Machine learning, Statistics and Big Data

3-day short course

Participants: Australian Bureau of Statistics

Presenters: Hugh Anderson, Jacinta Holloway, Miles McBain, James McGree, Chris McCool, Kerrie Mengersen

Queensland University of Technology

October 2017

Program: Day 1

Time	Presentation
9.30am – 10.00am	Registration and Coffee
10.00am – 12.00pm	Welcome and overview of course 1. Overview of big data 2. Overview of stats & ML for big data: concepts, philosophy, terminology 3. Overview of computational frameworks: from divide & recombine to cloud computing 4. Case Study: grading images
12.00pm – 12.45pm	Lunch
12.45am – 2.45pm	1. Preparing your data 2. Overview of methods 3. Overview of algorithms
2.45pm – 3.00pm	Break
3.00pm – 4.30pm	Digging Deeper: Classification and Regression. 1. Generalised linear regression 2. Spatial and time series models 3. Tree-based approaches: CART, RF, BRT, bagging boosting 4. Support vector machines
4.30pm – 5.00pm	Extended Topics: Classification and Regression Semi-parametric regression, KNNs, Ensembles, XGBoost Discussion of cloud computing Concluding remarks: Day 1

Program: Day 2

Time	Presentation
9.30am – 10am	Coffee
10am – 12.00pm	Brief recap and discussion Digging Deeper: Clustering and Dimension Reduction. 1. kmeans 2. Mixture models 3. Feature extraction 4. PCA, FA and extensions 5. Page Rank
12.00pm – 12.45pm	Lunch
12.45am – 2.45pm	Digging Deeper: Neural networks 1. Overview of NNs 2. Convolutional and recurrent NNs 3. Deep learning
2.45pm – 3.00pm	Break
3.00pm – 4.30pm	Extended topics: NNs NNs for time series and 2D images
4.30pm – 5.00pm	Extended topics: NNs Deep Learning Systems Concluding remarks: Day 2

Program: Day 3

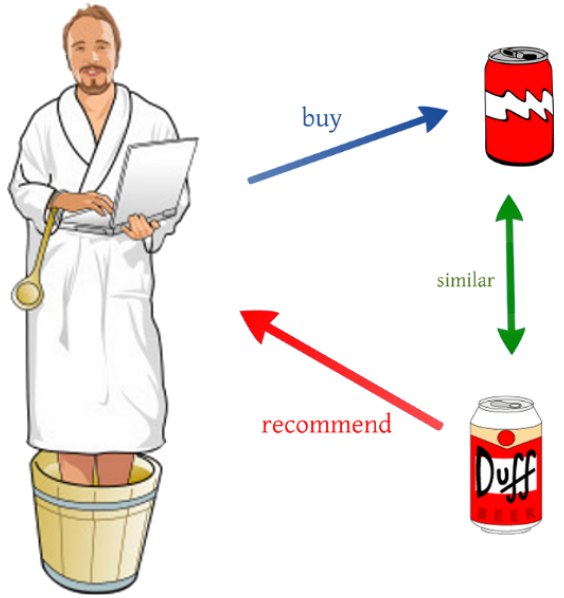
Time	Presentation
9.30am – 10.00am	Registration and Coffee
10.00am – 12.00am	Brief recap and discussion Case Study: Recommender systems. 1. Overview of recommender systems 2. Implementation 3. Use cases
12.00am – 12.45pm	Lunch
12.45pm – 2.45pm	The ABS context: Special Session. 1. Presentations from invited speakers 2. Discussion
2.45pm – 3.00pm	Break
3.00pm – 4.30pm	Extended Topics: 1. Overview of semi-supervised learning and ensembles of weak learners 2. Case study: return to classifying images
4.30pm – 5.00pm	Final issues Where to from here Concluding remarks: Day 3 Close

Day 3
Session 1

Case Study: Recommender systems

1. Overview of recommender systems
2. Implementation
3. Use cases

Recommender Systems



- Recommender systems apply statistical and knowledge discovery techniques to the problem of making product recommendations based on previously recorded data.

- The importance and the economic impact of research in this field is reflected by the Netflix Prize, a challenge to improve the predictions of Netflix's movie recommender system by more than 10% in terms of the root mean square error. The grand price of 1 million dollar was awarded in September 2009 to the Belcore Pragmatic Chaos team.

https://cran.r-project.org/web/packages/recommenderlab/index.html
https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Types of recommender systems

Content based:

If we can elicit the preference structure of a customer (user) concerning product (item) attributes then we can recommend items which rank high for the user's most desirable attributes. Typically, the preference structure can be elicited by analyzing which items the user prefers.

Examples: for movies the Internet Movie Database contains a wide range of attributes to describe movies including genre, director, write, cast, storyline, etc. For music, Pandora, a personalized online radio station, creates a stream of music via content-based recommendations based on a system of hundreds of attributes to describe the essence of music at the fundamental level including rhythm, feel, influences, instruments and many more

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Types of recommender systems

Collaborative filtering:

Given rating data by many users for many items (e.g., 1 to 5 stars for movies elicited directly from the users), one can predict a user's rating for an item not known to her or him or create for a user a so called top-N lists of recommended items.

The premise is that users who agreed on the rating for some items typically also agree on the rating for other items.

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Collaborative filtering

Collaborative filtering (CF) uses given rating data by many users for many items as the basis for predicting missing ratings and/or for creating a top-N recommendation list for a given user, called the active user. Formally, we have a set of users $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ and a set of items $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$. Ratings are stored in a $m \times n$ user-item rating matrix $\mathbf{R} = (r_{jk})$ where each row represents a user u_j with $1 \leq j \leq m$ and columns represent items i_k with $1 \leq k \leq n$. r_{jk} represents the rating of user u_j for item i_k. Typically only a small fraction of ratings are known and for many cells in $\mathbf{R}$ the values are missing. Many algorithms operate on ratings on a specific scale (e.g., 1 to 5 (stars)) and estimated ratings are allowed to be within an interval of matching range (e.g., $[1, 5]$). From this point of view recommender systems solve a regression problem.

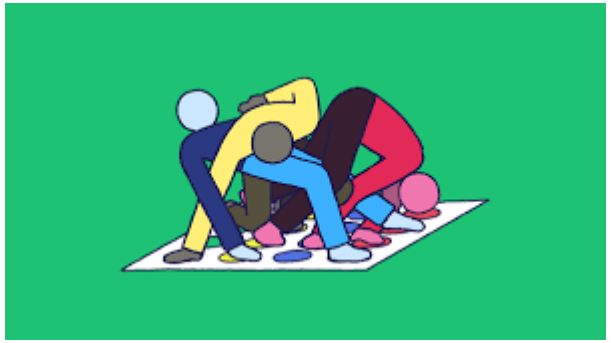
https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Collaborative filtering

The aim of collaborative filtering is to create recommendations for a user called the active user $u_a \in \mathcal{U}$. We define the set of items unknown to user u_a as $\mathcal{I}_a = \mathcal{I} \setminus \{i_l \in \mathcal{I} | r_{al} = 1\}$. The two typical tasks are to predict ratings for all items in $\mathcal{I}_a$ or to create a list of the best N recommendations (i.e., a top-N recommendation list) for u_a. Formally, predicting all missing ratings is calculating a complete row of the rating matrix $\hat{r}_a$. where the missing values for items in $\mathcal{I}_a$ are replaced by ratings estimated from other data in $\mathbf{R}$. The estimated ratings are in the same range as the original rating (e.g., in the range $[1, 5]$ for a five star rating scheme).

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Collaborative filtering



Creating a top-N list (Sarwar, Karypis, Konstan, and Riedl 2001) can be seen as a second step after predicting ratings for all unknown items in $\mathcal{I}_a$ and then taking the N items with the highest predicted ratings. A list of top-N recommendations for a user u_a is an partially ordered set $T_N = (\mathcal{X}, \geq)$, where $\mathcal{X} \subset \mathcal{I}_a$ and $|\mathcal{X}| \leq N$ ($|\cdot|$ denotes the cardinality of the set). Note that there may exist cases where top-N lists contain less than N items. This can happen if $|\mathcal{I}_a| < N$ or if the CF algorithm is unable to identify N items to recommend. The binary relation $\geq$ is defined as $x \geq y$ if and only if $\hat{r}_{ax} \geq \hat{r}_{ay}$ for all $x, y \in \mathcal{X}$. Furthermore we require that $\forall_{x \in \mathcal{X}} \forall_{y \in \mathcal{I}_a} \quad \hat{r}_{ax} \geq \hat{r}_{ay}$ to ensure that the top-N list contains only the items with the highest estimated rating.

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Types of collaborative filtering

- Model-based CF

Use the whole database online to create individual recommendations. Disadvantage: scalability.

- Memory-based CF

Use the whole (or at least a large sample of the) user database to learn a more compact model (e.g. clusters with users of similar preferences) that is later used to create recommendations

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

CF algorithms

User-based CF

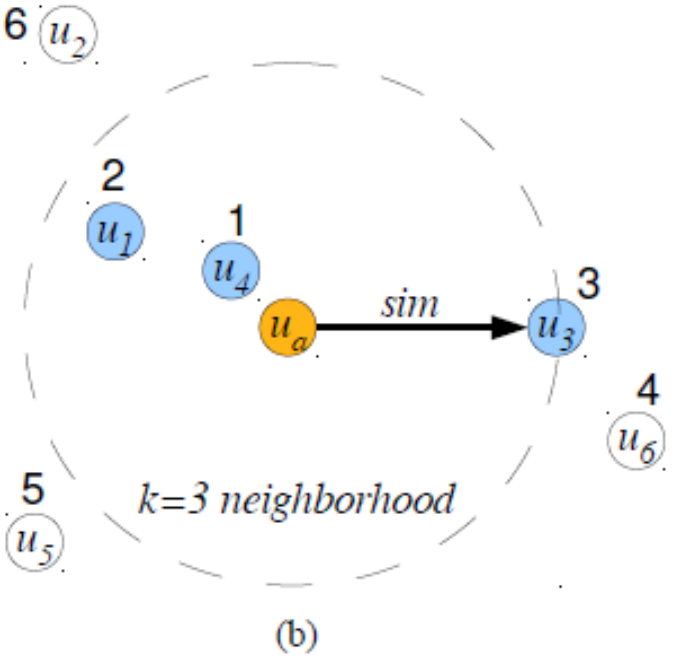
- Analyses ratings from many individuals. Assumption: users with similar preferences will rate items similarly.
- Thus missing ratings for a user can be predicted by first finding a *neighbourhood* of similar users and then aggregate the ratings of these users to form a prediction.
- Popular similarity measures between two users are the Pearson correlation coefficient and Cosine similarity.
- The neighbourhood for the active user can be selected by either a threshold on the similarity or by taking the *k* nearest neighbours.
- Once the users in the neighbourhood are found, their ratings are aggregated (e.g. via a weighted average) to form the predicted rating for the active user.

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Example

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	?	4.0	4.0	2.0	1.0	2.0	?	?
u_2	3.0	?	?	?	5.0	1.0	?	?
u_3	3.0	?	?	3.0	2.0	2.0	?	3.0
u_4	4.0	?	?	2.0	1.0	1.0	2.0	4.0
u_5	1.0	1.0	?	?	?	?	?	1.0
u_6	?	1.0	?	?	1.0	1.0	?	1.0
u_a	?	?	4.0	3.0	?	1.0	?	5.0
$\hat{r}_a$	3.5	4.0			1.3		2.0	

(a)



(b)

2D representation of similarities (users with higher similarity are displayed closer) with the active user in the centre.

The *k=3* nearest neighbours are selected and marked in the database to the left.

To generate an aggregated estimated rating, we compute the average ratings in the nieghbourhood for each item not rated by the active users.

To create a top-N recommendation list, the items are ordered by predicted rating.

Can improve the algorithm by removing user rating bias and variance (e.g. by Z-score normalisation of the rating data)

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

User-based CF

Drawbacks

- the whole user database has to be kept in memory
- expensive similarity computation between the active user and all other users in the database has to be performed

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

CF algorithms

Item-based CF

- Model-based approach which produces recommendations based on the relationship between items inferred from the rating matrix.
- Assumption: users will prefer items that are similar to other items they like.
- Calculate a similarity matrix containing all item-t-o-item similarities using a given similarity measure (eg Pearson correlation, Cosine similarity).
- For each item store only a list of the *k* most similar items and their similarity values. This improves the space and time complexity significantly but potentially sacrifices some recommendation quality.
- To make a recommendation, use similarities to calculate a weighted sum of the user's ratings for related items.

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Example

S	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	$\hat{r}_a$	$k=3$
i_1	-	0.1	0	**0.3**	**0.2**	**0.4**	0	0.1	-	
i_2	0.1	-	**0.8**	**0.9**	0	**0.2**	0.1	0	0.0	
i_3	0	**0.8**	-	0	**0.4**	0.1	0.3	**0.5**	4.6	
i_4	**0.3**	**0.9**	0	-	0	0.1	0	**0.2**	3.2	
i_5	**0.2**	0	**0.4**	0	-	0.1	**0.2**	0.1	-	
i_6	**0.4**	**0.2**	0.1	**0.3**	0.1	-	0	0.1	2.0	
i_7	0	**0.1**	**0.3**	0	**0.2**	0	-	0	4.0	
i_8	**0.1**	0	**0.5**	**0.2**	0.1	0.1	0	-	-	
u_a	2	?	?	?	4	?	?	5		

N=8 items, k=3

For the similarity matrix S only the k=3 largest entries are stored per row.

Assume we have ratings for the active user for items i1, i5, i8.

We can now compute the weighted sum using the similarities (using the reduced matrix with k=3 highest ratings) and the user's ratings.

The result below the matrix shows that i3 has the highest estimated rating for the active user.

Can also improve by normalising the user ratings.

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

User-based CF

Advantages

- More efficient than user-based CF since the model (reduced similarity matrix) is relatively small (N x k) and can be fully precomputed.
- Only a small loss of accuracy
- Used in large scale recommender systems, eg Amazon

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

But wait! There's more!

