no report)

STATISTICS (P1708) Review

RD: (Overview)

RD: (Intro)

WB

Why these functions, and not others (e.g. skewness, kertosis)?

MW

Plan: layered delivery (simple), followed by others like kertosis

Should they be delivered together? Separately?

RD

Delivering similar functionality as other language (esp. Python)

Based on existing literature

WB

Look for existing ISO paper for statistics?

PR

How to find them?

WB

ISO web site. Write to them to request relevant paper(s).

MW

Iâ€™ll take that on.

SG6 (Numerics) may add other perspectives & requirements beyond matching other languages (hence involvement of WB)

JO

Suggest simple function median() to call quantile with 0.5

I think itâ€™s counter-intuitive to have default value for quantile of 0.5

JO

Not mentioned; general weights should be non-negative

geometric_mean: should negative values be allowed (tricky); should be stated explicitly harmonic_mean: (same)

will send you notes

JO

quantile: the sum of weights is 1.0. Not identified in mean

JO

Pop variance (7): exponent should be outside parenthesis

MD

Keep stats function in <stats>

WB

Define why you want them in their own header

Random functionality is in <random>

Special math functions were put <cmath>

Context: <review mean example>

WB

Design slightly problematic

Result=double is questionable; More effort to eval the type (e.g. from input_range R)

Result template arg should come first (usability)

PR

Agree with WB

Suggest return type of auto

MC

Whatâ€™s the justification of these functions, in comparison to using lambdaâ€™s

RD

To provide a straight-forward implementation of statistical functions, similar to other languages.

MC

We have an opportunity to handle more specific implementations.

The current implementations imply naÃ¯ve implementations and other features may be important to be supported.

tag_invoke [should be looked at to support different, useful implementations]

WB

P1895

IL

might be relevant (?)

https://www.iso.org/obp/ui/#iso:std:iso:3534:-1:ed-2:v2:en

(Statistics â€” Vocabulary and symbols)

RM

tag_invoke.. https://wg21.link/p1895

Also.. I wrote an isolate more recent application of tag_invoke here.. https://github.com/bfgroup/duck_invoke

Context: <var func>

JO

NaN values: What is NaN?

WB

quiet NaN or signalling NaN

JO

Population vs sample: Which is more frequency used

MC

I think you should just have one variance and pass a weight. (weight instead of choice)

Context: sorted_quantile, unsorted_quantile

JO

sorted_ and unsorted_ are strange names:

median is quantile, quantile is not median: suggest median (helper) function

we always have 2 numbers; would like to have policy to choose one number

MC

When using quantile is used for text, then it doesnâ€™t make sense to return average

RD

unsorted_quantile has lower restrictions on type of range

WB

What about the concept of a sorted_input_range?

Allow overload resolution to select the proper function

PP

Examples might be simpler using structured bindings instead of get<n>()

MC

Disagree with handling of text

Nothing to do with statistics

MD

SG19 feedback was that it was almost a side benefit.

We should support it if it made sense and users find it useful.

JO

I would like to see in quantile a list of quantiles I can specify

MD

You want to specify multiple quantiles?

JO

Yes

Context: <mode>

Context: <accumulator objects>

LL

Is there a way for someone to do a kahan summation?

MC

Why is mean in an accumulator a function? Why not just a member value?

WB

That requires some thought. There are a number of values involved.

MD

Mode returns a collection, not a single value

PP

do we want to return a range?

Concepts might help clean things up. Would that be helpful?

MD

[Where are] kurtosis & skewness [used]?

MC

Theyâ€™re in Matlab

MW

Can they be useful outside of machine learning?

What other languages?

Where do we stop?

MC

Are there plans to have an implementation?

Boost proposal?

