Building A Crystal Ball: Forecasting Future Values For Multi-cyclic Time Series Metrics In Splunk

Mike Fisher
Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not, be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.
About Me

• Splunk user/administrator for 7 years

• Work for a Fortune 100 financial firm

• Currently leading a Monitoring and Operational Intelligence team

• I was not a Statistics major in college!
Agenda

- The Problem
- Existing Tools
- Finding A Better Way
- Implementation
- Results
- Caveats
- Questions
The Problem
Many Time Series Contain Cyclic Patterns

- Sales per hour
- Web page hits
- Network traffic
Some Have Multiple Concurrent Cycles

- Daily cycle
- Weekly cycle
How Do We Know What’s Normal?

- **Sales per minute** - Are sales abnormally low right now?
- **Web page hits** - Is my web site experiencing high traffic?
- **Network traffic** - Is that spike in network traffic expected?
How Do We Set Alert Thresholds?

Here vs Here
How Do We Alert?

... if we don’t know what’s normal at any given time?
Existing Tools
Splunk’s predict() Command

- The `predict` command forecasts values for one or more sets of time-series data.

- Two algorithms that deal with seasonal data:
  
  LLP – Seasonal local level
  
  LLP5 - Combines local level trend and seasonal local level
Al’s Online Toy Barn Sales

index=summary search_name= "Sales - Summary - 10 min count"
| timechart span=10m sum(count) as actual
| predict actual as pred algorithm=LLP upper90=high lower90=low future_timespan=432
Forecast Using LPP

5 weeks of data, 3 days of forecast, 90% confidence intervals
LLP Forecast vs Reality
Forecast Using LPP5

5 weeks of data, 3 days of forecast, 90% confidence intervals
LLP Forecast vs Reality
The Future Is Fuzzy...
Finding A Better Way
Requirements

Handle multi-cyclic time series
  Fast
  Efficient
  Accurate
  Reusable
Predict The Future

...without hiring this guy
The Data

index=summary
search_name="Sales - Summary - 10 min count"
| timechart span=1h sum(count) as actual
Al’s Online Toy Barn Sales

![Al’s Online Toy Barn Sales](image-url)
Week-over-week View
Week Over Week 10 Minute Resolution
Multi-Series View
Take A Slice of Time

11 AM – Noon
# The Slice in Numbers

<table>
<thead>
<tr>
<th></th>
<th>11:00</th>
<th>11:10</th>
<th>11:20</th>
<th>11:30</th>
<th>11:40</th>
<th>11:50</th>
<th>12:00</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>04/20/16</strong></td>
<td>374</td>
<td>327</td>
<td>313</td>
<td>337</td>
<td>330</td>
<td>331</td>
<td>437</td>
</tr>
<tr>
<td><strong>04/13/16</strong></td>
<td>304</td>
<td>295</td>
<td>291</td>
<td>300</td>
<td>318</td>
<td>317</td>
<td>358</td>
</tr>
<tr>
<td><strong>04/06/16</strong></td>
<td>311</td>
<td>300</td>
<td>301</td>
<td>323</td>
<td>325</td>
<td>331</td>
<td>376</td>
</tr>
<tr>
<td><strong>03/30/16</strong></td>
<td>319</td>
<td>323</td>
<td>328</td>
<td>339</td>
<td>353</td>
<td>357</td>
<td>395</td>
</tr>
<tr>
<td><strong>03/23/16</strong></td>
<td>312</td>
<td>318</td>
<td>319</td>
<td>329</td>
<td>335</td>
<td>355</td>
<td>394</td>
</tr>
</tbody>
</table>
The Target