- **Collaborative filtering**: makes predictions of what might interest a person based on the taste of many other users. It assumes that if person X likes Snickers, and person Y likes Snickers and Milky Way, then person X might like Milky Way as well.
- **Content-based filtering**: focuses on the products themselves and recommends other products that have similar attributes. Content-based filtering relies on the characteristics of the products themselves, so it doesn't rely on other users to interact with the products before making a recommendation.
- **Demographic based recommender system**: categorizes users based on a set of demographic classes. This algorithm requires market research data to fully implement. The main benefit is that it doesn't need history of user ratings.
- **Utility-based recommender system:** makes recommendations based on a computation of its usefulness for each individual user. This relies on each industry's ability to decide on a user-specific utility function. The main advantage of this system is it can make recommendation that are unrelated product's attributes, such as availability and vendor reliability.
- **Knowledge based Recommender System**: makes suggestions based on information relating to each user's preferences and needs. Using function knowledge it can draw connections between a customer's need and a suitable product.
- **Hybrid filtering**: can implement a combination of any two of the above systems.

https://www.techemergence.com/use-cases-recommendation-systems/

Systematic review of recommender system methods

Total retrieved		93
Reason	Publications	Total
Conference / Proceedings entries		11
Not about RS	[23] [29] [39] [53] [71] [78] [81] [95]	8
Not describing a case study	[57] [90]	2
Not sufficiently describing the approach	[13] [16] [66]	3
Not able to access	[7] [31] [41] [82]	4
Other language	[93]	1
Unreadable	[64]	1
Duplicated	[69]	1
Book	[50]	1
Publications retained	[1] [4] [5] [8] [9] [10] [11] [14] [15] [17] [18] [19] [20] [21] [22] [24] [25] [26] [28] [30] [32] [35] [37] [40] [42] [43] [44] [45] [46] [47] [48] [49] [51] [52] [54] [56] [58] [59] [60] [61] [62] [63] [65] [67] [68] [70] [73] [74] [75] [77] [79] [80] [83] [84] [85] [86] [87] [88] [90] [92] [94]	61

https://arxiv.org/ftp/arxiv/papers/1511/1511.05262.pdf

Extensions

Consider the situation in which no large amount of detailed directly elicited rating data is available.

Eg: we can easily record in a supermarket what items a customer purchases, but not why other other products were not purchased.

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Evaluation of recommender algorithms

Given a rating matrix R, first partition the users (rows) into training and test sets. Learn the model on the training set and evaluate on the test set.

Approaches to splitting the data:

- Splitting: randomly assign a predefined proportion of the users to the training set and all others to the test set.
- Bootstrap sampling: sample from the test set with replacement to create the training set and use the users not in the training set as the test set. Good for smaller datasets (creates larger training sets and still have users left for testing).
- k-fold cross validation: split U into k sets (called folds) of approximately the same size. Evaluate k times, using one fold for testing and all other folds for learning. Average the k results. Obtain more robust results and error estimates.

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Evaluation of predicted ratings

Compute the deviation of the prediction from the true value

- mean average error MAE
- root mean squared error (RMSE)

RMSE penalises larger errors more strongly than MAE and thus is suitable for situations where small prediction errors are not very important.

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Evaluation of Top-N recommendations

Confusion matrix

How many items recommended in the top-N lists (d+b) were withheld items and thus correct recommendations (d) and how many were potentially incorrect (b); how many of the not recommended items (a+c) should have actually been recommended since they represent withheld items (c)

- Accuracy: (a+d)/(a+b+c+d)
- Mean absolute error (deviation): (b+c)/(a+b+c+d)
- Precision: (d)/(b+d)
- Recall: (d)/(c+d)

actual / predicted	negative	positive
negative	a	b
positive	c	d

https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Day 3
Session 1

Case Study: Recommender systems

1. Overview of recommender systems
2. Implementation
3. Use cases

Implementation

Many packages, e.g.:

- Apache Mahoot
  http://mahout.apache.org/

- R package: recommenderLab

https://cran.r-project.org/web/packages/recommenderlab/index.html

Both of these use collaborative filtering.

Day 3
Session 1

Case Study: Recommender systems

1. Overview of recommender systems
2. Implementation
3. Use cases

In business…

"At Amazon.com, we use recommendation algorithms to personalize the online store for each customer. The store radically changes based on customer interests, showing programming titles to a software engineer and baby toys to a new mother," explain Greg Linden, Brent Smith, and Jeremy York in their paper Amazon.com Recommendations: Item-to-Item Collaborative Filtering.

35% of Amazon.com's revenue is generated by its recommendation engine

https://www.techemergence.com/use-cases-recommendation-systems/

More business…

According to a paper written by Netflix executives Carlos A. Gomez-Uribe and Neil Hunt, the video streaming service's AI recommendation system saves the company around $1 billion each year.

According to McKinsey, 75 percent of what users watch on Netflix come from product recommendations.

Youtube video explains their approach:
https://www.youtube.com/watch?v=hqFHAnkSP2U

https://www.techemergence.com/use-cases-recommendation-systems/

And still more business…

- Spotify
- Best Buy
- YouTube

etc

https://www.techemergence.com/use-cases-recommendation-systems/

Systematic review of recommender system methods

Domain	Publications	Number of projects
Movies	[17] [18] [28] [32] [37] [42] [51] [52] [54] [62] [63] [65] [68] [77] [79] [85] [92]	17
Technical (Software Engineering)	[8] [14] [24] [25] [25] [25] [25] [25] [25] [25] [28] [30] [40] [56] [73]	15
Academic / Professional	[17] [44] [49] [59] [74] [75] [77] [86] [87]	9
Tourism	[4] [9] [10] [48] [60]	5
Music	[17] [26] [58] [79]	4
Health / Medical	[1] [5] [35] [83]	4
Social	[45] [80] [90]	3
Hotel	[21] [22] [43]	3
Games	[25] [70]	2
Books	[61] [11]	2
Clothing	[19] [69]	2
Image processing	[25] [47]	2
Transportation	[46]	1
TV	[94]	1
Emotion	[84]	1
Humor / Jokes	[20]	1
Real estate	[88]	1
Taxes	[15]	1

https://arxiv.org/ftp/arxiv/papers/1511/1511.05262.pdf

In Government…

- Personalised e-government services: Tourism recommender system framework https://link.springer.com/chapter/10.1007/978-3-642-22810-0_13

- Digital Communities in a Networked Society: e-Commerce, e-Business and e-Government. Edited by Manuel J. Mendes, Reima Suomi, Carlos (2003, Brazil)

- Recommender systems for e-Government in Smart Cities. http://arantxa.ii.uam.es/~cantador/doc/2017/citrec17egovernance.pdf

- Health
- Cultural heritage

etc

Day 3
Session 3

Extended Topics

1. Overview of semi-supervised learning and ensembles of weak learners

2. Case study: return to classifying images

Semi-supervised learning

- A class of **supervised learning** tasks and techniques that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data – to improve learning accuracy.

- Useful when the cost associated with a labeling process thus render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive.

https://en.wikipedia.org/wiki/Semi-supervised_learning

Semi-supervised learning approaches

- Generative models
- Low-density separation
- Graph-based models
- Heuristic approaches

https://en.wikipedia.org/wiki/Semi-supervised_learning

Ensembles of weak learners

“Ensemble modeling is the process of running two or more related but different analytical models and then synthesizing the results in order to improve the accuracy of predictive analytics and
data mining applications.”

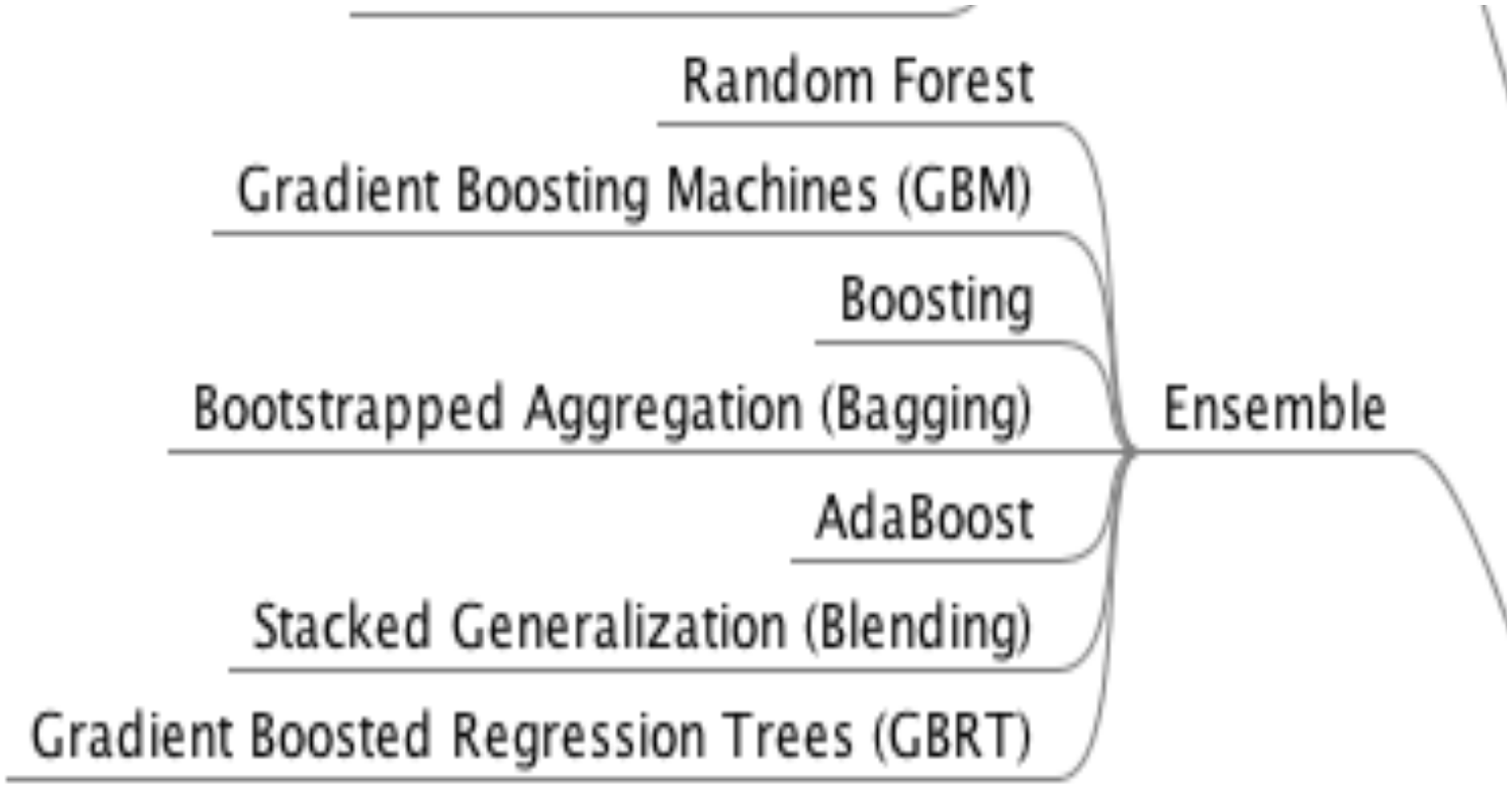
http://searchbusinessanalytics.techtarget.com/definition/Ensemble-modeling
https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

5 questions about ensemble modelling

1. What is an ensemble model?
2. What are bagging, boosting and stacking?
3. Can we ensemble multiple models of same ML algorithm?
4. How can we identify the weights of different models?
5. What are the benefits of ensemble model?

https://www.analyticsvidhya.com/blog/2015/09/questions-ensemble-modeling/

Ensemble models



https://s3.amazonaws.com/MLMastery/MachineLearningAlgorithms.png?__s=17oupmsuthrzaaubgepa

Day 3
Session 3

Extended Topics

1. Overview of semi-supervised learning and ensembles of weak learners

2. Case study: return to classifying images

ML methods for classifying satellite images

- Decision trees
- K-nn
- Neural Networks (NNs)
- Bayesian Networks
- Dimension reduction (PCA or independent Components Analysis) can be combined with any machine learning technique
- Support Vector Machines (SVMs)
- Regression trees (Gradient Boosted Machines, Random Forests and Boosted Regression Trees)

Boosted Regression Trees (BRTs)

- Iteratively create new trees trained using the residual errors of the previous steps
- Observations which are predicted poorly are given a higher weight for the next step in creating a new tree.
- It goes on successively to generate each new set of residuals until all trees have been created.

The result from the single steps in creating those trees is summed to give a successive accumulative model.

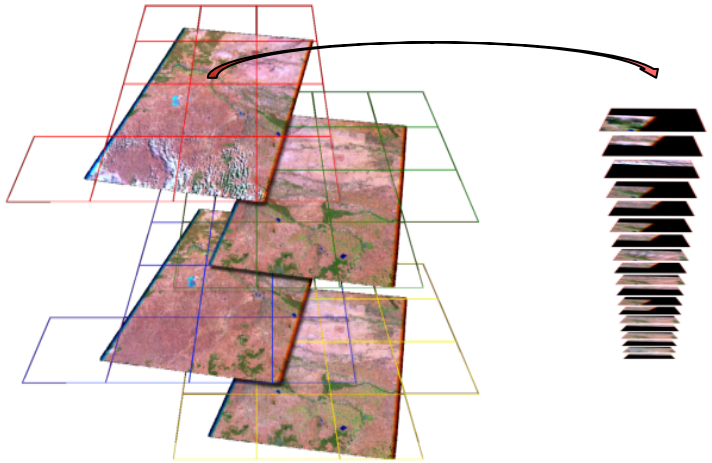
Shrinkage/learning rate: We can make smaller steps by only adding a fraction of the result given by each tree to achieve a more stable convergence towards the correct values.

Boosting combines existing weak learners in a new form so the classification process achieves higher accuracy, and therefore less errors.

Weak learners often have just one feature and are simply structured, but can be interpreted well and quickly.

Boosting combines all weak classifiers with a weighting factor. Basically the residuals will be used to fit a new model to it.

Big Data Case Study: Landsat Satellite Imagery data



www.datacube.org.au

Data specifications

Landsat 7 imagery data from the Australian Geoscience Data Cube (AGDC).

The Landsat imagery data in the Australian Geoscience Data Cube (AGDC) is stored in one-degree latitude and longitude tiles. Each tile is approximately 100km x 100km, with each pixel representing a 25m x 25m area of land.

Each pixel has six reflectance measurements associated with it, corresponding to six distinct bandwidths on the electromagnetic spectrum, namely red, green, blue, near infra-red (NIR), short-wave infra-red (SWIR) 1 and SWIR 2. Landsat 7 revisits the same area of land, on average every 16 days, meaning that most pixels have a set of six reflectance measurements every 16 days.

ABS acquired satellite imagery data from the AGDC for the 2010-12 calendar years. The Landsat data was used for crop classification.

Big Data Case Study: Landsat Satellite Imagery data

Data processing

- Landsat data was split into a data set and a training set. Further, the “crop” column was chosen as a categorical response variable with a multinomial distribution. Out of the bag fraction was selected as “without replacement” to ensure unique values.
- In all scenarios the amount of trees was very limited (>20 iterations) due to the sheer volume of data. One result of eight iterations on the whole data set did not collapse and produced preliminary results. As a result it is necessary to either use a very limited amount of observations in the modelling process or use High Performance Computing (HPC) facilities.

