AN INTERCOMPARISON OF PRECIPITATION EXTREMES FROM COMMUNITY EARTH SYSTEM MODEL LARGE ENSEMBLES WITH HISTORICAL OBSERVATIONS

A Thesis Presented by

Udit Bhatia

to

The Department of Civil and Environmental Engineering

In partial fulfillment of the requirements for the degree of

Master of Science

in the field of

Environmental Engineering and Water Resources

Northeastern University
Boston, Massachusetts

December 2016
ABSTRACT

Projection of changes in extreme indices of climate variables such as temperature, precipitation and wind are critical to assess the potential impacts of climate change on built and natural systems. While suite of Multi Model Ensembles (MMEs) of General Circulation Models (GCMs) have been used to quantify the uncertainty in both mean and extreme attributes, influence of internal variability to the uncertainty is often confused with model error. In this study, statistical attributes of 35-member ensemble of Community Earth System Model Large Ensemble (CESM-LE) are analyzed and compared with historical observations. We measure bias for each ensemble and observations between a) 30-year return level; b) 100-year return level; c) parameters of underlying extreme value distribution at each grid point for Contiguous United States (CONUS) for the period of 1948-2005. Parameters of statistical distribution are obtained by individually considering all 35 runs as well as using Extreme Value Blending (EVB) approach. All individual members and combined samples unequivocally underestimate precipitation extremes for most of the regions in the CONUS. By contrast, statistical dispersion of the underlying probability distribution does not exhibit any consistent patterns, which is mainly due to inconsistencies in shape parameter of different runs. Moreover, the EVB approach proposed here can inform how natural climatic variability can be assimilated into Intensity Duration Frequency (IDF curves) to account for natural climatic variability.
ACKNOWLEDGEMENT

Funding was provided by Northeastern University, NSF CISE Expeditions in Computing award # 1029711, NSF CyberSEES award #1442728, and NSF BIGDATA award # 1447587. Author thanks Professor Auroop R. Ganguly, Thomas J. Vandal from Northeastern University and Evan Kodra, CEO, risQ Incorporation for their invaluable comments and feedback on the experiment design and manuscript.
TABLE OF CONTENTS

1. Introduction ......................................................................................................................... 1
2. Motivation and Scope .......................................................................................................... 3
3. Methodology and Datasets ................................................................................................. 4
4. Results and Discussion ....................................................................................................... 6
5. Conclusion .......................................................................................................................... 11
6. References .......................................................................................................................... 12
Appendix .................................................................................................................................. 14
1. INTRODUCTION

Climate extremes may be defined inclusively as severe hydrological or weather events, as well as significant regional changes in hydrometeorology, which are caused or exacerbated by climate change [1,2], and which may in turn cause severe stresses on regional resources, economy and the environment. While regional warming and heat waves, and perhaps heavy precipitation, can be attributed to climate change with a degree of credibility and projected relatively reliably, significant uncertainties continue to exist for regional hydrology, including floods and soil moisture, as well as tropical cyclones or hurricanes and droughts. Intergovernmental Panel of Climate Change (IPCC) Special Report on Extremes 2012 (SREX) has adopted event based definition of climate extremes. It defines extremes as “occurrence of a value of a weather or climate variable above (or below) a threshold value near the upper (or lower) ends of the range of observed values of the variable” [3].

The major constraints in translating climate extremes science to adaptation-relevant insights are the uncertainties in our understanding and in projections at (local to regional) scales and (decadal) planning horizons relevant to stakeholders. At regional and decadal scales process understanding and model projections are less accurate, while at decadal scales the uncertainties are dominated by natural variability and hence difficult to translate to risk based design principles. While there are strong evidences of human influence in warming of atmosphere and the ocean and in changes in the global water cycle and changes in climatic extremes [4], the low confidence in presence of trends in certain extreme events such as intensification of hurricanes, droughts and the subsequent attribution to human-activities makes adaptation and planning for these extreme events daunting task.

In context of climate related extreme events, The four time horizons to be considered are the following [5]:

(a) Near-real time to weeks
(b) Seasonal to inter-annual
(c) Decadal to mid-century
(d) Multi-decadal to century and beyond.