<table>
<thead>
<tr>
<th>Date</th>
<th>11:00</th>
<th>11:10</th>
<th>11:20</th>
<th>11:30</th>
<th>11:40</th>
<th>11:50</th>
<th>12:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/27/16</td>
<td></td>
<td></td>
<td></td>
<td>???</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/20/16</td>
<td>374</td>
<td>327</td>
<td>313</td>
<td>337</td>
<td>330</td>
<td>331</td>
<td>437</td>
</tr>
<tr>
<td>04/13/16</td>
<td>304</td>
<td>295</td>
<td>291</td>
<td>300</td>
<td>318</td>
<td>317</td>
<td>358</td>
</tr>
<tr>
<td>04/06/16</td>
<td>311</td>
<td>300</td>
<td>301</td>
<td>323</td>
<td>325</td>
<td>331</td>
<td>376</td>
</tr>
<tr>
<td>03/30/16</td>
<td>319</td>
<td>323</td>
<td>328</td>
<td>339</td>
<td>353</td>
<td>357</td>
<td>395</td>
</tr>
<tr>
<td>03/23/16</td>
<td>312</td>
<td>318</td>
<td>319</td>
<td>329</td>
<td>335</td>
<td>355</td>
<td>394</td>
</tr>
</tbody>
</table>
## Average

<table>
<thead>
<tr>
<th>Date</th>
<th>11:00</th>
<th>11:10</th>
<th>11:20</th>
<th>11:30</th>
<th>11:40</th>
<th>11:50</th>
<th>12:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/27/16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04/20/16</td>
<td>374</td>
<td>327</td>
<td>313</td>
<td>337</td>
<td>330</td>
<td>331</td>
<td>437</td>
</tr>
<tr>
<td>04/13/16</td>
<td>304</td>
<td>295</td>
<td>291</td>
<td>300</td>
<td>318</td>
<td>317</td>
<td>358</td>
</tr>
<tr>
<td>04/06/16</td>
<td>311</td>
<td>300</td>
<td>301</td>
<td>323</td>
<td>325</td>
<td>331</td>
<td>376</td>
</tr>
<tr>
<td>03/30/16</td>
<td>319</td>
<td>323</td>
<td>328</td>
<td>339</td>
<td>353</td>
<td>357</td>
<td>395</td>
</tr>
<tr>
<td>03/23/16</td>
<td>312</td>
<td>318</td>
<td>319</td>
<td>329</td>
<td>335</td>
<td>355</td>
<td>394</td>
</tr>
</tbody>
</table>

Average = 333.57

Standard Deviation = 31.66
High and Low Bounds

\[ \text{prediction} = \text{average} \]

\[ \text{bounds} = \text{prediction} \pm \text{stdev} \times \sqrt{1 / (1 - \text{confidence} / 100)} \]
High and Low Bounds

Average = 333.57
Standard deviation = 31.65

predicted = average

bounds = predicted +/- stdev * (sqrt(1/(1-confidence/100)))

low = 333.57 – 31.65 * (sqrt(1/(1-90/100))) = 233.46
high = 333.57 + 31.65 * (sqrt(1/(1-90/100))) = 433.68
How’d We Do?

Predicted = 333.57
Low bound = 233.46
High bound = 433.68

Apr 27 11:30 actual = 318
Implementation
So..... How Do We Do That In Splunk?

Simple.

Just build a macro.
### forecast5w(val, confidence, reltime, days)

```plaintext
eval w=case(
    (_time>relative_time(now(), "$reltime@$d-5w-30m") AND _time<=relative_time(now(), "$reltime@$d-5w+$days$d+30m")), 5,
    (_time>relative_time(now(), "$reltime@$d-4w-30m") AND _time<=relative_time(now(), "$reltime@$d-4w+$days$d+30m")), 4,
    (_time>relative_time(now(), "$reltime@$d-3w-30m") AND _time<=relative_time(now(), "$reltime@$d-3w+$days$d+30m")), 3,
    (_time>relative_time(now(), "$reltime@$d-2w-30m") AND _time<=relative_time(now(), "$reltime@$d-2w+$days$d+30m")), 2,
    (_time>relative_time(now(), "$reltime@$d-1w-30m") AND _time<=relative_time(now(), "$reltime@$d-1w+$days$d+30m")), 1)

| eval shift=case(isnotnull(w), ""+"w-30m +"w-20m +"w-10m +"w-0m +"w+10m +"w+20m +"w+30m")
| where isnotnull(shift)
| makemv shift
| mvexpand shift
| eval time=relative_time(_time, shift)
| eventstats avg($val$) as pred by time
| eval upper=if($val$>pred, $val$, pred)
| eval lower=if($val$<pred, $val$, pred)
| stats avg($val$) as pred, stdev(upper) as ustdev, stdev(lower) as lstdev by time
| eval low=pred-lstdev*(sqrt(1/(1-$confidence$/100)))
| eval low=if(low<0, 0, low)
| eval high=pred+ustdev*(sqrt(1/(1-$confidence$/100)))
| event _time=time
| timechart span=10m min(pred) as pred, min(low) as low, min(high) as high
| where _time>relative_time(now(), "$reltime@$d") AND _time<=relative_time(now(), "$reltime+$days$d@d")
```