Big Data Case Study:
Landsat Satellite Imagery data

Data quality issues:
2 approaches for quality control and assessment

1. Confusion matrix
2. Receiver Operating Characteristic (ROC) curve

ROC analysis is beneficial to assess cost/benefit analysis of diagnostic decision-making.

Big Data Case Study: Landsat Satellite Imagery data

In the confusion matrix the predicted values were mapped against the actual class. The confusion matrix is a visualisation of the performance of the algorithm by identifying four different options for results;

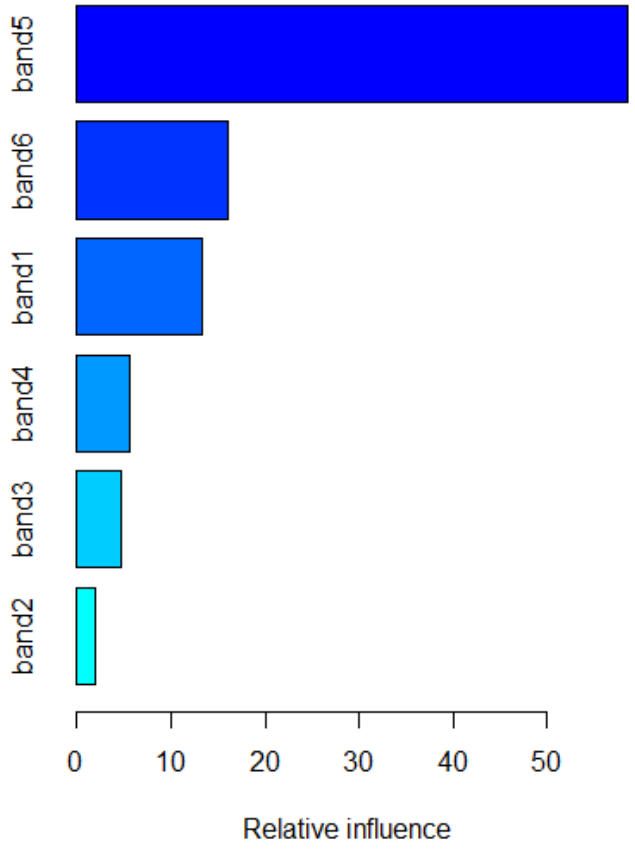
- A true positive (correctly classified)
- A true negative (a correct rejection)
- A false positive (has been not rejected but is not correct in this group) and;
- A false negative (has been rejected but should not have been since it actually belongs to this group).

Big Data Case Study: Landsat Satellite Imagery data

Results

In the preliminary results band number 5 of the Landsat imagery was identified as the strongest predictor.

Band 5 (1.55 - 1.75u m): is sensitive to the turgidity or the amount of water in plants. In other environmental studies band 5 has separated forest lands, croplands, and water body distinctly from each other. Forests have appeared as a comparatively darker tone than the croplands (light grey). Band 5 has separated water body (dark tone) from barren lands, croplands, and grass lands (lighter tone). Since urban area and croplands have responded almost in same spectral reflectance band 5 could not be able to separate these areas. See http://web.pdx.edu/~emch/ip1/bandcombinations.html.

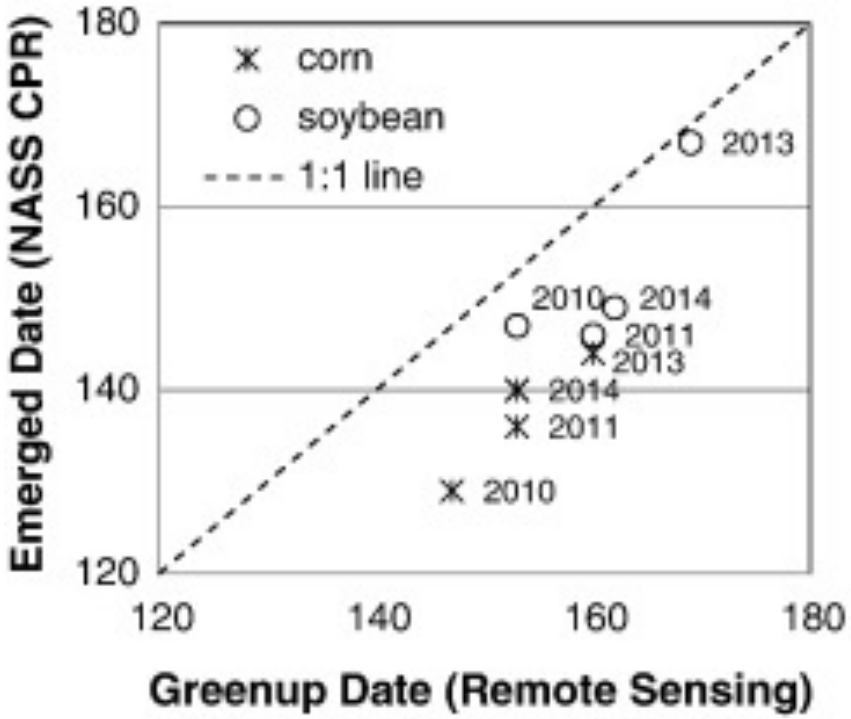


Relative influence of Landsat imagery bands

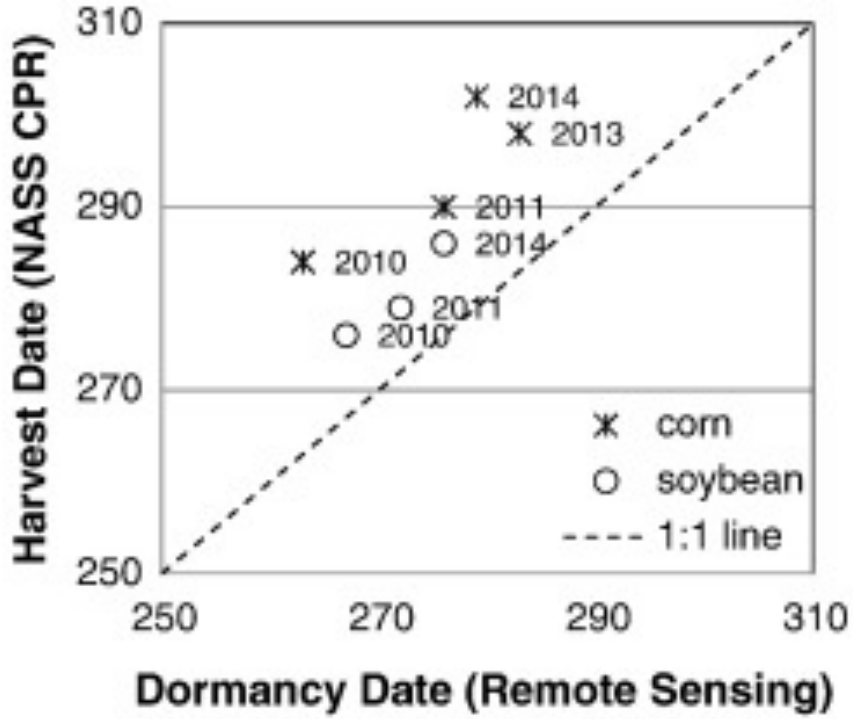
Experiences in USA

Gao *et al.* (2017) *Remote Sensing of the Environment*

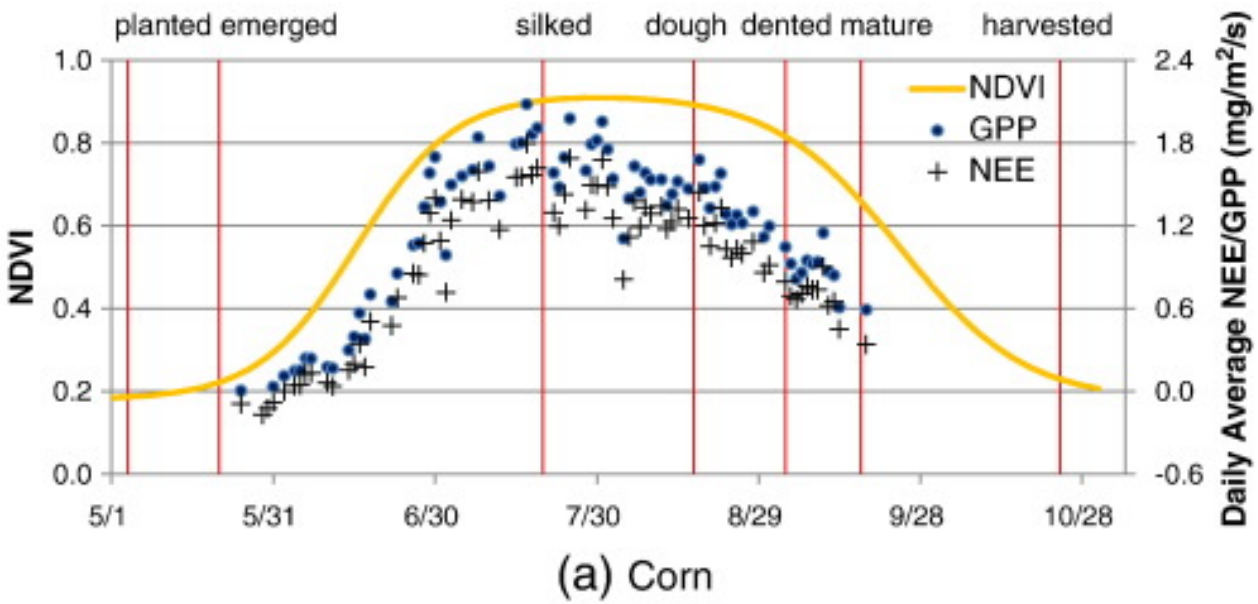
- Aim: Regionally monitor crop progress & condition through the growing season to benefit crop management & yield estimation.
- Currently: The USDA National Agricultural Statistical Service (NASS) report these metrics weekly at state or district levels using field observations, which are time consuming & subjective, and too coarse for some uses.
- Want to: Use remote sensing approaches for mapping crop phenology using vegetation index time-series generated by fusing Landsat and MODIS to improve temporal sampling over that provided by Landsat alone.
- Study:
  - The mean difference (bias) in NDVI between actual Landsat observations and the fused Landsat-MODIS data, generated for Landsat overpass dates, is in the range of − 0.011 to 0.028 for every year.
  - Strong correlations between remotely sensed phenological stages, based on NDVI curve inflection points, and the observed crop physiological growth stages from the NASS Crop Progress (CP) reports: green-up (2-4 leaves) and time to harvest.



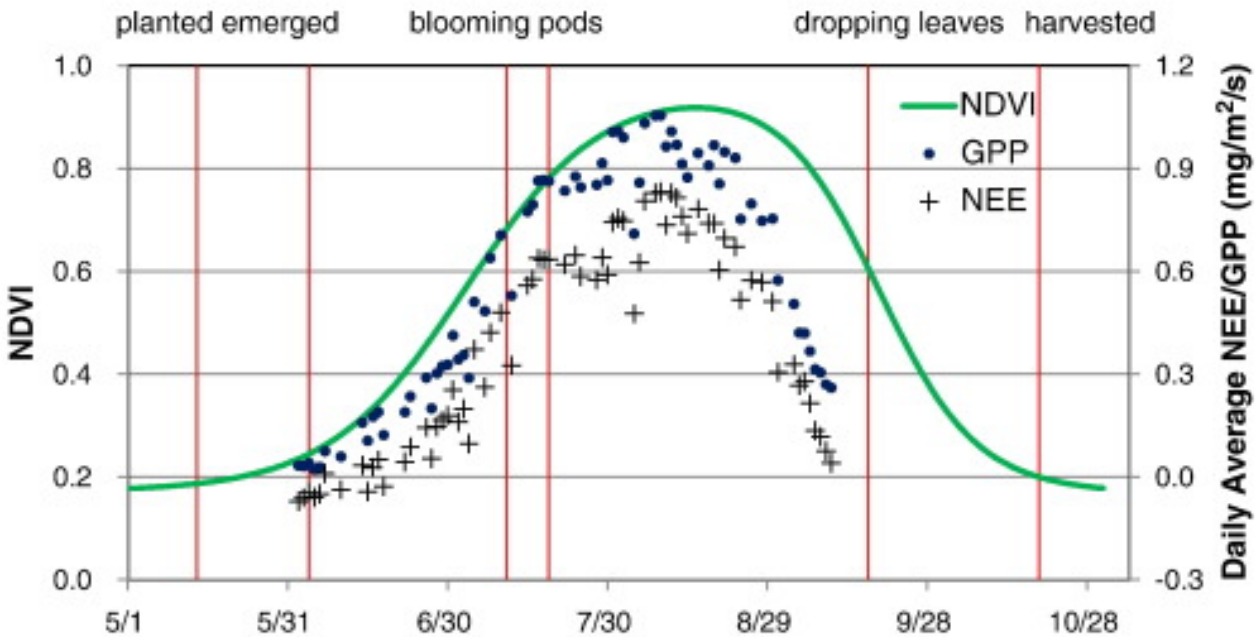
(a) green-up vs. emergence



(b) dormancy vs. harvest



(a) Corn



(b) Soybean

NEE: Net Ecosystem Exchange

GPP: Gross primary production

Experiences in South Korea

Yeong & Kim (2015), *Remote Sensing*

- Aim: How much does increased frequency of observation improve monitoring of crop development?
- Study:
  - Four year testing period (2010-2014) using satellite images from the world's first Geostationary Ocean Color Imager (GOCI) for spectral analyses of paddy rice in South Korea.
  - A vegetation index was calculated from GOCI data based on the bidirectional reflectance distribution function (BRDF)-adjusted reflectance, which was then used to visually analyze the seasonal crop dynamics. These vegetation indices were then compared with those calculated using the MODIS NDVI based on Nadir BRDF-adjusted reflectance.
- Results:
  - GOCI provided four times better temporal resolution than the combined MODIS sensors
  - Easier to find cloudless pixels and interpret subtle characteristics of the vegetation development
  - Ground spectral measurements from CROPSCAN were also compared with satellite-based vegetation products, showing a similar crop development pattern to the GOCI products.