The four time horizons have natural “boundaries” from multiple perspectives, specifically, natural systems and their predictability, engineered systems and their
vulnerability, as well as human systems and their adaptability, as well as different tradeoffs in terms of what action may be necessary and indeed what insights may be actionable [6].

Near-real time to weekly time scales represent horizons over which high-resolution predictions are possible, while beyond these timescales limits to predictability in hydrological and meteorological systems creep in owing to chaos, or extreme sensitivity to initial conditions [7]. Within these time frames, short term monitoring and predictions may lead to urgent and immediate events and emergency management. While weather changes and short-term population movements may be important at these timescales, major changes in climate or demographics are not expected, while infrastructures, lifelines, economic resources and technological capabilities are expected to remain unchanged.

Seasonal to interannual time scales produce changes in seasonal and interannual climate patterns such as the El Nino phenomenon where phenomena related to natural climate and water variability such as seasonal changes in monsoons in the Southwest US or seasonal floods in Iowa or the severity of a blizzards in winter in the Northeast US, or the possible water quality degradation in lakes such as Erie leading to drinking water pathogens in Ohio, or seasonal droughts in California or the Southeast and the implications of these droughts on annual wine or orange production in California are of concern. While specific weather events are not predictable at these timescales, the average seasonal and annual patterns may be characterized.

Decadal to mid-century time scales, range from about 5 to 30 years in the future, and perhaps all the way to the 2050's, does not have weather or seasonal or even annual predictability for hydrological or hydro-meteorological processes. However, physics-based climate models can project climate trends and variability based on assumed emission scenarios, which in turn are based on storylines for population and technological or societal changes at aggregate scales. However, while non-stationary signals, which include trends in global warming and changes in the statistics of weather patterns and relations, are expected to start becoming prominent at these horizons, the variability in mean and extreme climate is expected to be large and exhibit an increase. In fact, the uncertainty is expected to be comparable to the trend, and the intrinsic variability of the climate projections is expected to be a large component of the overall uncertainty [8].
Beyond decadal to mid-century time scales, climate trends are expected to dominate over internal variability, but projections for population and human systems are not available and indeed difficult to project other than as what-if scenarios. In addition, many stakeholder decision horizons do not extend to these scales, other than perhaps for long-term policy makers. Credible and actionable insights are possible only at aggregate scales, which further restricts which of these insights may be actionable. The climate change adaptation community, as well as the related integrated assessments community, has been working at these models with a variety of interconnected but highly simplified models. The models range from being physically based to agent-based and often rely on system dynamics approaches. The community is used to examining simple models so much so that global climate or earth system models in their full complexity was never used in these integrated assessment models (IAMs), although recent developments have attempted to incorporate the latest earth system models.

In present study, we focus on accounting for variability in extremes by analyzing the outputs of Multiple Initial Condition Ensembles (MICE) of Community Earth System Large Ensembles (CESM-LE) [9]. The reasoning behind focusing on this type of uncertainty is that since internal variability can mask or enhanced the consequences of changing climate at annual to decadal scales, it must be properly accounted while considering observations, and understanding projections [10]. More details about CESM-LE are discussed in subsequent sections.

2. Motivation and Scope

While more and more modeling groups are producing multiple runs [9], the question that still remains unanswered is that how many realizations are necessary to know that model outputs are statistically significant. In other words, what is the minimum number of initial condition realizations are required to estimate model mean output with a given statistical confidence level and error tolerance? Since number of runs depend on multiple factors such as a) whether mean or extreme statistics are required b) what is the spatial and temporal scales at which statistics are required and c) what is the statistical certainty and error tolerance? Answer to these questions lies in understanding the ergodicity of climate systems:
In context of climate systems, ergodicity refers to the fact that inter-realization variance of mean of each period (say decade) form an ensemble of transient runs is statistically to the variance of the decadal mean from a stationary control run [11]. If models are ergodic, variance of control sample run can help in estimating number of runs required.\(^1\)

Before estimating the ergodicity of Multiple Initial Condition Ensembles, it is crucial to analyze the convergence of attributes of interest (return periods and distribution parameters of Generalized Extreme Value in present case) and evaluate the agreement between observations and model outputs. The scope of this work is to: a) Estimate the agreement between various extreme indices (expressed in terms of return-level). b) Evaluate the performance of Extreme Value Blending approach in simulating parameters of extreme value distribution in comparison to observations.