36
Any Questions?
How Do We Do That In Splunk, Really?

Short answer:

Time travel and cloning
Time Travel

Shift data points from prior weeks forward in time to where they are needed.
Cloning

Each data point will be used seven times as the forecast window slides by.
Cloning

Duplicate each data point so it can be used to calculate seven different forecast points.
But How Do We Do That In Splunk?

• Use relative_time() to calculate the time shifts.

• Use makemv and mvexpand to duplicate data.
Take Timechart Output As Our Input

index=summary search_name="Sales - Summary - 10 min count"
| timechart span=10m sum(count) as actual
| `forecast5w(actual,90,+1d,1)`
Arguments To The Macro

$\text{val}$ - The name of the field to forecast

$\text{confidence}$ - A number, $0 < N < 100$, that determines the width of the bounds

$\text{reltime}$ - Start time of the forecast relative to current time

$\text{days}$ - How many days to forecast
Only Shift The Data We Need

Example, for five weeks ago:

```plaintext
_time > relative_time(now(), "$reltime@d-5w-30m")
AND
_time <= relative_time(now(), "$reltime@d-5w+$days@d+30m")
```
For each week of data:
Compute shifts needed to move the data to seven locations needed for the forecast.

\[
\text{reltime} + 5w - 30m, \\
\text{reltime} + 5w - 20m, \\
\text{reltime} + 5w - 10m, \\
\text{reltime} + 5w - 0m, \\
\text{reltime} + 5w + 10m, \\
\text{reltime} + 5w + 20m, \\
\text{reltime} + 5w + 30m
\]
The Full Shift

eval w = case(
    (_time>relative_time(now(), "$reltime@$d-5w-30m$") AND _time<=relative_time(now(), "$reltime@$d-5w+$days$d+30m$")), 5,
    (_time>relative_time(now(), "$reltime@$d-4w-30m$") AND _time<=relative_time(now(), "$reltime@$d-4w+$days$d+30m$")), 4,
    (_time>relative_time(now(), "$reltime@$d-3w-30m$") AND _time<=relative_time(now(), "$reltime@$d-3w+$days$d+30m$")), 3,
    (_time>relative_time(now(), "$reltime@$d-2w-30m$") AND _time<=relative_time(now(), "$reltime@$d-2w+$days$d+30m$")), 2,
    (_time>relative_time(now(), "$reltime@$d-1w-30m$") AND _time<=relative_time(now(), "$reltime@$d-1w+$days$d+30m$")), 1)

| eval shift=case(isnotnull(w), ”+"+w+"w-30m +"+w+"w-20m +"+w+"w-10m +"+w+"w-0m +"+w+"w+10m +"+w+"w+20m +"+w+"w+30m"

47
Drop What We Don’t Need

| where isnotnull(shift) |
Clone The Data And Compute New Time For Each Event

| makemv shift |
| mvexpand shift |
| eval time=relative_time(_time, shift) |
Do The Math

| eventstats avg($val$) as pred by time
| eval upper=if($val$>pred,$val$,pred)
| eval lower=if($val$<pred,$val$,pred)

| stats avg($val$) as pred, stdev(upper) as ustdev, stdev(lower) as lstdev by time
| eval low=pred-lstdev*(sqrt(1/(1-$confidence$/100)))
| eval low=if(low<0, 0, low)
| eval high=pred+ustdev*(sqrt(1/(1-$confidence$/100)))
_time Travel!