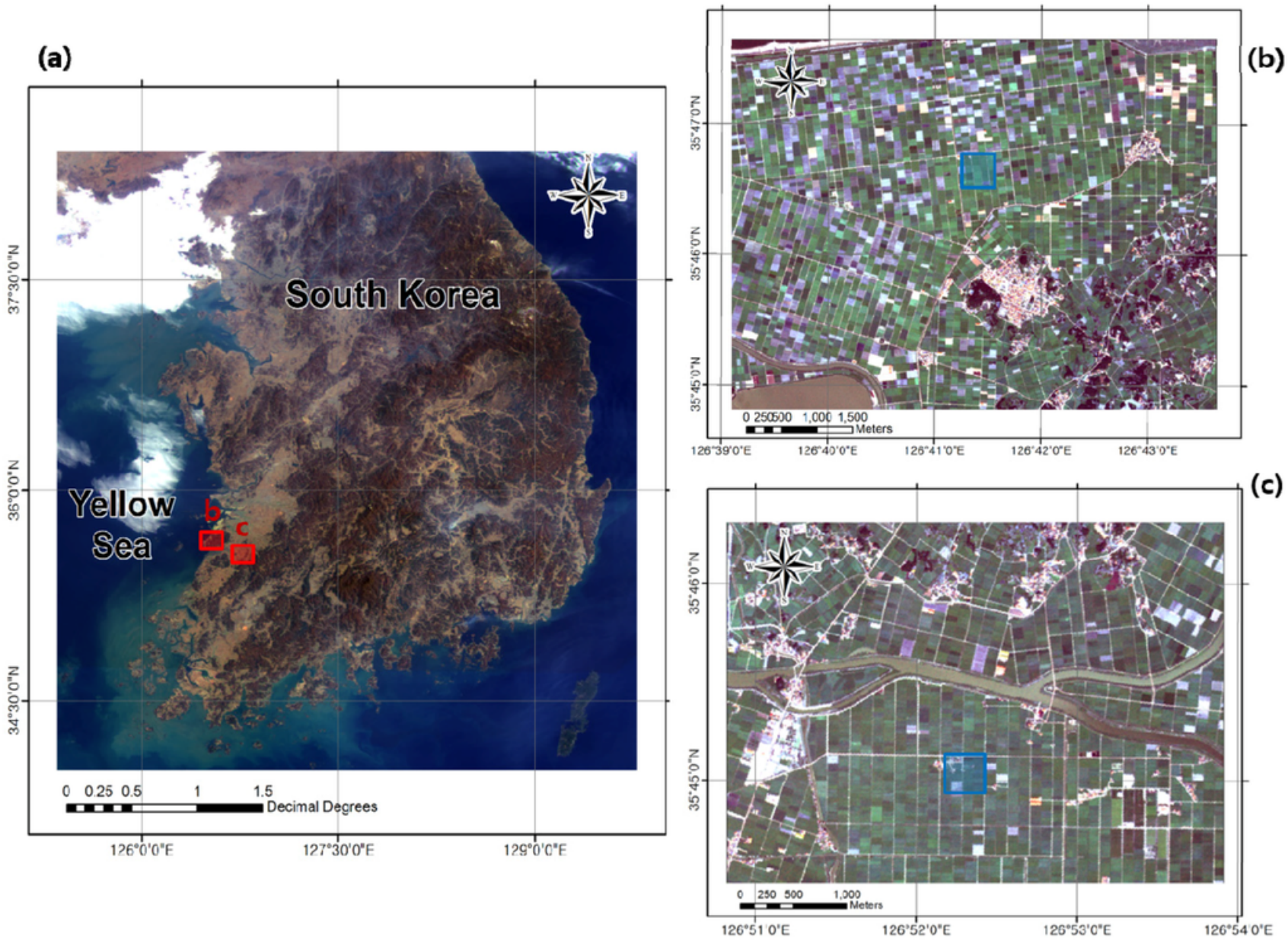
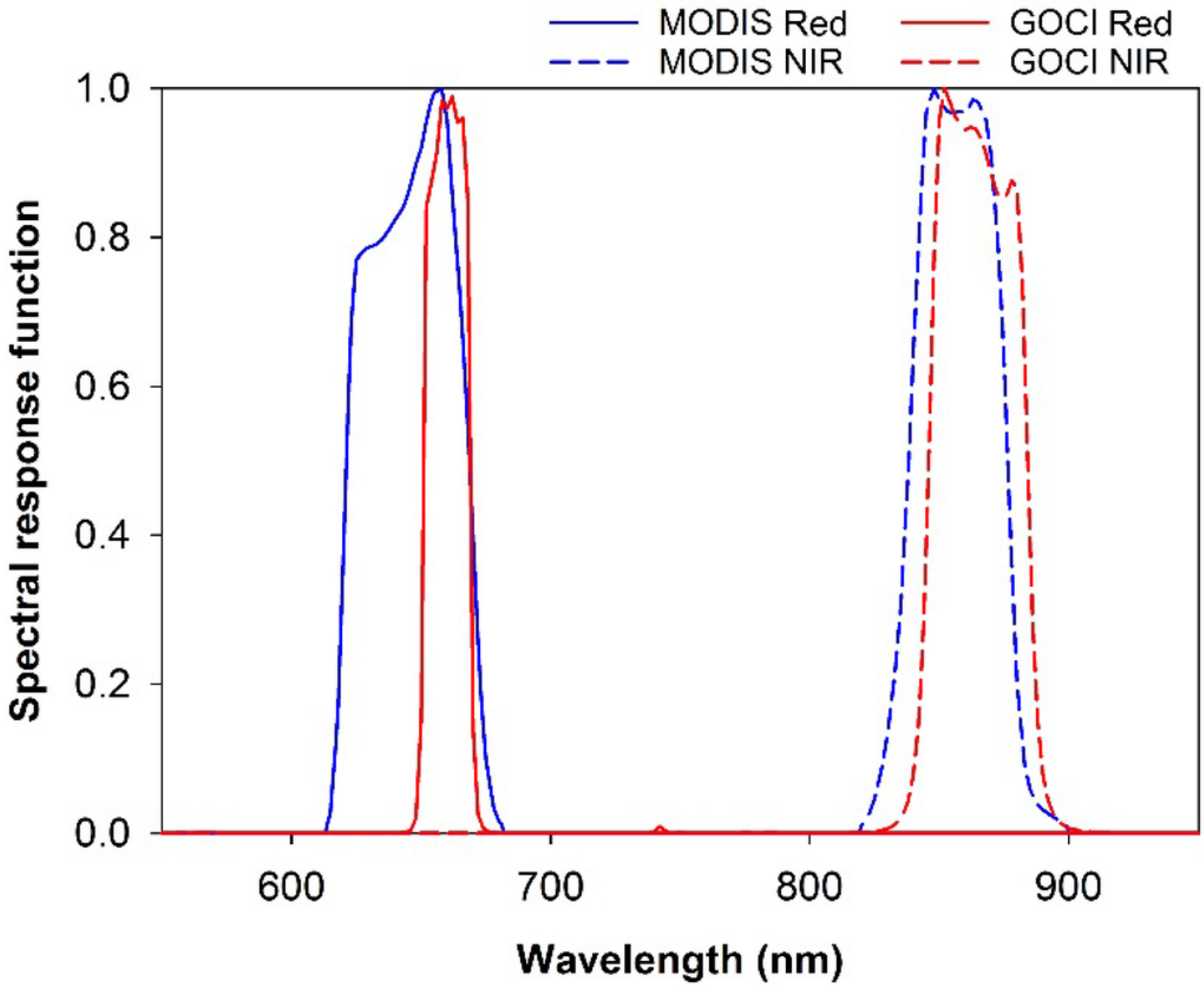

(a)
South Korea
Yellow Sea
b
c
37°30'0"N
36°0'0"N
34°30'0"N
126°0'0"E
127°30'0"E
129°0'0"E
0 0.25 0.5 1 1.5
Decimal Degrees
(b)
35°47'0"N
35°46'0"N
35°45'0"N
126°39'0"E
126°40'0"E
126°41'0"E
126°42'0"E
126°43'0"E
0 250 500 1,000 1,500
Meters
(c)
35°46'0"N
35°45'0"N
126°51'0"E
126°52'0"E
126°53'0"E
126°54'0"E
0 250 500 1,000
Meters

Table 1. Detailed characteristics of the GOCI and MODIS sensors used for estimating land-surface products.

Satellite Sensor	Orbit Type	Altitude	Wavelength	Spatial Resolution
GOCI	Geo-synchronous	≈36,000 km	B1: 402–422 nm	Approximately 500 m over South Korea area (≈390 m at nadir)
			B2: 433–453 nm	
			B3: 480–500 nm	
			B4: 545–565 nm	
			B5: 650–670 nm	
			B6: 675–685 nm	
			B7: 735–755 nm	
			B8: 845–885 nm	
MODIS	Sun-synchronous	≈705 km	B1: 620–670 nm	500 m at nadir
			B2: 841–876 nm	
			B3: 459–479 nm	
			B4: 545–565 nm	
			B5: 1230–1250 nm	
			B6: 1628–1652 nm	
			B7: 2105–2155 nm	


MODIS Red
MODIS NIR
GOCI Red
GOCI NIR
Spectral response function
1.0
0.8
0.6
0.4
0.2
0.0
600
700
800
900
Wavelength (nm)

Ground measurements were performed using the multispectral radiometer (MSR).
CROPSCAN MSR16 equipped with 16 spectral sensor bands in the 450–1750 nm region.