MD

I have a quick-and-dirty implementation

MC

It would be helpful to propose it to boost

On Thu, Jul 9, 2020 at 2:21 PM Michael Wong <fraggamuffin_at_[hidden]> wrote:

>
>
> On Wed, Jul 8, 2020 at 9:43 PM Michael Wong <fraggamuffin_at_[hidden]>
> wrote:
>
>> SG19 Machine Learning 2 hours. This session will focus on the Stats paper
>> but can also update other topics.
>>
>> Hi,
>>
>> Michael Wong is inviting you to a scheduled Zoom meeting.
>>
>> Topic: SG19 monthly Apr 2020-Oct 2020
>> Time: 18:00 UTC (2:00pm Eastern Time US and Canada)
>> Every month on the Second Thu, until Oct 8, 2020, 7 occurrence(s)
>> Apr 9, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
>> May 14, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
>> Jun 11, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
>> Jul 9, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
>> Aug 13, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)

>> Sep 10, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
>> Oct 8, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
>> Please download and import the following iCalendar (.ics) files to
>> your
>> calendar system.
>> Monthly:
>>
>>
https://iso.zoom.us/meeting/v50sceqopj4pyLdu5Mx1orYgnZZUj0RNqw/ics?icsToken=98tyKuuhrz0pGtyQs1-CArUqE53ibvG1kmhirrYIsQe0DDJqZQ3MDNdIYoBRAc-B
>>
>> Join from PC, Mac, Linux, iOS or Android:
>> https://iso.zoom.us/j/291630853?pwd=WUlKbS9SNFNRa0QyWXRWenlGSDhaQT09
>> Password: 339768
>>
>> Or iPhone one-tap :
>> US: +14086380968,,291630853# or +16468769923,,291630853#
>> Or Telephone:
>> Dial(for higher quality, dial a number based on your current
>> location):
>> US: +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1
>> 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799
>> or 877 853 5247 (Toll Free)
>> Meeting ID: 291 630 853
>> Password: 339768
>> International numbers available: https://iso.zoom.us/u/abhaIjFKLZ
>>
>> Or Skype for Business (Lync):
>> https://iso.zoom.us/skype/291630853
>>
>> Agenda:
>>
>> 1. Opening and introductions
>>
>> The ISO Code of conduct:
>> https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf
>> The IEC Code of Conduct:
>> https://basecamp.iec.ch/download/iec-code-of-conduct-for-delegates-and-experts/
>> The WG21 Practices and Procedures and Code of Conduct:
>> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures
>>
>> 1.1 Roll call of participants
>>
> Richard Dosselmann, Paul Preeny, Antony Peacock, Bran Erfani, Inbal Levi,
> Jessum Firoz, Jolanta Opara, Kevin Deweese, Larry Lewis, Michael Chiu, Phil
> Ratzloff, Rene Morell, Scott McMillan, Walter Brown, Michael Wong, Paul

> Preney
>
>> 1.2 Adopt agenda
>>
> Approve. Phil and Michael Minute takers
>
>> 1.3 Approve minutes from previous meeting, and approve publishing
>> previously approved minutes to ISOCPP.org
>>
>> 1.4 Action items from previous meetings
>>
>> 2. Main issues (125 min)
>>
>> 2.1 General logistics
>>
>> Meeting plan, focus on one paper per meeting but does not preclude other
>> paper updates:
>>
>> Apr 9, 2020 02:00 PM: stats paper- DONE
>> May 14, 2020 02:00 PM: Stats paper replaces Differential calculus
>> DONE
>> Jun 11, 2020 02:00 PM: Graph paper- DONE
>> Jul 9, 2020 02:00 PM: Stats paper
>> Aug 13, 2020 02:00 PM: Differential calculus + Reinforcement Learning
>> Sep 10, 2020 02:00 PM: Graph paper
>> Oct 8, 2020 02:00 PM: stats paper
>>
>> ISO meeting status
>>
>> CPPCON status
>>
>> future C++ Std meetings
>>
>> 2.2 Paper reviews
>>
>> 2.2.1: ML topics
>>
>> 2.2.1.1 Stats review Richard Dosselman et al
>>
>> P1708R1: Math proposal for Machine Learning
>>
>>
https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit
>>
>> > std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2

