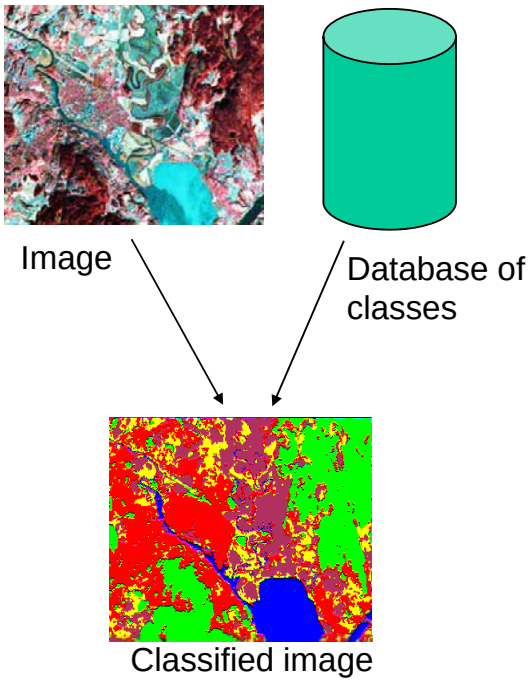
INF 5300 - 2.5.2012
Contextual classification

Anne Schistad Solberg

- Bayesian spatial models for classification
- Markov random field models for spatial context
- Will use the notation from ”Random field models in image analysis” by Dubes and Jain, Journal of Applied Statistics, 1989, pp. 131-154, except section 2.3 and 2.4.
- For the extension to using other types of constraints, more details can be found in ”A Markov random field model for classification of multisource satellite imagery”, by Solbert, Taxt and Jain.

Steps in supervised scene classification

- Feature extraction
- Classifier modelling
- If the features are good/information classes well separated, the choice of classifier is not important
- Typical application-oriented study:
  - Careful selection of features
  - Classifier design:
    - Choose a Gaussian ML classifier
    - Use a MLP neural net or SVM (support vector machines) to avoid making any assumptions of data distribution

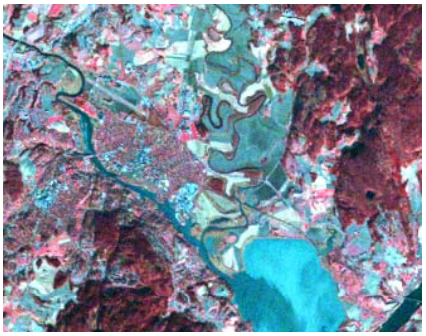


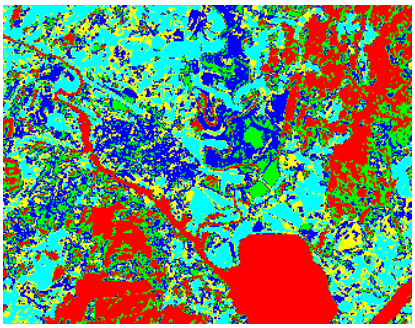
Steps in classification modelling

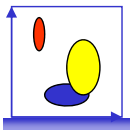
- How to proceed when the data/features are difficult to separate:
  - Choose a classifier with complex decision boundaries
  - Using prior constraints on the scene:
    - Spatial context
    - Multisensor data/data fusion
    - Temporal information

Background – contextual classification

- An image normally contains areas of similar class
  - neighboring pixels tend to be correlated.
- Classified images based on a non-contextual model often contain isolated misclassified pixels (or small regions).
- How can we get rid of this?
  - Majority filtering in a local neighborhood
  - Remove small regions by region area
  - Relaxation (Kittler and Foglein – see INF 3300 Lecture 23.09.03)
  - Bayesian models for the joint distribution of pixel labels in a neighborhood.
- How do we know if the small regions are correct or not?
  - Look at the data, integrate spatial models in the classifier.

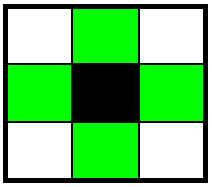




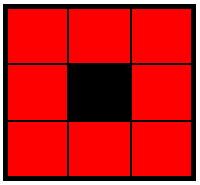


Relation between classes of neighboring pixels

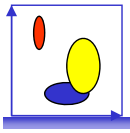
- Consider a single pixel i.
- Consider a local neighborhood N_i centered around pixel i.
- The class label at position i depends on the class labels of neighboring pixels.
- Model the probability of class k at pixel i given the classes of the neighboring pixels.
- More complex neighborhoods can also be used.