Growing season

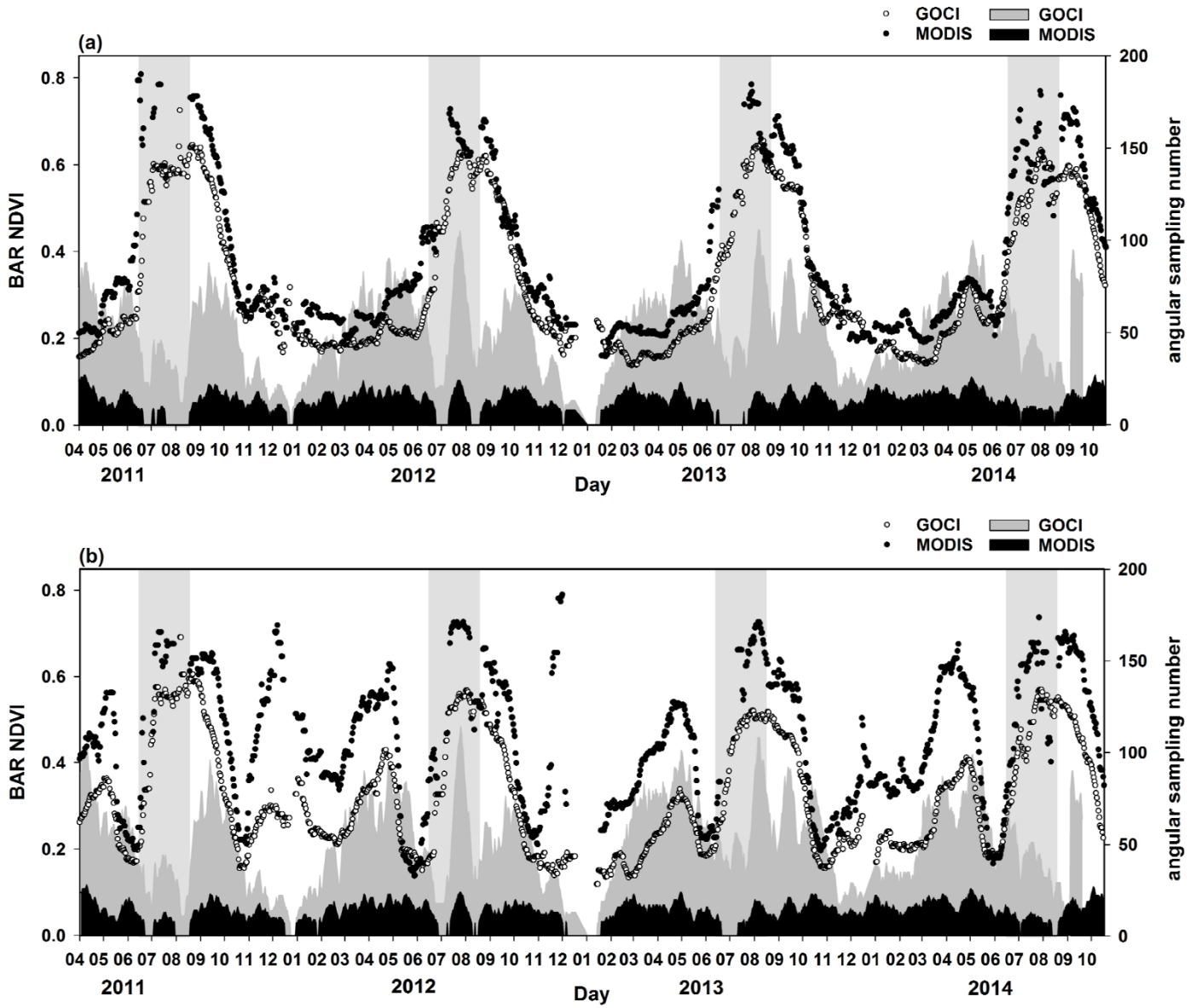
Growing season

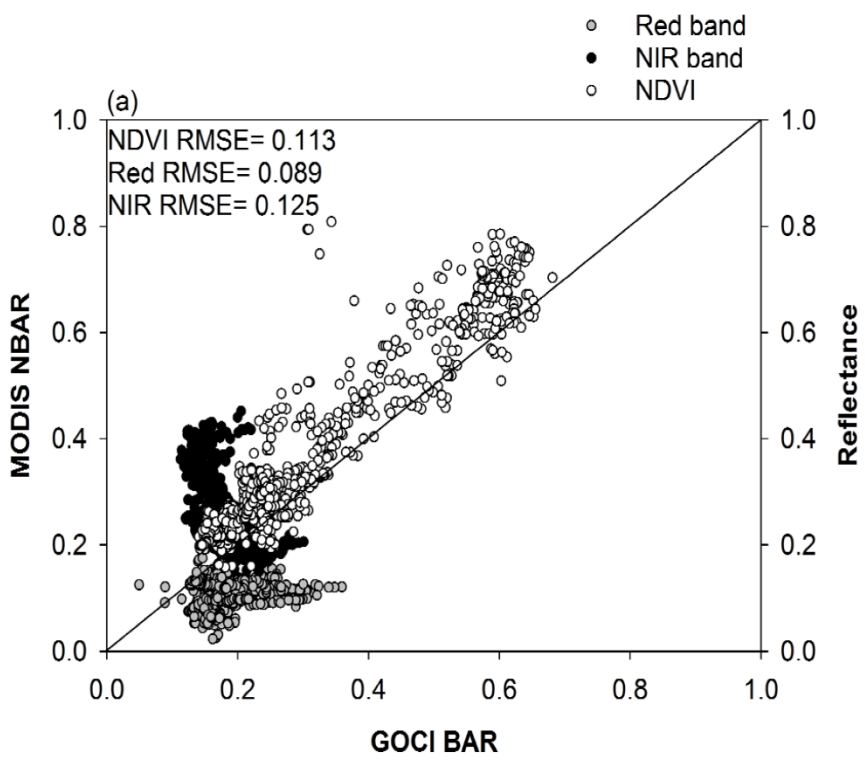
Earing season

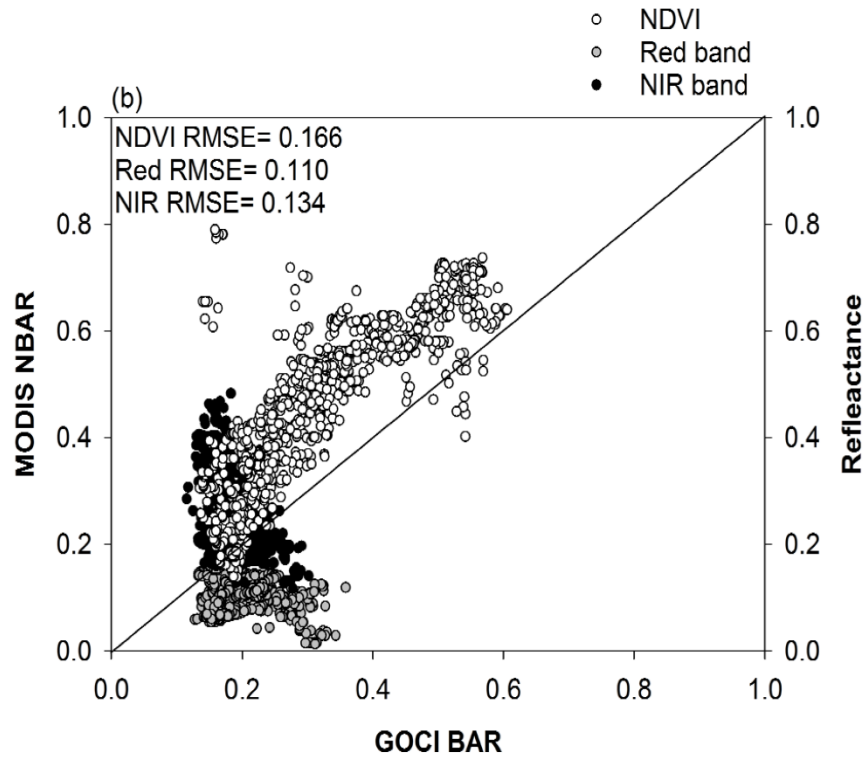
Heading season

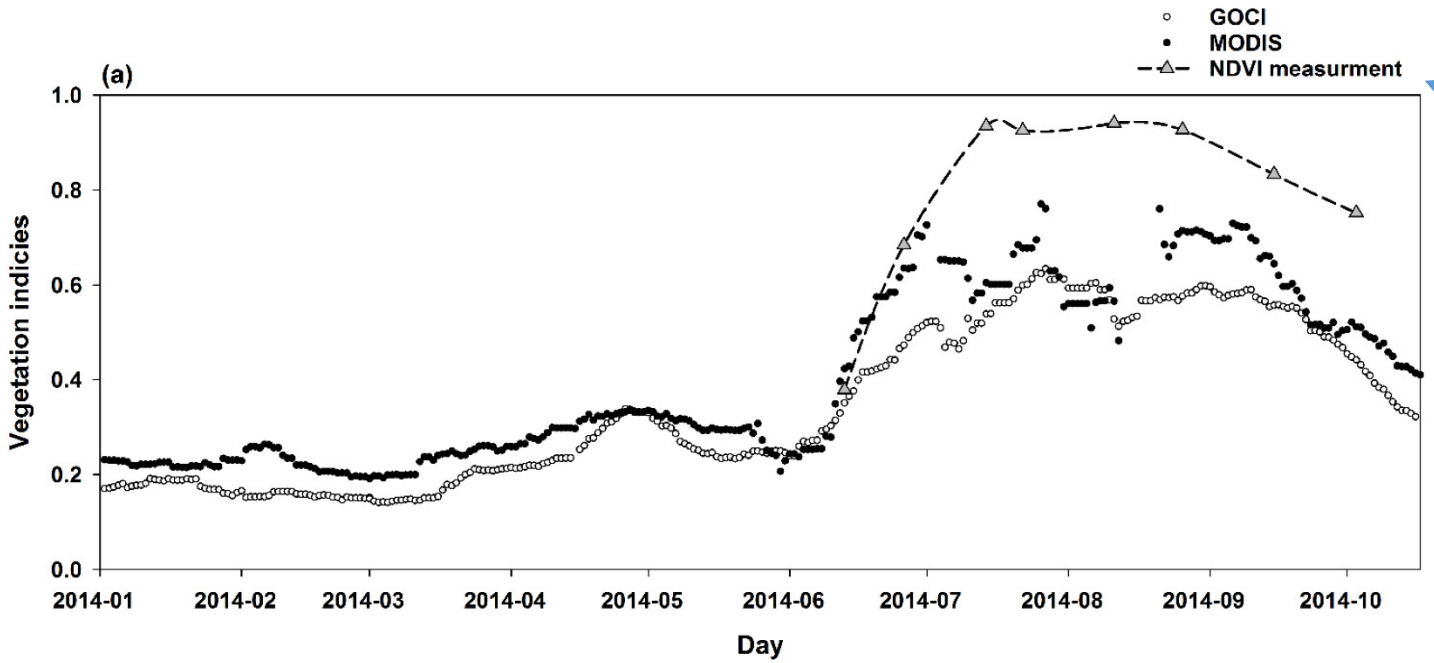
Harvest

(a) Intermediate-late-maturing rice paddy, (**b**) Early maturing rice paddy. Light gray bars are rainy summer seasons.

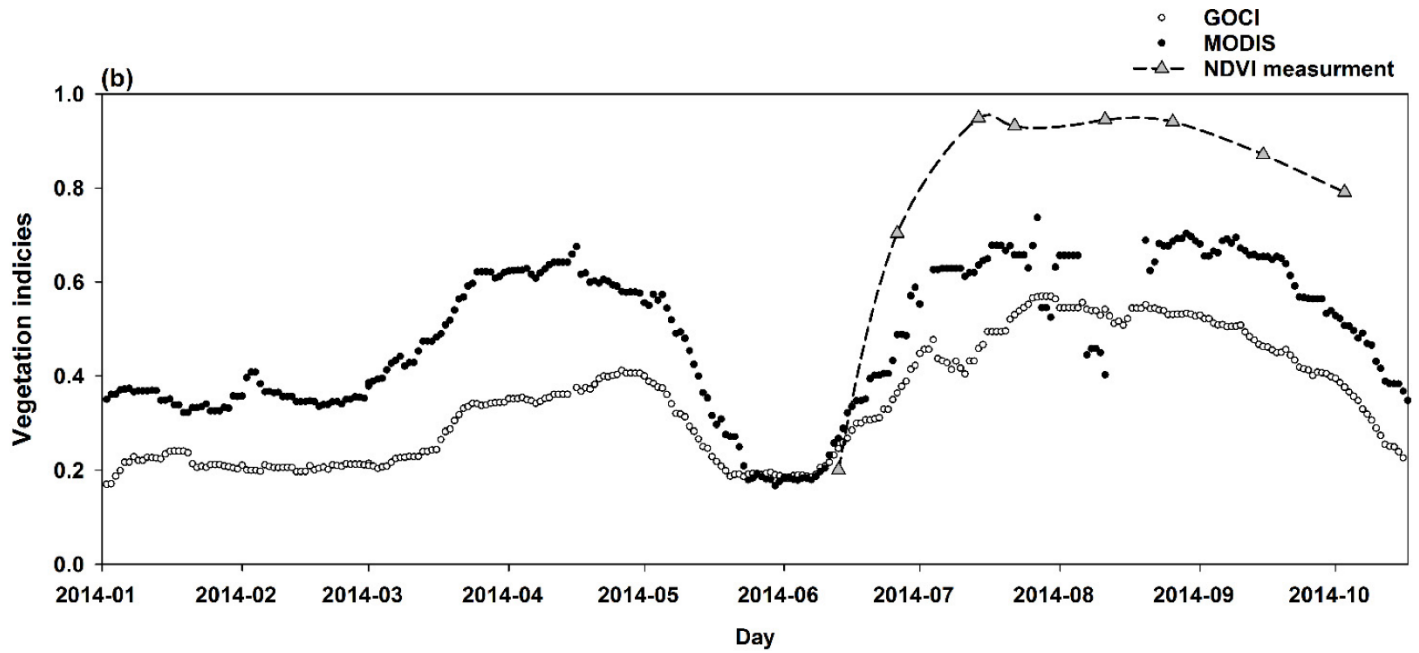



Red band
NIR band
NDVI
(a)
NDVI RMSE= 0.113
Red RMSE= 0.089
NIR RMSE= 0.125
MODIS NBAR
GOCI BAR
Reflectance
0.0
0.2
0.4
0.6
0.8
1.0


NDVI
Red band
NIR band
(b)
NDVI RMSE= 0.166
Red RMSE= 0.110
NIR RMSE= 0.134
MODIS NBAR
GOCI BAR
Refleactance
0.0
0.2
0.4
0.6
0.8
1.0



Dashed line is based on ground meas'ts (cubic spline)

