A Cotton Irrigator’s Decision Support System
Using National, Regional and Local Data

ISESS 2015, Melbourne

Jamie Vleeshouwer, Nicholas J. Car, John Hornbuckle
26 March 2015
Irrigation in the cotton industry

• Background
  • Water is scarce
  • Irrigation entitlements for regions
  • Farmers maximise revenue by getting best return on water “more crop per drop”

• IrriSATSMS
  • Weather/Sattelite based irrigation decision support system
  • Provide information on how much to irrigate
  • SMS User Interface
  • Has become outdated

• IrriSAT – Next Gen
  • Use modern technology / automation
  • Add forecasting (7 days) / benchmarking functionality
How much water should I use?

- Water balance
  - Rain + Irrig – CropWaterUse = 0
  - Too much = inefficient
  - Too little = restricted growth

\[ \text{ET}_c = \text{ET}_o \times K_c \]
What are we doing about it?

• IrriSAT - weather-based irrigation scheduling service
• Satellite imagery determines crop coefficients (Kc) to calculate crop water requirements and provides customised scheduling information
• Irrigation scheduling with:
  • low-cost
  • wide coverage
• Add value - not replace existing methods
• Daily water balance updated in real time
• Targeting smartphone interfaces as well as web
How will it work?

- Regional weather data – ETo (weather stations)
- Crop growth data - Kc (satellite imagery)
- Rainfall and irrigation data (farmers via phones)

Remotely Sensed

Water balance model

Benchmarking data store

Mobile phone app
What could we improve upon?

• What didn’t work well in the previous system
  • Satellite processing was all manual (lots of time, effort, data storage)
  • Data entry errors (SMS autocorrect interfering)
  • Had commercial interest but too complex to transfer
  • Tightly coupled

• How technology has evolved
  • Cloud computing
  • Web services
  • High performance computing
  • Increased usage of smartphones / tablets

After many thoughts, discussions and experimentation...
System Architecture Overview

CSIRO

- Website
- Forecasted Weather
- CSIRO Automatic Weather Stations

Google Cloud Platform

- Google Accounts
- Google App Engine
  - Google Endpoints
  - Google Datastore
  - Google Analytics
  - Google Drive
  - Google Memcache
- GEE API
  - REST/JSON

IrriSAT API (Google Endpoints)

Android
- Authentication
iOS
Web
Other Products

52 North SOS

Sensor Observation Service (SOS)

XML / O&M
Google Earth Engine

• Not to be confused with Google Earth
• Develop and run algorithms on large satellite imagery archives (Landsat, MODIS, etc)
• Access web services via restful API
  • Python
  • Javascript
• Runs in real-time on Google’s parallel processing platform
• Just-In-Time distributed model
• Ideal for IrriSAT
  • Define a field and analyse instantly
  • Data archive constantly being updated
  • No need to manage any data
Our approach for determining $K_c$

- Landsat Sources 30 m (operational missions)
  - Landsat 8 OLI
  - Landsat 7 ETM+
- 16 Days to orbit earth
- LS8 and LS7 offset 8 days from each other
- Combining provides full coverage every 8 days
Computing Kc (the workflow)

Orthorectified scenes

NIR - RED
NDVI = ----------
NIR + RED

LS8 OLI
NDVI

Remove Cloud Affected Areas

Backfill Missing Pixels

LS7 ETM+
NDVI

Remove Cloud Affected Areas

Blended NDVI

Compute Crop Coefficient (Kc)

Aggregate Paddock

SimpleCloudScore algorithm.
Better cloud / cloud shadow detection being developed (FMASK).

Developed using the GEE API

Linear Scaling for Cotton
Kc = NDVI x 1.36 - 0.086

Orthorectified scenes

NIR - RED
NDVI = ----------
NIR + RED

LS8 OLI
NDVI

Remove Cloud Affected Areas

Backfill Missing Pixels

LS7 ETM+
NDVI

Remove Cloud Affected Areas

Blended NDVI

Compute Crop Coefficient (Kc)

Aggregate Paddock

SimpleCloudScore algorithm.
Better cloud / cloud shadow detection being developed (FMASK).

Developed using the GEE API

Linear Scaling for Cotton
Kc = NDVI x 1.36 - 0.086
Computing Kc (spatial representation)

- Define fields
- Detect where clouds are present
- Need to address cloud shadows
- Remove pixels from NDVI
- Compute Kc
Generating time series data for a field

- Spatial algorithm can be reused in time domain
- Aggregation of pixels over a field (i.e., Kc, Field visibility)
- Again, execution occurs in real-time on Google’s servers
Google App Engine

- Cloud computing platform we are using to host IrriSAT
  - Scalable – Create new app instances when heavy loads occur
  - Cheap – free so far
- Platform as a Service (PlaaS)
  - App managed via a web console
  - No need to worry about operating system specifics, DBA’s etc
- Develop in: Python, Java, PHP, Go (Python for IrriSAT)
- Simple to transfer to industry if opportunity comes along again

![Developer Console](image-url)
Weather Station Data (ETo)

- Sensor Observation Service
  - Standardised Web service interface to
    - Query observation data
    - Sensor metadata
    - Features over the web
- OGC specification – XML/O&M/GML/SensorML Markups

- 3 Core methods
  - GetCapabilities - offers operations and endpoints as well as the available sensor data
  - GetObservation - observed values, including their metadata
  - DescribeSensor - provides sensor metadata (location, parameters, etc)

Enable Google Cloud Platform to access Weather Observations within CSIRO
Current Users

• Farmers (around Moree, NSW + southern QLD)
• Consultants (HMAg, Sundown Pastoral Company)
• DPI – Studying water use efficiency between different irrigation setups (spatial variation)
  • Spray
  • Drip, etc
• Software Engineers – building other products on top of IrriSAT API
Future challenges

• Complex irrigation regimes
  • Refine the accuracy and ability of the system to meet cotton growers’ needs
• Buffering?
• Multipolygon?
• Further algorithms to auto detect roads? Etc
• Decision Support Systems
  • Too complex = won’t use
  • Too simple = not useful
How users are actually using IrriSAT

“This saved me 4 months of work”...
Conclusions

• IrriSAT will provide *real time* crop water use at *broad scale* and *low cost*

• Work to date (enabling data services): $\text{ET}_c = \text{ET}_o \times K_c$

• Future work involves: refining the *accuracy*; incorporating *7 day forecast* of irrigation demand; and also *benchmarking* against nearby fields.

• Visit the IrriSAT website: [www.irrisat-cloud.appspot.com](http://www.irrisat-cloud.appspot.com)
Thank you

Land and Water Flagship
Jamie Vleeshouwer
Software Engineer

+61 7 3833 5589
jamie.vleeshouwer@csiro.au
www.csiro.au