4-neighborhood



8-neighborhood



Reminder – pixelwise classification

- Prior probabilities $P(\omega_r)$ for each class
- We have S classes.
- Bayes classification rule: classify a feature vector y_i (for pixel i) to the class with the highest posterior probability $P(\omega_r | y_i)$

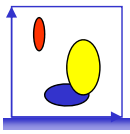
$$P(\omega_r | y_i) = \max_{s=1,\ldots S} P(\omega_s | y_i)$$

- $P(\omega_s | y_i)$ is computed using Bayes formula

$$P(\omega_s | y_i) = \frac{p(y_i | \omega_s) P(\omega_s)}{p(y_i)}$$

$$p(y_i) = \sum_{s=1}^{R} p(y_i | \omega_s) P(\omega_s)$$

- $p(y_i | \omega_s)$ is the class-conditional probability density for a given class (e.g. Gaussian distribution)(corresponds to $p(y_i | x_i=\omega_s)$ here)
- This involves only one pixel i.



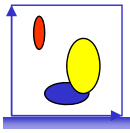
A Bayesian model for ALL pixels in the image

$Y = \{y_1,...,y_N\}$ Image of feature vectors to classify
$X = \{x_1,...x_N\}$ Class labels of pixels

- Classification consists choosing the class that maximizes the posterior probabilities for **ALL** pixels in the image

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{\sum_{\text{all classes}} P(Y \mid X)P(X)}$$

- Maximizing P(X|Y) with respect to $x_1,.....x_N$ is equivalent to maximizing P(Y|X)P(X) since the denominator does not depend on the classes $x_1,.....x_N$.
- Note: we are now maximizing the class labels of ALL the pixels in the image simultaneously.
- This is a problem involving finding N class labels simuntaneously.
- P(X) is the prior model for the scene. It can be simple prior probabilities, or a model for the spatial relation between class labels in the scene.

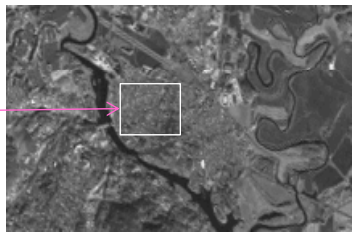


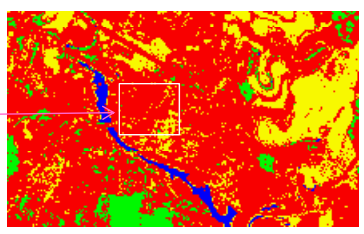
Two kinds of pixel dependency

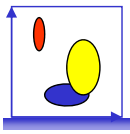
- Interpixel feature dependency:
  - Dependency between the feature vectors.
- Interpixel class dependency:
  - Dependency between class labels of neighboring pixels.

These two types will now be explained more formally.

Model the joint distribution of the gray level of neighboring pixels $p(y_1,y_2|x_1,x_2)$
y_1,and y_2 are the feature vectors
x_1 and x_2 are the class labels
Model the probability for the class labels $p(x_1|x_2)$







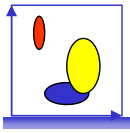
Background: A little statistics

- Consider two events A and B.
- P(A) and P(B) is the probability of events A and B.
- P(B|A) is the conditional probability of B assuming A, and is defined as:

$$P(B \mid A) = \frac{P(B, A)}{P(A)}$$

$$P(B \mid A)P(A) = P(B, A) = P(A \mid B)P(B)$$

- P(A,B) is the joint probability of the two events A and B.

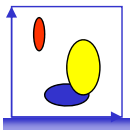


Interpixel feature dependency

- $P(y_1,y_2,....y_N \mid x_1, x_2,...., x_N)$ is generally the joint probability of observing feature vectors $y_1,....y_N$ at pixel positions 1,...N given the underlying true class labels of the pixels.
- The observed feature vector for pixel *i* might depend on the observed feature vector for pixel *j* (neighboring pixels)
- We will not consider such models (If you are interested, see Dubes and Jain 1989).