### 3. Methodology and Datasets

Based on the prior literature [12–14], Generalized Extreme Value distribution, which is based on the block maxima theory is adopted to quantify the intensity of precipitation extremes. GEV theory is a family of continuous distribution that combine type I (Gumbel), II (Fréchet) and type III (Weibull) extreme value distributions. The GEV is the only possible limit distribution of sequence independent and identically distributed random variables’ maxima that are properly normalized. The GEV has cumulative distribution function:

\[
F(x; \mu, \sigma, \xi) = \exp \left\{- \left[1 + \xi \left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}
\]

(1)

It is the three-parameter distribution where \(\mu, \sigma\) and \(\xi\) represents location parameter, scale parameter and the shape parameter respectively. In statistics, location parameter determines the shift of distribution, scale parameter quantifies spread (or variability) of the distribution and shape parameter controls symmetry of the distribution [15]. To model series of extremes, a series of independent observations \(X_1, X_2, \ldots, X_n\) is considered for some large value of \(n\). Data are blocked into such sequences and a series of block maxima \(M_{n1}, M_{n2}, \ldots, M_{nm}\) to which GEV is to be fitted is generated. For example, if \(n\) corresponds

---

\(^1\)http://www.cesm.ucar.edu/working_groups/Variability/presentations/2006/wehner_joint_session.ppt
to the number of observations in each year and m number of years are considered, block maxima corresponds to annual maxima.

Estimates of extreme quantiles of the annual maximum distribution are then obtained by inverting (1):

\[
x_p = \mu - \frac{\sigma}{\xi} \left[ 1 - \{-\log (1 - p)^{-\frac{1}{\xi}} \right] \quad \text{for } \xi \neq 0 \quad (2)
\]

\[
x_p = \mu - \sigma \log (1 - p) \quad \text{for } \xi = 0 \quad (3)
\]

where \(x_p\) is the return level associated with return period \(1/p\). In other words, \(x_p\) is exceeded by the annual maxima in given year by probability \(p\). We employ the Annual Maxima approach to construct the series of block maxima because assumption of independent and identically distributed samples can be better satisfied using annual, likely dependent, points.

To simultaneously account for MICE outputs, we propose Extreme Value Blending (EVB) approach in combination with GEV. EVB-GEV is implemented as follows:

- For each initial condition run output, \(O_i\), annual maximum precipitation is computed for each year for time-period of interest.
- For \(n\) years, \(O_i\) yields \(n\) data points. If \(N\) models are considered, total of \(N \times n\) are available as sample point. It is hypothesized that the representation of large number of sample points with same underlying generation process can better represent the population of extreme values in comparison to single runs.
- \(N \times n\) samples are then used to estimate the parameters of underlying distribution. (\(\mu, \sigma\) and \(\xi\) in present case).

We consider the period of 1948-2005 to analyze precipitation extremes over Contiguous United States. CESM-LE project output provides the set of climate simulations for advancing the understanding of internal variability and climate change. All simulations are performed with the nominal 1-degree latitude/longitude version of the Community Earth System Model (CESM). The Large Ensemble Project includes a 40-member ensemble
out of which 35 are publicly available through project website. Each member of CESM-LE experiment is subject to same radiative forcing scenario with a slightly different initialization of atmospheric state. For observations in the corresponding periods, we use National Oceanic and Atmospheric Administration (NOAA) Climate Prediction Center (CPC) 0.25° daily unified Precipitation. Since the resolution of model outputs and observations are different, observed precipitation is linearly interpolated to match the grid resolution of the model (See Appendix). Multiple return periods (10 years, 30 years, 50 years and 100 years), shape, scale and location parameter is then computed for each grid point at model resolution. All algorithms are coded in Python 2.7 and codes are enclosed in Appendix.

4. RESULTS AND DISCUSSION

![Figure 1: 30 Year return period of daily accumulated precipitation for contiguous United States obtained from Community Earth System Model-LE (CESM-LE) model output.](http://www.cesm.ucar.edu/projects/community-projects/LENS/)
Figure 1 shows the 30-year return period for 30 initial condition runs as obtained from CESM-LE output. It is observed that while all 30 models can capture large scale characteristics (i.e. drier patches in western and mid-western regions in comparison to gulf coastal regions), almost all runs unequivocally underestimate the return levels in gulf coast (Figure 2).