>> > above is the stats paper that was reviewed in Prague
>> > http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
>> >
>> > Review Jolanta Polish feedback.
>> > http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html
>>
>> Preview attachment stats feedback.docx
>> <https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671130344765626535&th=17310c31cd2198a7&view=att&disp=safe>
>> stats feedback.docx
>> 15 KB
>>
>> <https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671130344765626535&th=17310c31cd2198a7&view=att&disp=safe>
>>
>>
>>
>>
>>
>>
>> Preview attachment stats.pdf
>> <https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671397124497524079&th=1731fed44c48c56f&view=att&disp=safe>
>> stats.pdf
>> 224 KB
>>
>> <https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671397124497524079&th=1731fed44c48c56f&view=att&disp=safe>
>>
>>
>>
>>
>>
>>
>>
>> 2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva
>>
>> Reinforcement Learning proposal:
>>
>> 2.2.1.3 Graph Proposal Phil Ratsloff et al
>>
>> P1709R1: Graph Proposal for Machine Learning
>>
>> P1709R3:
>>
>>

https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing
>>
>>
>>
https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing
>> <
>>
>>
https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE%2Fedit%3Fusp%3Dsharing&data=02%7C01%7CPhil.Ratzloff%40sas.com%7C729b2cf8502641e4ae5e08d749064578%7Cb1c14d5c362545b3a430
>>
9552373a0c2f%7C0%7C0%7C637058163592253027&sdata=4UQm8tqrcUbiZsr200UMrOaEModJYGNgP1oNot9PbAg%3D&reserved=0>
>>
>>
>> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>
>>
>> 2.2.1.4: Differentiable Programing by Marco Foco
>>
>>
>>
<https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit>
>>
>> 2.2.3 any other proposal for reviews?
>>
>> 2.3 Other Papers and proposals
>>
>> P1416R1: 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
>>
>> 2.2.2 SG14 Linear Algebra progress:
>> Different layers of proposal
>>

>>
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit
>>
>> 2.5 Future F2F meetings:
>>
>> 2.6 future C++ Standard meetings:
>> https://isocpp.org/std/meetings-and-participation/upcoming-meetings
>>
>> -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
>>
>> Jul 9, 2020 02:00 PM
>> Aug 13, 2020 02:00 PM
>> Sep 10, 2020 02:00 PM
>> Oct 8, 2020 02:00 PM

Minutes for 2020/08/13 SG19 Conference Call

Meeting notes.

SG19 Machine Learning 2 hours. This session will focus on Reinforcement learning and differential calculus but with updates from Stats and Graph proposals.

Hi,

Michael Wong is inviting you to a scheduled Zoom meeting.

Topic: SG19 monthly Apr 2020-Oct 2020
Time: 18:00 UTC (2:00pm Eastern Time US and Canada)
Every month on the Second Thu, until Oct 8, 2020, 7 occurrence(s)
Apr 9, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
May 14, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
Jun 11, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
Jul 9, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
Aug 13, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
Sep 10, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
Oct 8, 2020 18:00 UTC (2:00pm Eastern Time US and Canada)
Please download and import the following iCalendar (.ics) files to your calendar system.
Monthly:
https://iso.zoom.us/meeting/v50sceqopj4pyLdu5Mx1orYgnZZUj0RNqw/ics?icsToken=98tyKuuhrz0pGtyQs1-CArUqE53ibvG1kmhirrYIsQe0DDJqZQ3MDNdIYoBRAc-B

Join from PC, Mac, Linux, iOS or Android:
https://iso.zoom.us/j/291630853?pwd=WUlKbS9SNFNRa0QyWXRWenlGSDhaQT09
Password: 339768

Or iPhone one-tap :
US: +14086380968,,291630853# or +16468769923,,291630853#
Or Telephone:
Dial(for higher quality, dial a number based on your current location):
US: +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833 or +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1 346 248 7799 or 877 853 5247 (Toll Free)
Meeting ID: 291 630 853
Password: 339768
International numbers available: https://iso.zoom.us/u/abhaIjFKLZ

Or Skype for Business (Lync):
 https://iso.zoom.us/skype/291630853

Agenda:

1. Opening and introductions

The ISO Code of conduct:
https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100397.pdf
The IEC Code of Conduct:
https://basecamp.iec.ch/download/iec-code-of-conduct-for-delegates-and-experts/
The WG21 Practices and Procedures and Code of Conduct:
https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures

1.1 Roll call of participants

1.2 Adopt agenda

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

1.4 Action items from previous meetings

2. Main issues (125 min)

2.1 General logistics

Meeting plan, focus on one paper per meeting but does not preclude other paper updates:

Apr 9, 2020 02:00 PM: stats paper- DONE
May 14, 2020 02:00 PM: Stats paper replaces Differential calculus DONE
Jun 11, 2020 02:00 PM: Graph paper- DONE
Jul 9, 2020 02:00 PM: Stats paper -DONE
Aug 13, 2020 02:00 PM: Differential calculus + Reinforcement Learning
Sep 10, 2020 02:00 PM: Graph paper
Oct 8, 2020 02:00 PM: stats paper

ISO meeting status

CPPCON status

future C++ Std meetings

2.2 Paper reviews

2.2.1: ML topics

2.2.1.1 Stats review Richard Dosselman et al

P1708R1: Math proposal for Machine Learning
https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit

> std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
> above is the stats paper that was reviewed in Prague
> http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
>
> Review Jolanta Polish feedback.
> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html

Preview attachment stats feedback.docx
<
https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671130344765626535&th=17310c31cd2198a7&view=att&disp=safe
>
stats feedback.docx
15 KB
<
https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671130344765626535&th=17310c31cd2198a7&view=att&disp=safe
>

Preview attachment stats.pdf
<
https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671397124497524079&th=1731fed44c48c56f&view=att&disp=safe
>
stats.pdf
224 KB
<
https://mail.google.com/mail/u/0?ui=2&ik=7b3eb0b4ff&attid=0.1&permmsgid=msg-f:1671397124497524079&th=1731fed44c48c56f&view=att&disp=safe
>

2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva

Reinforcement Learning proposal:

2.2.1.3 Graph Proposal Phil Ratsloff et al

P1709R1: Graph Proposal for Machine Learning

P1709R3:
https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing

https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing
<
https://nam02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.google.com%2Fdocument%2Fd%2F1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE%2Fedit%3Fusp%3Dsharing&data=02%7C01%7CPhil.Ratzloff%40sas.com%7C729b2cf8502641e4ae5e08d749064578%7Cb1c14d5c362545b3a430
9552373a0c2f%7C0%7C0%7C637058163592253027&sdata=4UQm8tqrcUbiZsr200UMrOaEModJYGNgP1oNot9PbAg%3D&reserved=0>

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>

2.2.1.4: Differentiable Programming by Marco Foco

<
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit
>

2.2.3 any other proposal for reviews?

2.3 Other Papers and proposals

P1416R1: 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

2.2.2 SG14 Linear Algebra progress:
Different layers of proposal
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit

2.5 Future F2F meetings:

2.6 future C++ Standard meetings:
https://isocpp.org/std/meetings-and-participation/upcoming-meetings

-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

Jul 9, 2020 02:00 PM
Aug 13, 2020 02:00 PM
Sep 10, 2020 02:00 PM
Oct 8, 2020 02:00 PM

Minutes for 2020/10/08 SG19 Conference Call

Meeting notes.

> 1.1 Roll call of participants
>

Phil Ratzloff
Kevin Deweese
Matthew Galati
Richard Dosselmann
Will Wray
Xu Tony Liu
Scott McMillan
Harish Naik
Michale Wong
Andrew Lumsdaine
Larry Lewis
Jens Maurer

> 1.2 Adopt agenda
>
> 1.3 Approve minutes from previous meeting, and approve publishing
> previously approved minutes to ISOCPP.org
>
> 1.4 Action items from previous meetings
>
> 2. Main issues (125 min)
>
> 2.1 General logistics
>
> Meeting plan, focus on one paper per meeting but does not preclude other
> paper updates:
>
> Apr 9, 2020 02:00 PM EDT 1800 UTC : stats paper- DONE
> May 14, 2020 02:00 PM 1800 UTC : Stats paper replaces Differential
> calculus DONE
> Jun 11, 2020 02:00 PM 1800 UTC : Graph paper- DONE
> Jul 9, 2020 02:00 PM 1800 UTC : Stats paper -DONE
> Aug 13, 2020 02:00 PM 1800 UTC : Differential calculus + Reinforcement
> Learning
> Sep 10, 2020 02:00 PM 1800 UTC : CPPCON cancellation
> Oct 8, 2020 02:00 PM 1800 UTC : Graph paper
>
> Nov 12, 2020: 1800 UTC: DST Madness cancellation and break
>
> Dec 10, 2020: 1800 UTC: stats paper
>
> Jan 14, 2021: 1800 UTC: differential calculus + reinforcement learning
>
> Feb 11, 2021: 1800 UTC: Graph paper
>
> ISO meeting status