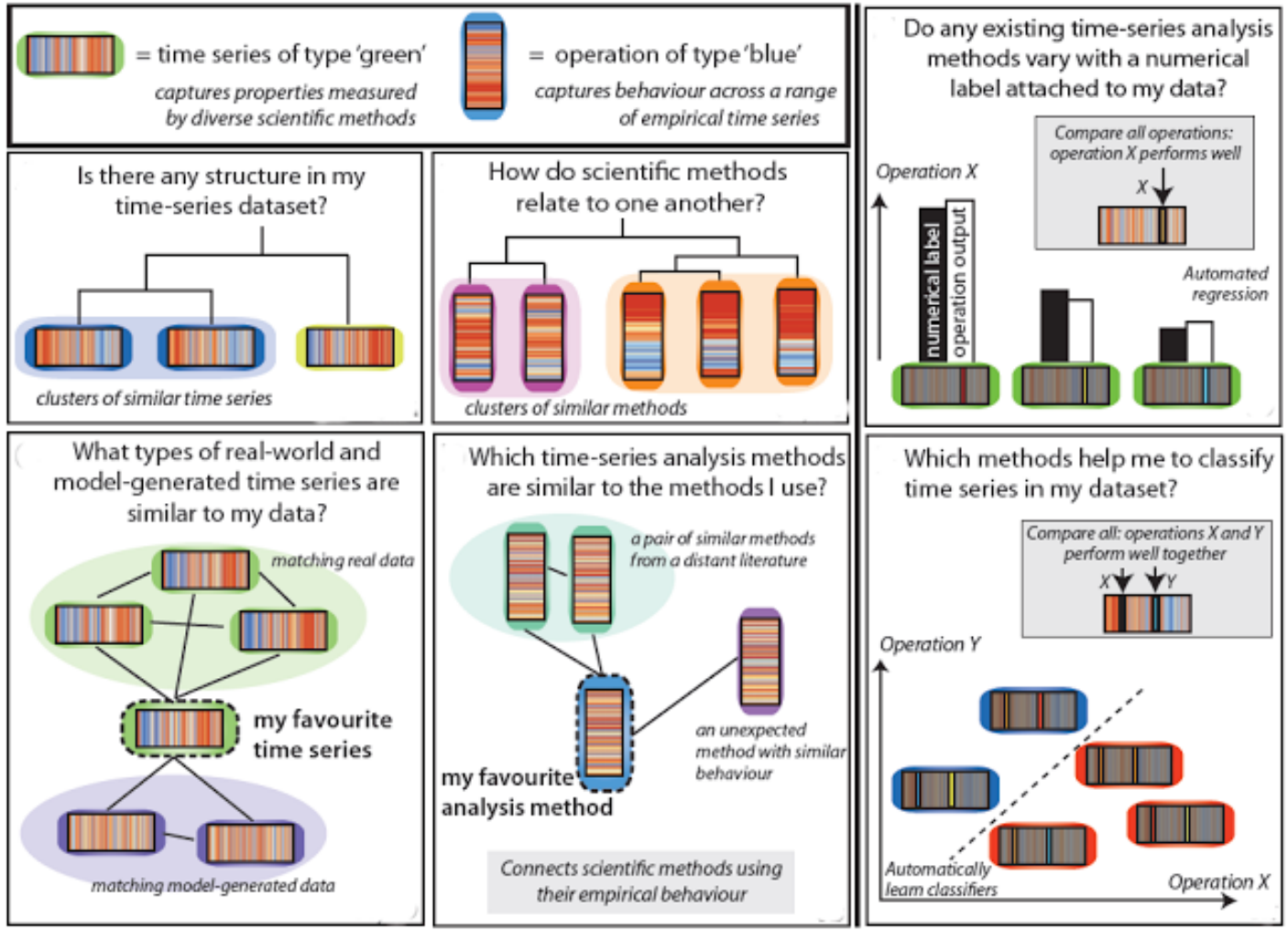


Time Series Modelling

1. Autoregressive, moving average and ARIMA models

2. State space models

Time series: taxonomy of aims and methods

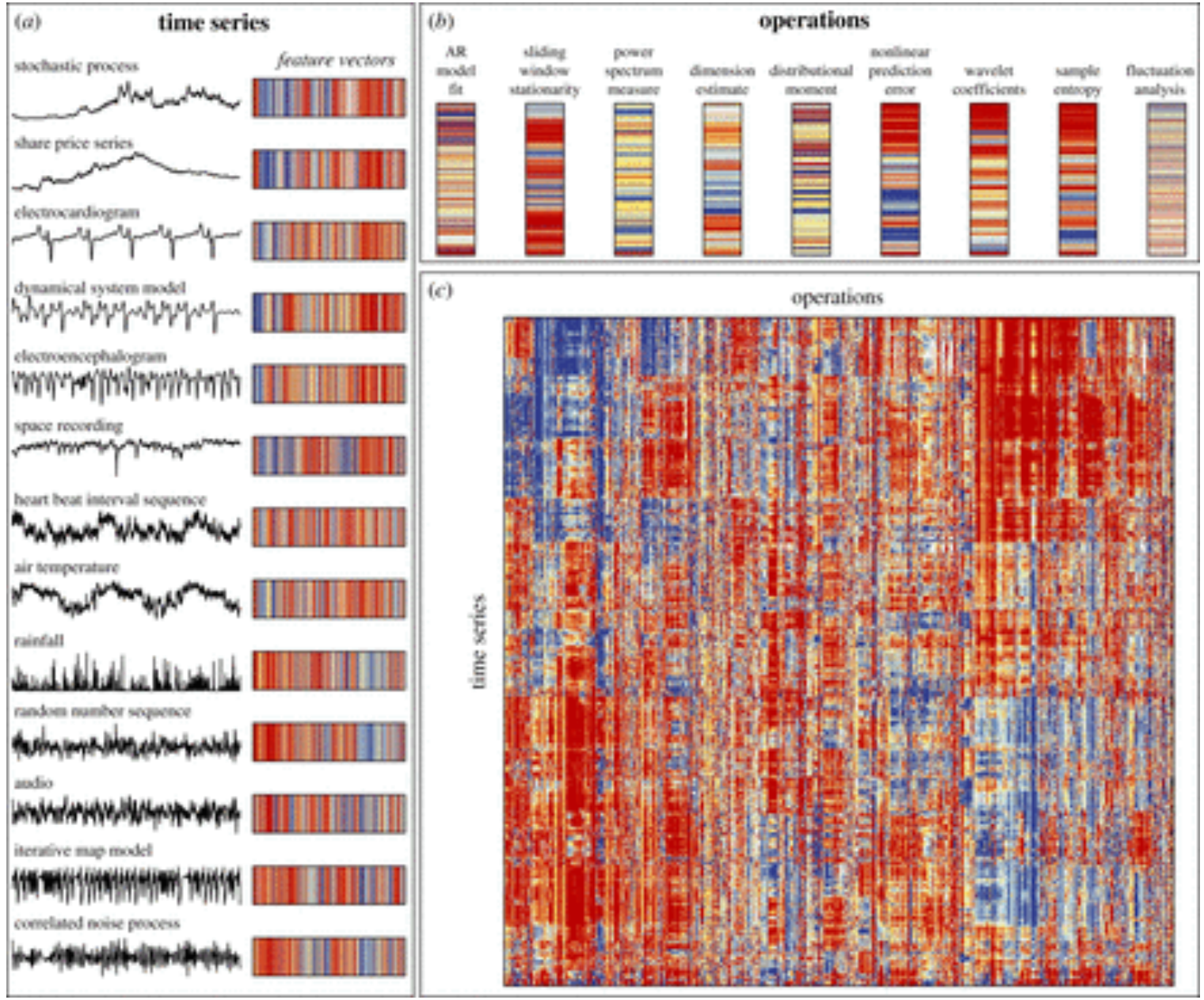


>9000 methods for analysing signals

>35 000 real-world & model-generated time series

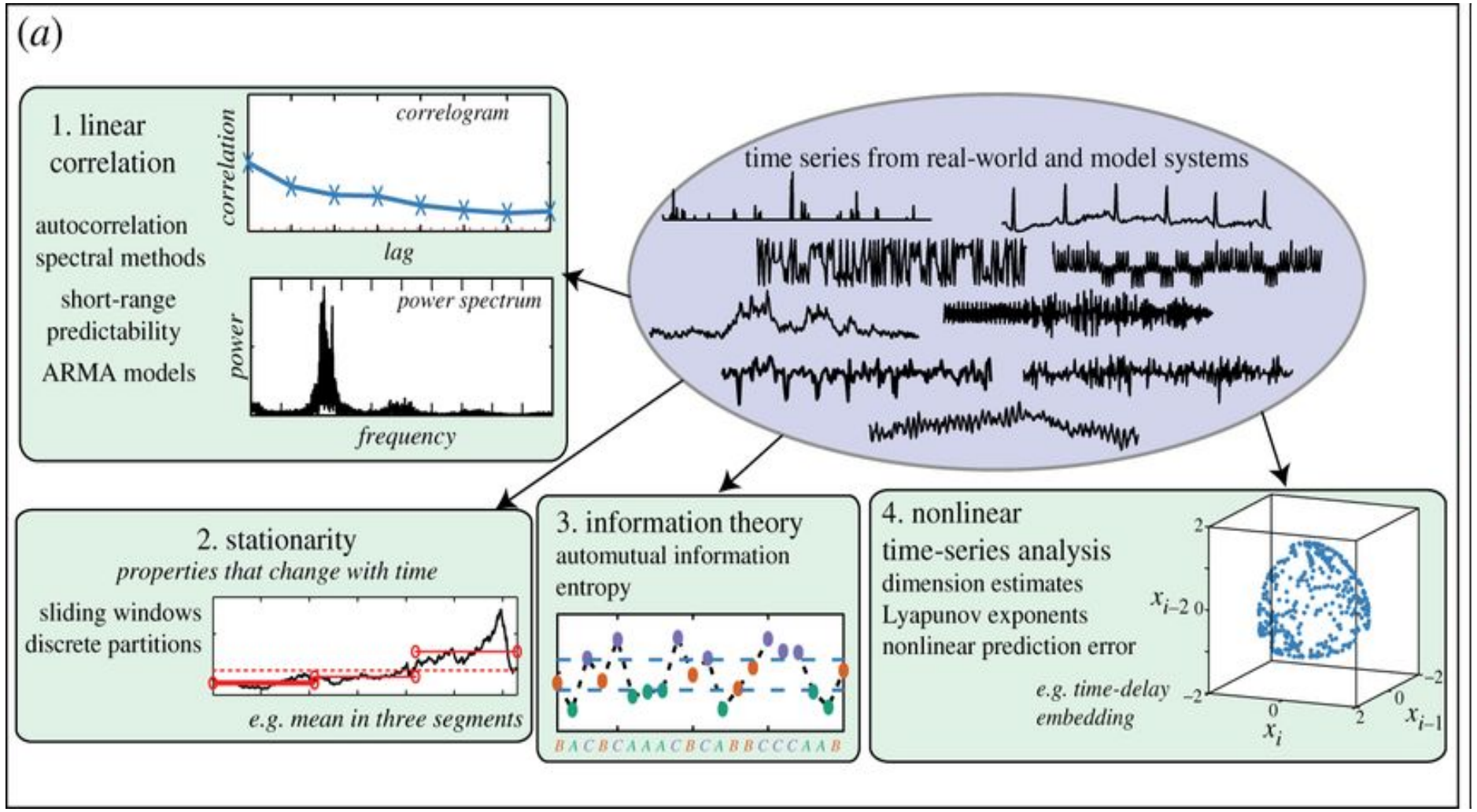
~ 200 types of methods

http://systems-signals.blogspot.com.au/2013/04/a-compound-methodological-eye-on.html



http://rsif.royalsocietypublishing.org/content/10/83/20130048

Cluster analysis of operations



http://rsif.royalsocietypublishing.org/content/10/83/20130048

Model based approaches: State space models

standard univariate DLM combines a normal linear observation equation,

$$y_t = \mathbf{F}_t \theta_t + \nu_t, \tag{1}$$

with a conditionally normal, multivariate linear system equation to govern the state evolutions of θ_t from time t to $t+1$,

$$\theta_{t+1} = \mathbf{G}_{t+1} \theta_t + \omega_{t+1}. \tag{2}$$

Model based approaches: State space models

Write as a structural time series (trend, seasonality, error terms):

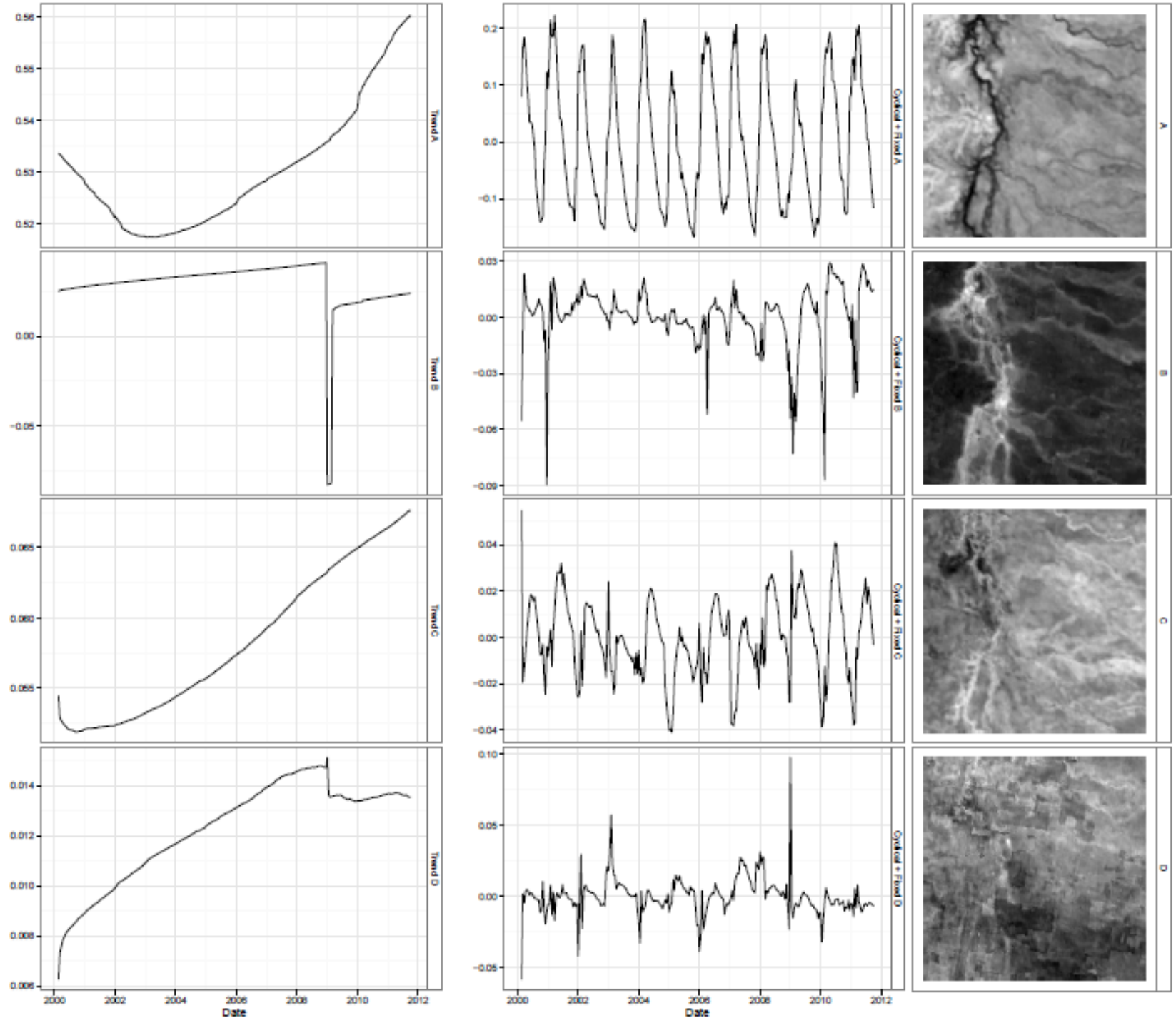
$$
\begin{aligned}
y_t &= \mu_t + \varepsilon_t, \quad & \varepsilon_t &\sim N(0, \sigma_\varepsilon^2) \\
\mu_{t+1} &= \mu_t + \nu_t + \xi_t, \quad & \xi_t &\sim N(0, \sigma_\xi^2) \\
\nu_{t+1} &= \nu_t + \varsigma_t, \quad & \varsigma_t &\sim N\left(0, \sigma_\xi^2\right)
\end{aligned}
$$

Model based approaches: State space models

Write as a state space model:

$$y_t = (1 \quad 0)\begin{pmatrix}\mu_t \\ \nu_t\end{pmatrix} + \varepsilon_t,$$

$$\begin{pmatrix}\mu_{t+1} \\ \nu_{t+1}\end{pmatrix} = \begin{bmatrix}1 & 1 \\ 0 & 1\end{bmatrix}\begin{pmatrix}\mu_t \\ \nu_t\end{pmatrix} + \begin{pmatrix}\xi_t \\ \varsigma_t\end{pmatrix}.$$


Trend A
Trend B
Trend C
Trend D
Cyclical + Fixed A
Cyclical + Fixed B
Cyclical + Fixed C
Cyclical + Fixed D
A
B
C
D
Date
Date

Time series regression

Gruber and West (2016): graphical dynamical linear models (SGDLMs) for forecasting and scalable multivariate volatility analysis

https://arxiv.org/pdf/1606.08291v1.pdf

These involve:

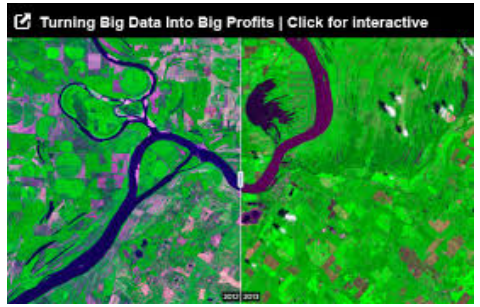
(i) A set of decoupled univariate dynamic linear models for individual series

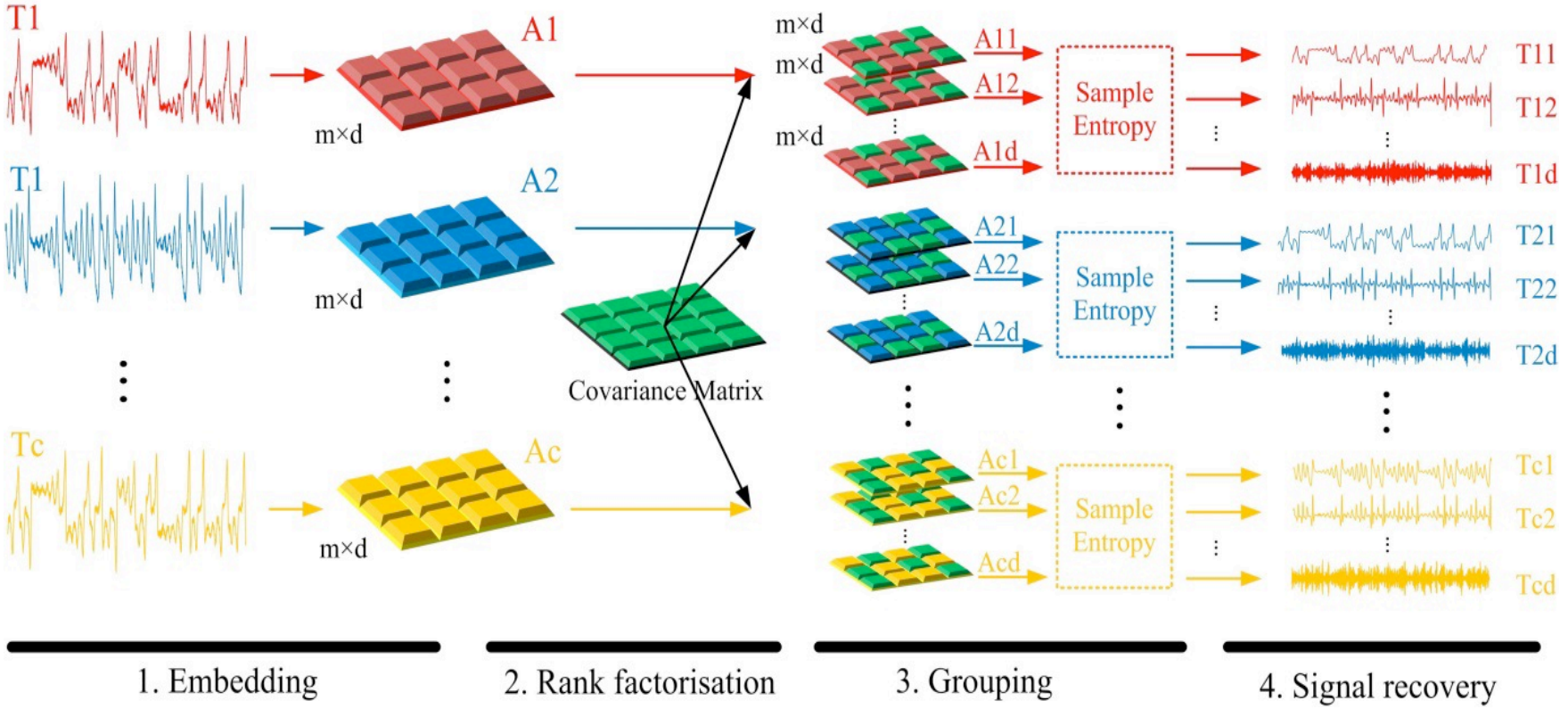
(ii) Sparse graphical modelling to recouple the series

(iii) (iii) variational Bayesian methods combined with importance sampling to integrate/couple the series for forecasting and decisions.