| eval _time=time

* This doesn’t work reliably in Splunk versions prior to 5.4.3.
Post Jump Cleanup

| timechart span=10m min(pred) as pred, min(low) as low, min(high) as high |
| where _time>relative_time(now(), "$reltime@$d") AND _time<=relative_time(now(), "$reltime+$days$d@d") |
forecast5w(val, confidence, reltime, days)

eval w=case(
  (_time>relative_time(now(), "$reltime@d-5w-30m") AND _time<=relative_time(now(), "$reltime@d-5w+$days@d+30m")), 5,
  (_time>relative_time(now(), "$reltime@d-4w-30m") AND _time<=relative_time(now(), "$reltime@d-4w+$days@d+30m")), 4,
  (_time>relative_time(now(), "$reltime@d-3w-30m") AND _time<=relative_time(now(), "$reltime@d-3w+$days@d+30m")), 3,
  (_time>relative_time(now(), "$reltime@d-2w-30m") AND _time<=relative_time(now(), "$reltime@d-2w+$days@d+30m")), 2,
  (_time>relative_time(now(), "$reltime@d-1w-30m") AND _time<=relative_time(now(), "$reltime@d-1w+$days@d+30m")), 1
)
| eval shift=case(isnotnull(w), ""+"+""w-30m"+"+""w-20m"+"+""w-10m"+"+""w-0m"+"+""w+10m"+"+""w+20m"+"+""w+30m")
| where isnotnull(shift)
| makemv shift
| mvexpand shift
| eval time=relative_time(_time, shift)
| eventstats avg($val$) as pred by time
| eval upper=if($val$>pred,$val$,pred)
| eval lower=if($val$<pred,$val$,pred)
| stats avg($val$) as pred, stdev(upper) as ustdev, stdev(lower) as lstdev by time
| eval low=pred-lstdev*(sqrt(1/(1-$confidence$/100)))
| eval low=if(low<0, 0, low)
| eval high=pred+ustdev*(sqrt(1/(1-$confidence$/100)))
| eval _time=time
| timechart span=10m min(pred) as pred, min(low) as low, min(high) as high
| where _time>relative_time(now(), "$reltime@d") AND _time<=relative_time(now(), "$reltime+days$d@d")
Results
Generate A Forecast

```bash
index=summary search_name="Sales - Summary - 10 min count"
| timechart span=10m sum(count) as actual
| `forecast5w(actual, 90.0, +1d, 3)`
```

Run over the last 5 weeks.
Forecast Using forecast5w()

5 weeks of data, 3 days of forecast, 90% confidence intervals
forecast5w() vs Reality
Automatic Forecasting

Save search as “Sales Volume Forecast” and schedule to run every day over the previous 5 weeks.

```
index=summary search_name="Sales - Summary - 10 min count"
| timechart span=10m sum(count) as actual
| `forecast(actual, 90.0, +1d, 1)`
```
index=summary

search_name="Sales - Summary - 10 min count" OR
search_name="Sales Volume Forecast"

| where count<low
Test The Alert Based On History

• Backfill the forecast for the last month or so:
  splunk cmd python fill_summary_index.py -app search \ 
  -name "Sales Volume Forecast" -et -1mon -lt now -j 8

• Use timechart to find out when your alert would have fired:
  index=summary 
  search_name="Sales - Summary - 10 min count" OR search_name="Sales Volume Forecast" 
  | timechart sum(count) as count, sum(low) as low 
  | where count<low
Caveats

• Doesn’t perform well on low volume time series data
• Must adjust the default MAX_DAYS_HENCE in props to create forecast data more than two days in advance
• Needs a feedback loop so that abnormal data can be excluded from future forecast calculations
• Your mileage may vary
Wrap Up
Go Forth And Predict The Future!

Now that you’ve seen how to build a crystal ball, the only question is...

What will you forecast?
Questions?
THANK YOU
Photo Credits

rjrgmc28  https://www.flickr.com/photos/rjrgmc28/  (cropped from original)
License: https://creativecommons.org/licenses/by-sa/2.0/

Judy van der Velden  https://www.flickr.com/photos/judy-van-der-velden/
License: https://creativecommons.org/licenses/by-nc-nd/2.0/

Silverisdead  https://www.flickr.com/photos/56624456@N00/
License: https://creativecommons.org/licenses/by/2.0/

Nicolas Connault  https://www.flickr.com/photos/nicolasconnault/
License: https://creativecommons.org/licenses/by-nd/2.0/