![Figure 1: 30-year return period for CESM-LE output.](image)

**Figure 2: Difference of 30 years return levels obtained from model and observations. (30YR\textsubscript{MODEL} - 30YR\textsubscript{OBSERVATION}).**

We further investigate the statistical attributes to examine why models underestimate return values. Since return values are function of shape, scale and location parameters, we compute bias for each parameter for each run, where bias of each attribute is measured as difference between the value of attribute obtained from model run and observation at each grid point. We observe that while consistency in patterns of bias in location parameter explains unequivocal underestimation, inconsistent variations in bias of shape parameters could be the plausible explanation for statistical dispersion of biases in northern and central regions.
FIGURE 3: BIAS IN SHAPE FACTOR (SHAPE_{MODEL} - SHAPE_{OBSERVATIONS})

FIGURE 4: BIAS IN SCALE FACTOR (SCALE_{MODEL} - SCALE_{OBSERVATIONS})
Finally, we estimate the 30 year and 100-year return levels using EVB-GEV approach and note that we could capture the entire spectrum of return periods for each grid point (Figure 6) but with reduced bias in location and scale parameters (Figure 7). However, statistical dispersion still does not exhibit any consistent patterns (Figure 8), which needs further investigation.

**Figure 5:** Same as Figure 3 and 4 but for location parameter.

**Figure 6:** (Left) 30-year and (right) 100-year return levels as obtained from EVB-GEV approach.
Figure 7: Bias in (left) location and (right) scale parameters as obtained from EVB-GEV approach.

Figure 8: Bias in (left) 30 year return levels and (right) shape parameters as obtained from EVB-GEV approach.
5. Conclusion

By examining the Extreme Value Theory (EVT) based precipitation extremes and MICE of CESM-LE, it is shown that almost all individual members and combined samples underestimate precipitation extremes for most of the regions in the CONUS. However, statistical dispersion of the underlying probability distribution does not exhibit any consistent patterns, which is mainly due to inconsistencies in shape parameter of different runs. While multimodel ensembles have been used for estimation of climate extreme indices [16] and design of Intensity, Duration and Frequency (IDF) curves [12], it is crucial to address the role of internal variability specially for design and planning purposes at annual to decadal scales. It is shown that the proposed EVB approach has potential to not only capture the entire spectrum of extreme indices, but also reduce the bias that is otherwise predominant in statistical attributes of individual runs. Proposed EVB approach can be assimilated into design Intensity Duration Frequency (IDF curves) to account for natural climatic variability at annual to decadal scales in addition to climate trends.
6. References


APPENDIX

A. RETURN PERIODS AND DISTRIBUTION PARAMETERS FOR OBSERVATIONS

As discussed under section methods and datasets, model outputs obtained from CESM-LE and daily accumulated precipitation obtained from CPC are at different resolution, we interpolate the observations (0.25 degree x 0.25 degree) to match the grid resolution of model output (1.25 degree x 1.25 degree). Since interpolation to coarser grid resolution could alter the statistical properties of each region, we inspect the agreement between various statistical properties at original and interpolated grid resolution of observations. Before we generate conclusive insights, agreement between original and interpolated attributes should be evaluated. Figure A1 presents the summary of comparison of various statistical attributes computed on original and interpolated observations.

![Figure A1: Location parameter of (left) observations at original grid resolution and (right) at interpolated grid resolution.](image-url)
Figure A3: Same as figure A1 but for shape.