>
> CPPCON report
>
> future C++ Std meetings
>
> 2.2 Paper reviews
>
> 2.2.1: ML topics
>
> 2.2.1.1 Stats review Richard Dosselman et al
>
> P1708R1: Math proposal for Machine Learning
>
>
https://docs.google.com/document/d/1VAgcyvL1riMdGz7tQIT9eTtSSfV3CoCEMWKk8GvVuFY/edit
>
> > std.org/jtc1/sc22/wg21/docs/papers/2020/p1708r2
> > above is the stats paper that was reviewed in Prague
> > http://wiki.edg.com/bin/view/Wg21prague/P1708R2SG19
> >
> > Review Jolanta Polish feedback.
> > http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html
>
> Unit library cppcon 2020:
>
> https://www.youtube.com/watch?v=aN6-Kz0HqRw&feature=emb_logo
>
stats proposal now have history, concept and range use, addressing feedback
from Jolant and Walter, one scan for sorted_quantiles
have sorted variants working any kind of range
unsorted works on random access range
example use structured binding
still support median of a string
different defaults for skewness
send to Jolanta, Walter email

> 2.2.1.2 Reinforcement Learning Larry Lewis Jorge Silva
>
> Reinforcement Learning proposal:
>
we need something that owns the data as opposed to mdspan, but we need
something for mdarray
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p1684r0.pdf
or ndarray
https://www.youtube.com/watch?v=aHw0UxiCAFs&feature=emb_logo

> 2.2.1.3 Graph Proposal Phil Ratsloff et al
>
> P1709R1: Graph Proposal for Machine Learning
>
> P1709R3:
>
>
https://docs.google.com/document/d/1kLHhbSTX7j0tPeTYECQFSNx3R35Mu3xO5_dyYdRy4dM/edit?usp=sharing
>
>
>
https://docs.google.com/document/d/1QkfDzGyfNQKs86y053M0YHOLP6frzhTJqzg1Ug_vkkE/edit?usp=sharing
>
> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p2119r0.html>
>
changes since R2:
input from LEWG and Intel, want to be involved so changes are based on that input
name changes are made
used to have directed adjacency array is now directed adjacency vector, represents storage for the vector to help understand underlying storage, array is confusing
Always use long names and no abbreviations; this makes me happy because code is read more then is written
drop _c for concepts because it is not the naming convention, all dropped except in one case where there is a name conflict

Graph trait introduced has type we need for the graph, depends on the needs of the algo are and which type to be used, want to emphasize the uniform ones to work on direct (with inward and outward type) and indirected graph graph work on organized adjacency, no such thing as a directed adjacency list, we need to think zen gardening, what does it do
if you have inward and outward type then you might want that distinction

in graph trait structure seems to have many redundant names, is there any way to remove half of those typedefs and streamline with a structure that captures hierarchy?
AL: graphs is a range of ranges so expect user to specialize the graph trait for their own graph, so better to minimize those requirements, thus having global template aliases or spend that much on names depends : use some of those trait classes for iterators or ranges, rather then having to describe them here
JM: one type you have is the range type and the rest falls out from that; some kind of adaptation or user input may be ok