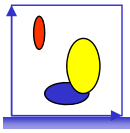
- If the feature vector for pixel i is independent of all the other pixels, this can be simplified as:

$$P(y_1,....y_N \mid X) = \prod_{i=1}^{N} P(y_i \mid x_i) = P(y_1 \mid x_1) \cdot P(y_2 \mid x_2) \cdots P(y_N \mid x_N)$$



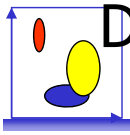
Interpixel class dependency

- The class labels for pixel *i* depends on the class labels of neighboring pixels, but not on the neighbors' observed feature vectors.
  - Such models are normally used for classification.
  - Reasonable if the features are not computed from overlapping windows
  - Reasonable if the sensor does not make correlated measurement errors
- What this means is that when we estimate the class label of pixel *i*, we think that it will be valuable to know the class labels of the neighboring pixels (the image consists of regions with partly continuous class type).



Introduction to Markov random field modelling

- Two elements are central in Markov modelling:
  - Gibbs random fields
  - Markov random fields
- There is an analogy between Gibbs and Markov random fields as we soon will see.
- This will result in an energy function minimization problem.

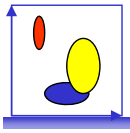


Discrete Gibbs random fields (GRF) - Global model

- A discrete Gibbs random field gives a global model for the pixel labels in an image:

$$P(\mathbf{X} = \mathbf{x}) = \mathrm{e}^{-\mathrm{U}(\mathbf{x})/\mathrm{Z}}$$

- **X** is a random variable, **x** is a realization of **X**.
- U(**x**) is a function called energy function
- Z is a normalizing constant



Neighborhood definitions (for MRFs)

- Pixel site *j* is a neighbor of site *i≠j* if the probability

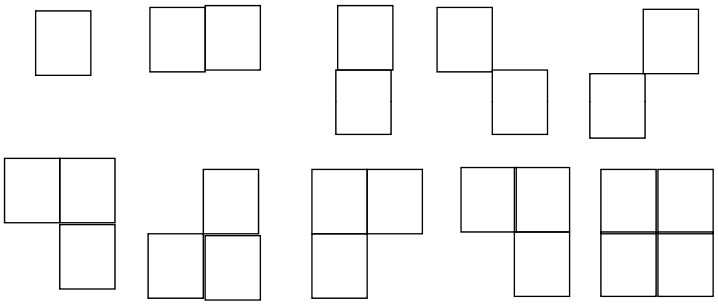
$$P(X_i = x_i \mid \text{all } X_k = x_k, k \neq i)$$

  depends on x_j, the value of X_j.
- A clique is a set of sites in which all pairs of sites are mutual neighbors. The set of all cliques in a neighborhood is denoted Q.
- A potential function or clique function $V_c(\mathbf{x})$ is associated with each clique *c*.
- The energy function U(x) can be expressed as a sum of potential functions $U(\mathbf{x}) = \sum_{c \in Q} V_c(\mathbf{x})$
- 8-neighborhoods are commonly used, but more complex neighborhoods can also be defined.

Neighborhoods and cliques

2	1	2
1	t	1
2	1	2

1st and 2nd order neighbors of pixel t



Clique types for a 2nd order neighborhood

Remark: we normally only use cliques involving two sites. The model is then called a *pairwise interaction model. Then a clique is just a pair of neighboring pixels.*

Common simple potential functions

- Derin and Elliott's model:

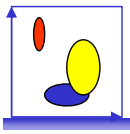
$$V_c(\mathbf{x}) = \begin{cases} \xi_c \text{ if all sites in clique c have the same class} \\ -\xi_c \text{ otherwise} \end{cases}$$

- Ising's model:

$$V_c(\mathbf{x}) = \beta I(x_i, x_k)$$

  - β controls the degree of spatial smoothing
  - $I(c_i,c_k) = -1$ if $c_i = c_k$ and 0 otherwise
  - This corresponds to counting the number of pixels in the neighborhood assigned to the same class as pixel i.

- These two models are equivalent (except a different scale factor) for second order cliques



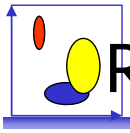
Discrete Markov random fields – local interaction models

- A Markov random field (MRF) is defined in terms of local properties.
- A random field is a discrete Markov random field with respect to a given neighborhood if the following properties are satisfied:
  1. Positivity: $P(\mathbf{X}=\mathbf{x})>0$ for all $\mathbf{x}$
  2. Markov property:

     $P(X_t=x_t|\mathbf{X}_{S|t}=\mathbf{x}_{S|t})=P(X_t=x_t|\mathbf{X}_{\partial t}=\mathbf{x}_{\partial t})$

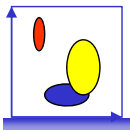
     $S|t$ refers to all M pixel sites, except site t

     ∂t refers to all sites in the neighborhood of site t
  3. Homogeneity: $P(X_t=x_t|\mathbf{X}_{\partial t}=\mathbf{x}_{\partial t})$ is the same for all sites t.