(iv) Parallel, GPU-based implementation enables on-line analysis of increasingly high-dimensional time series

High-D time series convolution

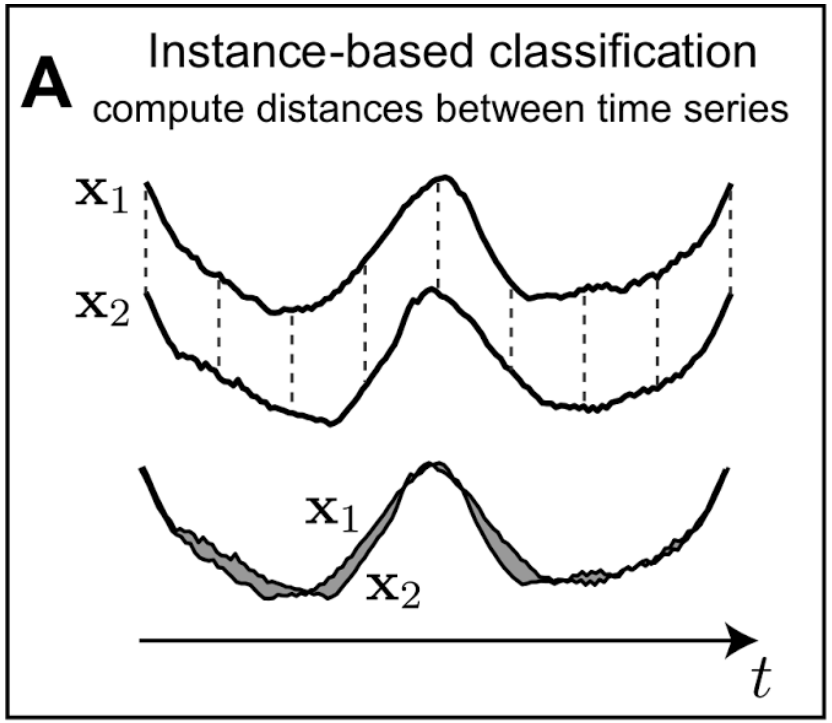




Hongbo Xie

Comparing curves: Alignment matching

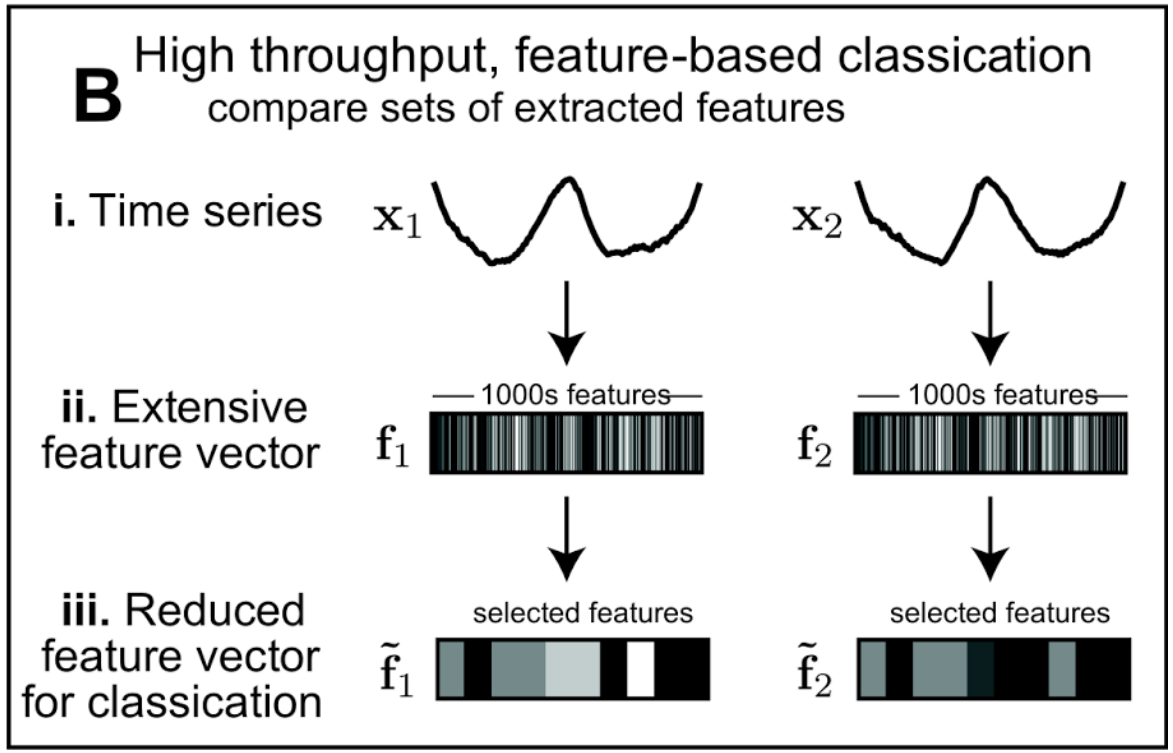
- Create a database of known time series and their classifications.
- Classify a new time series by comparing the distance between the series in the database.



http://systems-signals.blogspot.com.au/

Comparing curves: Feature-matching

- Extract (thousands of) features from the time series.
- Compare series based on these features or on a reduced feature vector.



http://systems-signals.blogspot.com.au/

Classifying curves

Comparative approaches

- cluster analysis
- principal component analysis (PCA)

Model-based approaches

- cubic splines
- harmonic analysis
- state space models

Comparative approaches: Cluster analysis

Rani & Sikka (2012) Recent techniques of clustering of time series data: a survey

- Three groups of methods:
  1. work directly on time series data either in frequency or time domain
  2. work indirectly with features extracted from time series
  3. work with models built from time series

1. Work directly on time series

Table 2.1 Summary of Temporal-Proximity-Based Clustering Approach

Paper	Distance Measure	Algorithm	Application
M. Kumar	Based on the assumed independent Gaussian models of data errors	Agglomerative Hierarchical	Seasonality pattern in retails
T.-W. Liao	Euclidean and symmetric version of Kullback–Liebler distance	K-Means and Fuzzy C-Means	Battle simulations
T.-W. Liao	Dynamic Time Warping	K-Medoids Based Genetic Clustering	Battle simulations
C.S. Möller-Levet	Short time series (STS) distance	Modified Fuzzy C-Means	DNA microarray
Shumway	Kullback–Leibler discrimination information measure	Agglomerative Hierarchical	Earthquakes and mining explosions
Vit Niennattrakul	Dynamic Time Warping	K-Means, K-Medoids	Multimedia time series
Pooya Sobhe Bidari	Pearson Correlation	K-Means, Fuzzy C-Means	Pattern extraction in genes
Hardy Kremer	Dynamic Time Warping	Density Based Subsequence Clustering	Detecting climate change
Jian Yin	Grey Relation	Hierarchical Clustering	Change trend of traffic flow data
S. Chandrakala	Euclidean	Kernal DBScan	Multivariate time series clustering
Aurangzeb Khan	Euclidean	K-Mean+ MFP(Most Frequent Pattern)	Stock and inventory data
Mengfan Zhang	CVT(Computational Verb Theory)	K-Means	Stock market data
S.R.Nanda	Euclidean	K-Means	Portfolio management
Jianfei Wu	N/A	K-Means	Stock data

2. Work with extracted features

Table 2.2 Summary of Representation-Based Clustering Approach

Paper	Features	Distance Measure	Clustering Algorithm	Application
T.-C. Fu	Perceptually important points	Sum of the mean squared distance along the vertical and horizontal scales	Modified SOM	Hong Kong stock market
M. Vlachos	Haar wavelet transform	Euclidean	Modified k-means	Non-specific
Huiting Liu	Empirical mode decomposition	Euclidean	Forward propagation learning algorithm	Non-specific
Chonghui GUO	Independent component analysis	Euclidean	Modified k-means	Real world stock time-series
Jian Xin Wu	Independent component analysis	N/A	support vector regression	Financial time-series
Geert Verdoolaege	Wavelet transform	Kullback- Liebler divergence	k-means	Detection of activated voxels in FMRI data
Liu Suyi	Hough transform	N/A	Mean shift algorithm	Feature recognition of underwater images
Dong Jixue	Wavelet transform	N/A	Grid-based partitioning method	Financial time-series

3. Work with time series models

Table 2.3 Summary of Model-Based Clustering Approach

Paper	Model	Distance measure	Clustering algorithm	Application
Baragona	ARMA	Cross-correlation based	Tabu search, GA and	Non-specific
K. Kalpakis	AR	Euclidean	Partitioning around medoids	Public data
Xiong and Yeung	ARMA mixture	Log-liklihood	EM learning	Public data
L. Wang	Discrete HMM	Log-liklihood	EM learning	Tool condition monitoring
Xin Huang	Fuzzy set and R/S analysis model	N/A	Fuzzy clustering iteration method	Predicting agriculture drought
Shan Gao	ARMA-ARCH	N/A	N/A	To analyze the effects of wind data series

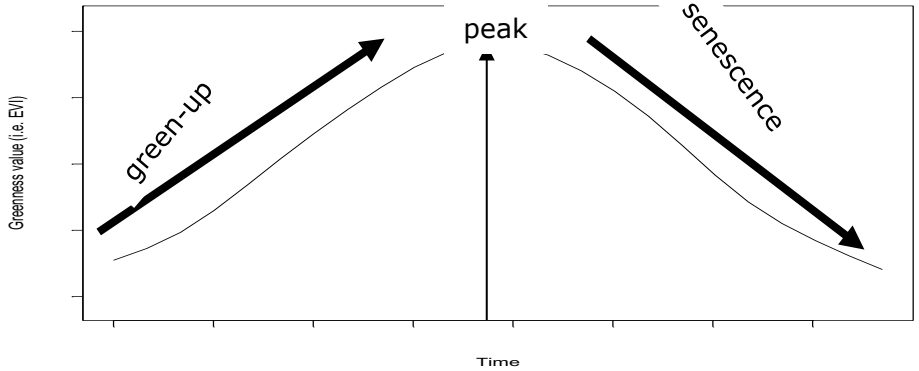
Comparative approaches:
PCA

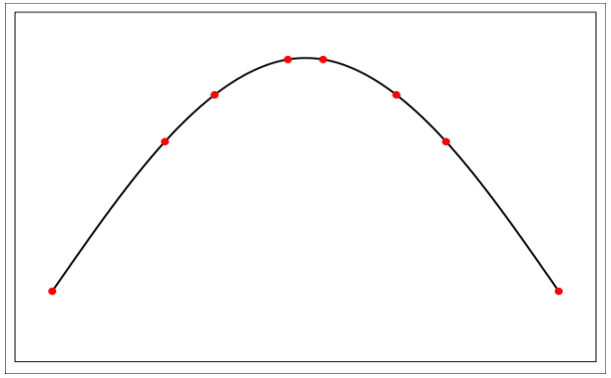
- Considerations:
  1. What assumptions?
  2. What features to include?
  3. What dimension reduction method to choose?
  4. How many components to include?
  5. How to interpret the results?

Plant (2012) Spatial Data Analysis in Ecology and Agriculture using R.