Figure A4: Same as figure A1 but for 30 year return period. As expected, interpolation masked regional extremes at multiple grid points, because of which interpolate results underestimate the return values of 30-year return levels.
B. Codes

B1. Class to generate Block Maxima

```python
import pprint, pickle
import xarray as xr
import os as os
import pandas as pd
import numpy as np
import glob
from matplotlib import *
import matplotlib
import matplotlib.pyplot as plt
import pylab
import sys
fig = pylab.gcf()
from joblib import Parallel, delayed
import multiprocessing
import datetime
import scipy.interpolate
import pprint, pickle

# model is xarray dataset

class Blockmaxima(object):
    def __init__(self, model, mm = "Y"):
        self.model = model
```

if mm == "N":
    self.model = self.model*(24*60*60*1000)
def time_extract(self):
    lats = self.model.lat.values
    longs = self.model.lon.values
    time_start = self.model.time.values[0]
    time_end = self.model.time.values[-1]
    start_year = pd.to_datetime(time_start).year
    end_year = pd.to_datetime(time_end).year
    array = np.zeros((len(lats),len(longs),(end_year - start_year)+1))
    array[array==0] = np.nan
    for i,lat1 in enumerate(lats):
        for j,lon1 in enumerate(longs):
            model = self.model.sel(lat = lat1,lon = lon1)
            mask = model.data_vars.keys()
            attr_name = "model."+mask[0]+".values[0]"
            if np.isnan(eval(attr_name)):
                continue
            for k,year in enumerate(range(start_year,end_year+1)):
                start = datetime.datetime(year,1,1)
                end = datetime.datetime(year,12,31)
                yearly_output = model.sel(time = slice(start,end))
                mask = model.data_vars.keys()
                attr_name = "yearly_output."+mask[0]+".values"
                maxima = max(eval(attr_name))
                array[i,j,k] = maxima
                print "hello"
    self.blockmaximum = xr.Dataset({"block_maxima":(('lat','lon','time'),array)},
            coords={"lat":lats, "lon":longs , "time" : range(start_year,end_year+1) })
    return (self.blockmaximum)

if __name__ == "__main__":
path = "Volumes/GCM/MICE/"
path2 = "Volumes/GCM/Observed/"

os.chdir(path)

# Opening the masked files created using masking.py
pkl_file = open('masked_gcms.pkl', 'rb')
masked_models = pickle.load(pkl_file)
pkl_file.close()

block_maxima_le = []

for i in range(len(masked_models)):
    model = Blockmaxima(masked_models[i])
    maximum = model.time_extract()
    block_maxima_le.append(maximum)

print i

output = open('block_maxima_le.pkl', 'wb')
pickle.dump(block_maxima_le 输出, output, -1)
output.close()
import pickle, pprint
from gev_rpy import*
from matplotlib import pyplot
import rpy2
import rpy2.robjects as ro
import rpy2.robjects.packages
from rpy2.robjects.packages import importr
from rpy2.robjects.vectors import FloatVector
ismev = importr("ismev")
extremes = importr("extRemes")

import os
import pickle

# Lists of models and observation files required as arguments. Also include pathnames.

class GEV(object):
    import xarray as xr
    import pandas as pd
    import numpy as np
    import rpy2
    import rpy2.robjects as ro
    import rpy2.robjects.packages as rpackages
    from rpy2.robjects.packages import importr
    from rpy2.robjects.vectors import FloatVector
    ismev = importr("ismev")
extremes = importr("extRemes")
    # Model is an xarray object
def __init__(self,model):
        self.model = model
def return_levels(self):
lats = self.model.lat.values
longs = self.model.lon.values
parameters = []
return_value = []
lati = []
loni = []
for i, lat1 in enumerate(lats):
    for j, lon1 in enumerate(longs):
        return_levels = {}
        params = {}
        maxima = self.model.sel(lat=lat1, lon=lon1)
        var_keys = maxima.data_vars.keys()
        attr_name = "maxima."+var_keys[0]+".values[0]"
        if abs(eval(attr_name)) >= 0:
            lati.append(lat1)
            loni.append(lon1)
            block_maxima = "maxima."+var_keys[0]+".values"
            block_maxima = eval(block_maxima)
            block_maxima = block_maxima[~np.isnan(block_maxima)]
            block_maxima = block_maxima[block_maxima < 400]
            block_maxima = FloatVector(block_maxima)
            fevd_object = extremes.fevd(block_maxima, verbose=False)
            params['location'] = fevd_object.rx2('results')[0][0]
            params['scale'] = fevd_object.rx2('results')[0][1]
            params['shape'] = fevd_object.rx2('results')[0][2]
            parameters.append(params)
return_periods = [2.5, 5, 10, 20, 30, 50, 100]
for r in return_periods:
    return_level = extremes.return_level(fevd_object, return_period=r)[0]
    name = 'years_' + str(r)
    return_levels[name] = return_level
return_value.append(return_levels)
return_value = pd.DataFrame(return_value).set_index([lati,loni])
return_value = return_value.to_xarray()
return_value = return_value.rename({'level_0': 'lat', 'level_1': 'lon'}, inplace=True)