AL: inner container is at least forward range and we can have a doubly linked list

added an edge key type based on recommendation, nothing in algo use that though I have used it internally
Why const edge key type?
edge key type with actual value type, then there should not be const variant; a const variant poins to something that is not modifiable

ranges has a lot more structure

on to functions:
no except is now only on the size functions
added new functions, range of vertices on a graph
can be based on vertex or vertex key
now also have vertice size function

what you get back is the inner range
what is a verex vs vertex key? Vertex range does not refer to the keys
function in the library may be expensive; as expensive as iterating through edges, these are constant time look ups,
If I have a vector of list, it would first pull out the key, and then do a search within the outer vector to lookup the vertex

Are these new functions useful and what should they be returning?
DO we separate vertex size function?
Vertex key gets back same vertex range type seems limiting if I want a different data structure when I do a key lookup. What does it do? give me adjacent vertex . So why is it a non-const reference? could be a mistake.
I was debating whether it is an integer or a string, but in vector of
lists example, when sifting you are shifting through the vector of edges, whereas when you enumerate vertexes of the entire graph and is a different kind of iteration
So essentially a double lookup, give me next edge of the list and also give me the internal lookup

Now the graph data structure
No changes to the algorithms
added 4 concepts but they were duplicates and now there are 2: extract value of the range or the property of the vertex and verify it is an input range
but the name ranges does not exists in global namespace, Oh Ok I was using range-v3
declare template parameter with template syntax, can omit ... so think you want to swap the first 2 template parameter for vertex_range_extractor

move this concept into template parameter list
will process on my own

show example graph trait to demonstrate how it might be used

this is partial specialization of class template instead of overriding, should be no partial specialized function template, but could be explicit specialization

added section at the bottom for adapting to external graphs, need expansion to allow other people's external graphs,

one comment: function that have to return when passing a graph and a vertex in an Edges function, when vertex is not a member of the graph, should I return an empty function? make it UB but it may be a vertex with no edges, unless we say it must be valid vertex; so give rationale why UB is chosen

suggestion: remove adjacency matrix since focus is on algorithm, Intel asked for it
should the paper only consists of algorithms, and whether we should provide a graph data structure that can be used out of the box - one case where it is useful is compressed sparse row matrix where inner container are all taken from contiguous array

want a trivial vector of lists works as a graph

separate paper: the difference between adjacency thing and the data that the graph comes from, functionality to help extract that kind of adjacency structure and that would be useful

I understand people are uncomfortable to have a data structure with this paper

graph have edges and vertices, adjacency structure gives a way to traverse edges, incidence records the connection between a vertex and edge, like an array
adjacency: vertexes in rows and columns
incidence: vertexes in rows but edges on columns, so edges will leave a 1 or -1, so when you multiple matrix times transpose, it will be adjacency matrix; records connection a set of things and another set of things e.g. set of actors that work together to create a movie database, index into an inner container is not something you can index into an outer container; recommender systems are based on connection between one set to another set of entities

there are so many ways to create graph, so lets provide something of general use

CSR has the possibility of gap between vertexes
data structure papers takes years because they add concerns for performance, caching, concurrency

dont spend too much time with integration with concept paper as that is still in flux

then it also decouples the adjacency matrix or not, wnat to kick the tire and give me something to work with

what about a graph view, interested in doing something for that

in what way will it differ from graph data structure? allow graph subset which is not like a view which is non-owning

need more concept

constexpr vector is another are of concern

> 2.2.1.4: Differentiable Programming by Marco Foco
>
> <
>
>
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit
> >
>
> 2.2.3 any other proposal for reviews?
>
> 2.3 Other Papers and proposals
>
> P1416R1: 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
>

> 2.2.2 SG14 Linear Algebra progress:
> Different layers of proposal
>
>
https://docs.google.com/document/d/1poXfr7mUPovJC9ZQ5SDVM_1Nb6oYAXlK_d0ljdUAtSQ/edit
>
> 2.5 Future F2F meetings:
>
> 2.6 future C++ Standard meetings:
> https://isocpp.org/std/meetings-and-participation/upcoming-meetings
>
> None
>
> 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
>
> Nov 12, 2020: 1800 UTC: DST Madness cancellation and break
>
> Dec 10, 2020: 1800 UTC: stats paper
>
> Jan 14, 2021: 1800 UTC: differential calculus + reinforcement learning

>
> Feb 11, 2021: 1800 UTC: Graph paper