Model-based approaches: Cubic splines

Reconstructing of crop growth profile





Select a set of knots
Interpolate between the points with polynomials

Cubic splines in R

- Extract NDVI phenological parameters on pixel-basis (See Reed et al. (1994) "Measuring phenological variability from satellite imagery“)
  http://r-forge.r-project.org/projects/modis

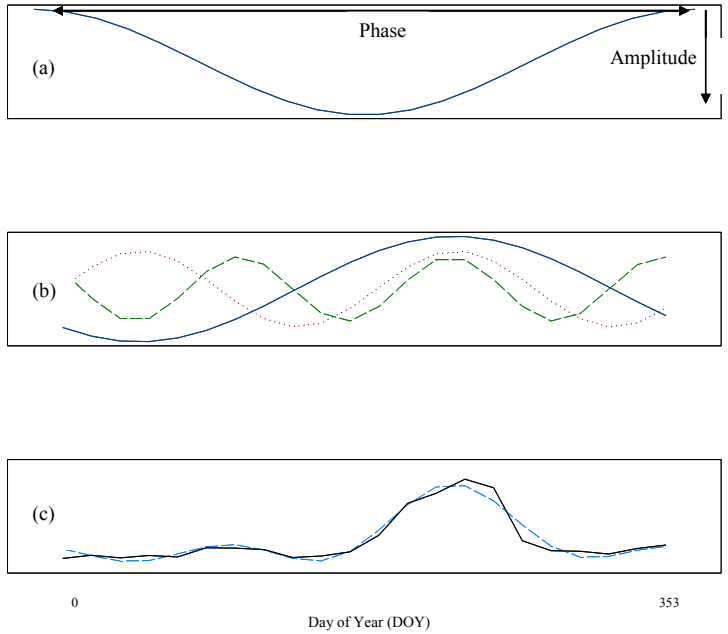
- (In Matlab, see TIMESAT program, Jonsson & Eklundh, 2004)

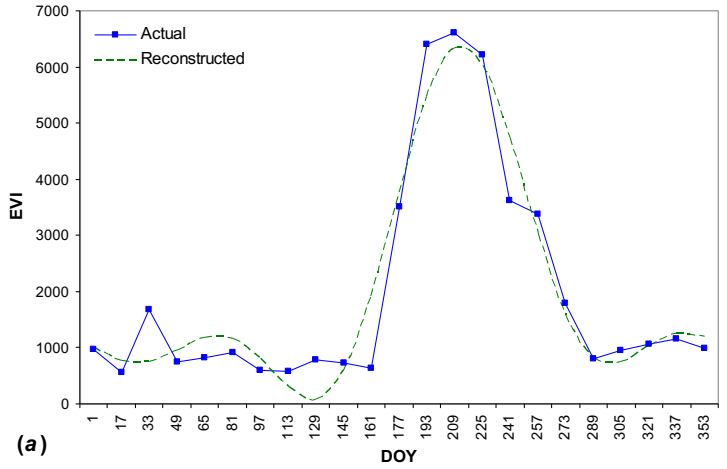
- See also
  https://github.com/mkao006/interpolate_ndvi_satellite_image

- Fit spline:
  http://www.r-bloggers.com/spline-interpolation-of-temporal-resolution-for-satellite-images/

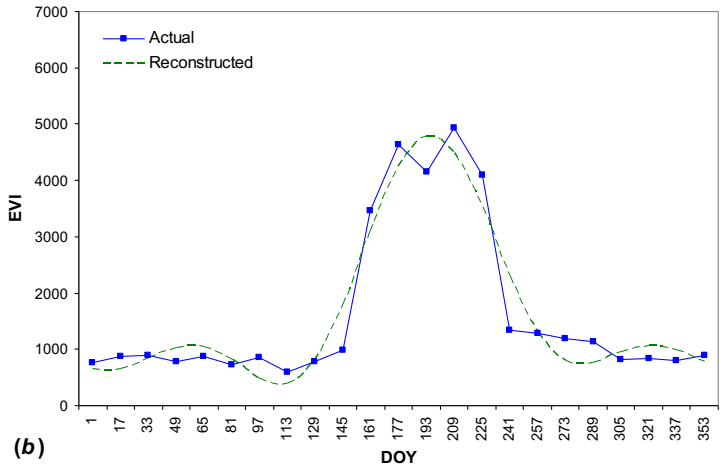
Model-based approaches: harmonic analysis

Harmonic analysis of time series





2005



2006

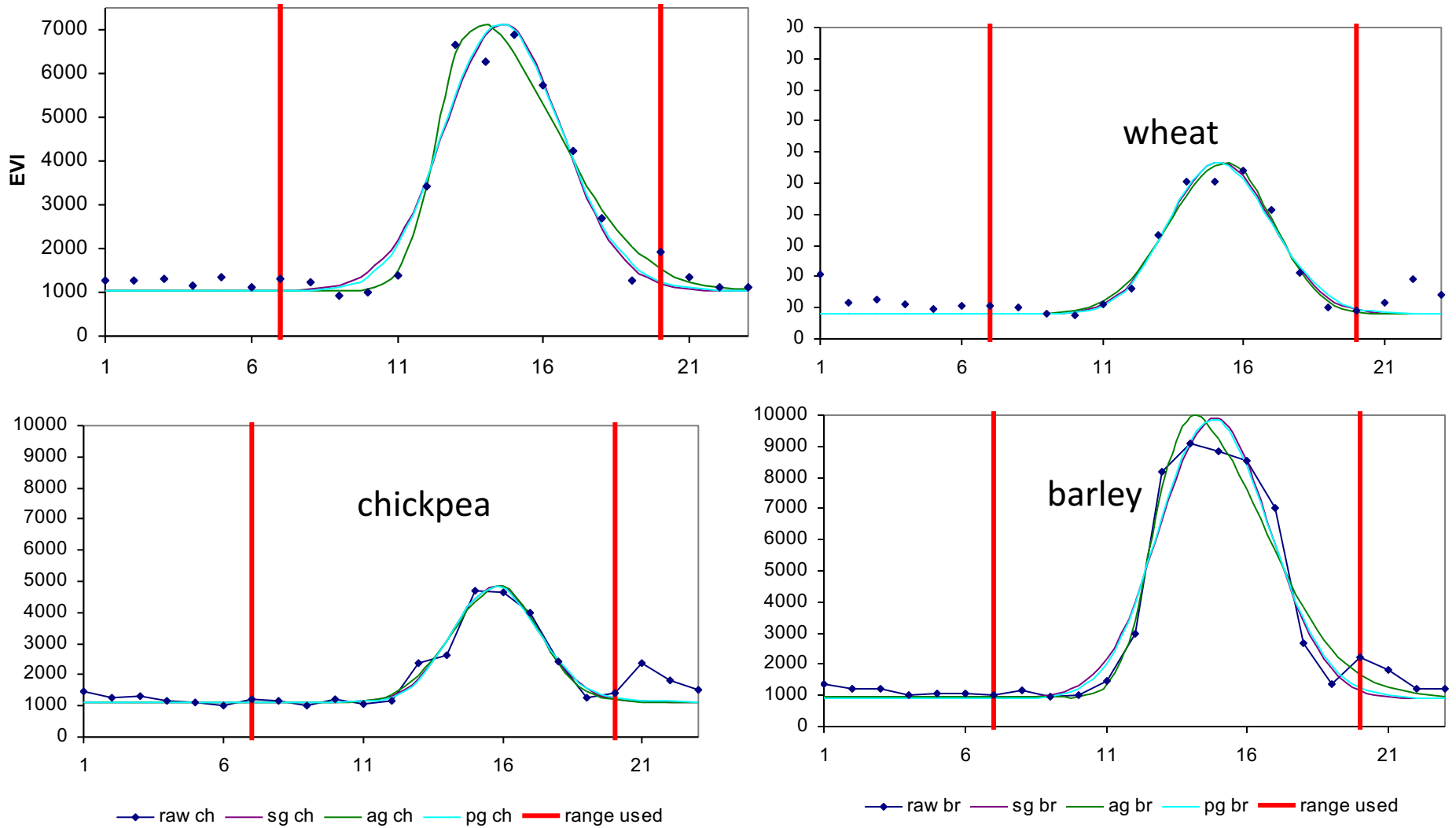
(Verhoef et al 1996; Potgieter et al. 2007, 2010, 2011)

Reconstructing: curve fitting

SG: Symmetric Gaussian; $EVI_p = A*exp((t-B)/C)^2 + D$

AG: Asymmetric Gaussian; $EVI_p = A*exp((t-B)/ C)^2 + D$, where C* takes different values if x< or > B

PG: Pseudo-gaussian; $EVI_p = A*(t/C)B*exp(-B/2 ((t/C)^2 - 1)) + D$



Comparing methods

Potgieter *et al.* (2007)

- Enhanced Vegetation Index (EVI) from 16-day MODIS satellite imagery within the cropping period (i.e. April–November) was investigated to estimate crop area for wheat, barley, chickpea, and total winter cropped area for a case study region in NE Australia.

- Each pixel classification method was trained on ground truth data collected from the study region.

- Three approaches to pixel classification were examined:
  - cluster analysis of trajectories of EVI values from consecutive multi-date imagery during the crop growth period;
  - harmonic analysis of the time series (HANTS) of the EVI values
  - principal component analysis (PCA) of the time series of EVI values.

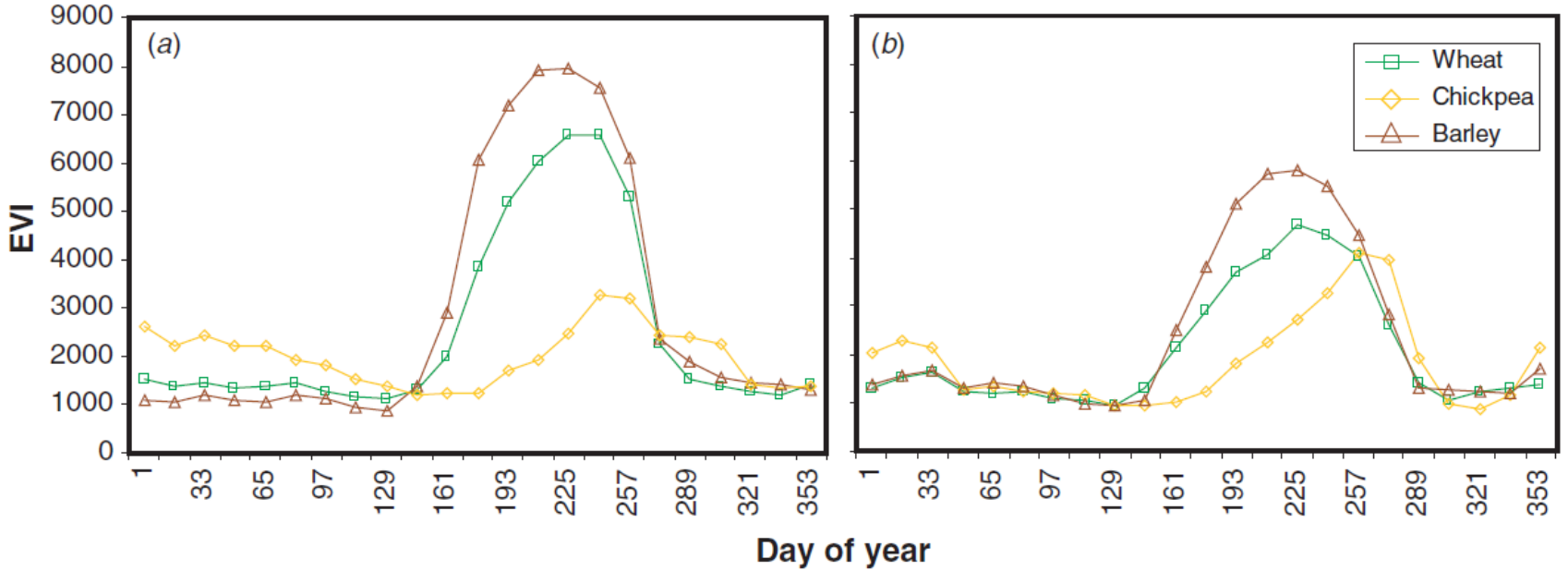


Fig. 2. Average temporal EVI profile throughout the growing season for wheat (square), barley (triangle) and chickpea (diamond) for (*a*) 2003 winter crop season and (*b*) 2004 winter crop season.

All multi-temporal methods showed significant overall capability to estimate total winter crop area. There was high accuracy at pixel scale (>98% correct classification) for identifying overall winter cropping. However, discrimination among crops was less accurate.

Model based approaches: Gaussian profiles

Potgieter *et al.* (2013) Determining crop acreage estimates for specific winter crops using shape attributes from sequential MODIS imagery

- 3 Gaussian curves: symmetric, asymmetric, pseudo-Gaussian
- > 90% classification accuracies in determining crop acreage estimates at pixel scale for each approach.
- Correlation for determining total winter crop areas ($R2 = 0.93$), crop acreage for wheat ($R2 = 0.86$) and barley ($R2 = 0.83$), chickpea acreage ($R2 \leq 0.26$).

Model based approaches: Gaussian profiles

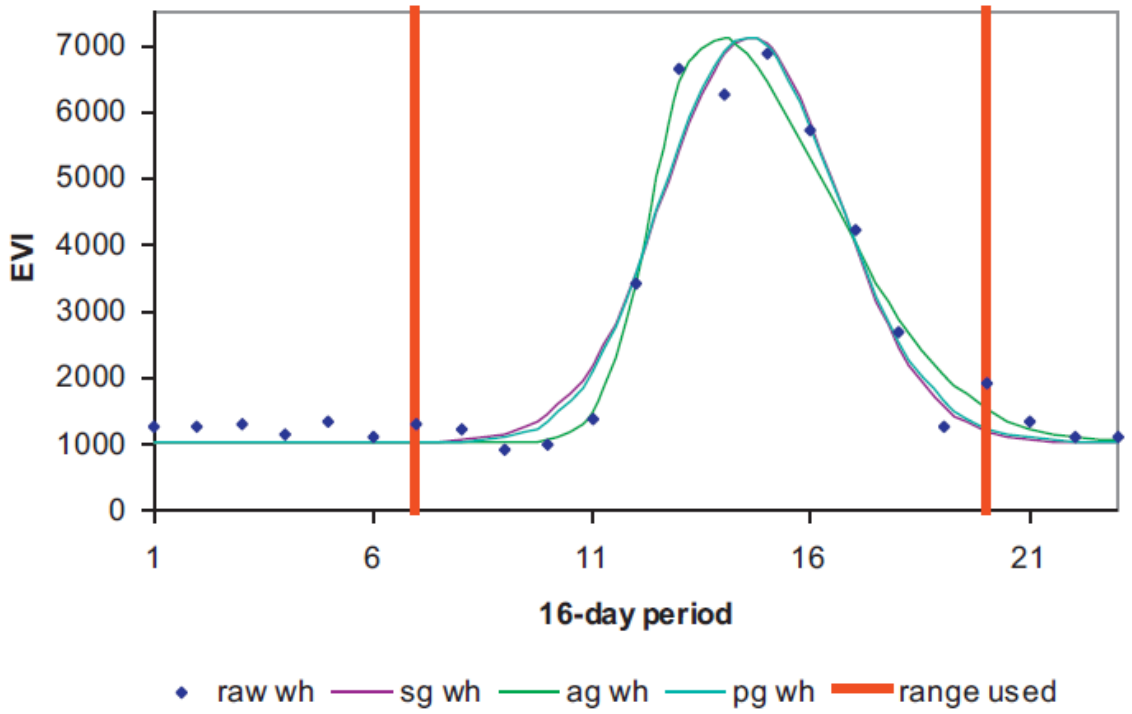


Fig. 3. Depicting the three fitted parametric functions (sg – symmetric, ag – asymmetric and pg – pseudo Gaussian) on Images 7–20 (i.e. April to November inclusive) for wheat using non-linear least squares estimates.

Day 3
Session 4

- Final issues
- Where to from here?
- Concluding Remarks
- Close

Day 3
Session 4

- Final issues
- Where to from here?
- Concluding Remarks
- Close

Day 3
Session 4

- Final issues
- Where to from here?
- Concluding Remarks
- Close