parameters = pd.DataFrame(parameters).set_index([lati,loni])
parameters = parameters.to_xarray()
parameters = parameters.rename({'level_0': 'lat', 'level_1': 'lon'}, inplace=True)
#parameters = xr.Dataset({'parameters': parameters})
self.return_value = return_value
self.params = parameters
return {'return values': self.return_value, 'parameters': self.params}

if __name__ == '__main__':
    path = '/Volumes/GCM/MICE/
    path2 = '/Volumes/GCM/Observed/
    os.chdir(path)
pkl_file = open('block_maxima_le.pkl', 'rb')
maximums = pickle.load(pkl_file)
pkl_file.close()

return_periods = []
for i in range(len(maximums)):

    model = GEV(maximums[i])
    maximum = model.return_levels()
    return_periods.append(maximum)
    print i

B3. Program to IMPLEMENT Extreme Value Blending

import xarray as xr
import os as os
import pandas as pd
import numpy as np
import glob
from matplotlib import *
import matplotlib.pyplot as plt
import pylab
import sys
fig = pylab.gcf()
from joblib import Parallel, delayed
import multiprocessing
import scipy.interpolate
import pickle, pprint
from gev_rpy import *
from matplotlib import pyplot
import rpy2
import rpy2.robjects as ro
import rpy2.robjects.packages as rpackages
from rpy2.robjects.packages import importr
from rpy2.robjects.vectors import FloatVector
ismev = importr("ismev")
extremes = importr("extRemes")
# Lists of models and observation files required as arguments. Also include pathnames.
path = "/Volumes/GCM/MICE/
path2 = "/Volumes/GCM/Observed/
print(path)
pkl_file = open('block_maxima_le.pkl', 'rb')
maximums = pickle.load(pkl_file)
pkl_file.close()
flatten = lambda l: [item for sublist in l for item in sublist]  # Flatten lists
lats = maximums[0].lat.values
lons = maximums[0].lon.values
parameters = []
return_value = []
lati = []
loni = []
for lat in lats:
    for lon in lons:
        return_levels = {}
        params = {
            maxima = maximums[0].sel(lat = lat, lon = lon)
        }
        keys = maxima.data_vars.keys()
        attr_name = "maxima."+keys[0]+".values[0]"
        if np.isnan(eval(attr_name)):
            continue
        appended_maxima = []
        lati.append(lat)
        loni.append(lon)
        for i, maximum in enumerate(maximums):
            maxima1 = maximums[i].sel(lat = lat, lon = lon)
            maxima_vals = list(eval("maxima1."+keys[0]+".values"))
            appended_maxima.append(maxima_vals)
        flat_max = flatten(appended_maxima)
        flat_max = FloatVector(flat_max)
        fevd_object = extremes.fevd(flat_max, verbose = False)
        params['location'] = fevd_object.rx2('results')[0][0]
        params['scale'] = fevd_object.rx2('results')[0][1]
        params['shape'] = fevd_object.rx2('results')[0][2]
        parameters.append(params)

return_periods = [2.5, 5, 10, 20, 30, 50, 100]
for r in return_periods:
    return_level = extremes.return_level(fevd_object, return_period = r)[0]
    name = 'years_' + str(r)
return_levels[name] = return_level

return_value.append(return_levels)

return_value = pd.DataFrame(return_value).set_index([lati,loni])

return_value = return_value.to_xarray()

return_value = return_value.rename({'level_0': 'lat', 'level_1': 'lon'}, inplace=True)

parameters = pd.DataFrame(parameters).set_index([lati,loni])

parameters = parameters.to_xarray()

parameters = parameters.rename({'level_0': 'lat', 'level_1': 'lon'}, inplace=True)

print parameters

print return_value

stats = {}

stats['parameters'] = parameters

stats['return_value'] = return_value

output = open('return_periods_single_parameter.pkl', 'wb')

pickle.dump(stats, output, -1)

output.close()