Relationship between MRF and GRF

- A unique GRF exists for every MRF field and vice-versa if the Gibbs field is defines in terms of cliques of a neighborhood system.
- Advantage: a global model can be specified using local interactions only.
- We can use a local energy function involving pixels in a neighborhood.



Back to the initial model…

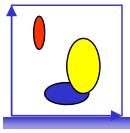
$Y = \{y_1,\ldots,y_N\}$ Image of feature vectors to classify

$X = \{x_1,\ldots x_N\}$ Class labels of pixels

Task: find the optimal estimate **x**′ of the true labels **x*** for all pixels in the image

- Classification consists choosing the class labels **x**′ that maximizes the posterior probabilities

$$P(\mathbf{X}=\mathbf{x}\mid\mathbf{Y}=\mathbf{y})=\frac{P(\mathbf{Y}=\mathbf{y}\mid\mathbf{X}=\mathbf{x})P(\mathbf{X}=\mathbf{x})}{\sum_{\text{all classes}}\mathrm{P}(\mathbf{Y}=\mathbf{y}\mid\mathbf{X}=\mathbf{x})\mathrm{P}(\mathbf{X}=\mathbf{x})}$$



- We assume that the observed feature vectors are conditionally independent:

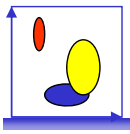
$$P(\mathbf{Y}=\mathbf{y}\mid\mathbf{X}=\mathbf{x})=\prod_{i=1}^{M}P(Y_i=y_i\mid X_i=x_i)$$

- We use a Markov field to model the spatial interaction between the classes (the term P(**X**=**x**)).

$$P(\mathbf{X}=\mathbf{x})=\mathrm{e}^{-\mathrm{U}(\mathbf{x})/\mathrm{Z}}$$

$$U(\mathbf{x})=\sum_{c\in Q}V_c(\mathbf{x})$$

$$V_c(\mathbf{x})=\beta I(x_i,x_k)$$



- Rewrite $P(Y_i=y_i|X_i=x_i)$ as

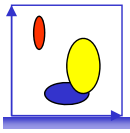
$$P(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x}) = \frac{1}{Z_1} e^{-Udata(Y|X)}$$

$$Udata(Y \mid X) = \sum_{i=1}^{M} -\log P(Y_i = y_i \mid X_i = x_i)$$

- Then, $$P(\mathbf{X} = \mathbf{x} \mid \mathbf{Y} = \mathbf{y}) = \frac{1}{Z_2} e^{-Udata(Y|X)} e^{-U(X)}$$

- Maximizing this is equivalent to minimizing

$$U_{data}(Y \mid X) + U(X)$$

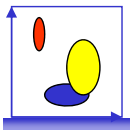


Udata(X|C)

- Any kind of probability-based classifier can be used, for example a Gaussian classifier with a k classes, d-dimensional feature vector, mean μ_k and covariance matrix Σ_k:

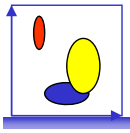
$$Udata(x_i \mid c_i) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log\left(\left|\Sigma_k\right|\right) - \frac{1}{2} x_i^T \Sigma_k^{-1} x_i + \mu_k^T \Sigma_k^{-1} x_i - \frac{1}{2} \mu_k^T \Sigma_k^{-1} \mu_k$$

$$\propto -\frac{1}{2} x_i^T \Sigma_k^{-1} x_i + \mu_k^T \Sigma_k^{-1} x_i - \frac{1}{2} \mu_k^T \Sigma_k^{-1} \mu_k - \frac{1}{2}\log\left(\left|\Sigma_k\right|\right)$$



Finding the labels of ALL pixels in the image

- We still have to find an algorithm that is able to find an estimate **x'** for all pixels.
- Alternative optimization algorithms are:
  - Simulated annealing (SA)
    - Can find a global optimum
    - Is very computationally heavy
  - Iterated Conditional Modes (ICM)
    - A computationally attractive alternative
    - Is only an approximation to the MAP estimate
  - Maximizing the Posterior Marginals (MPM)
- We will only study the ICM algorithm, which converges only to a local minima and is theoretically suboptimal, but computationally feasible.



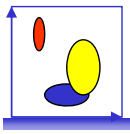
ICM algorithm

1. Initilalize x_t, t=1,...N as the non-contextual classification by finding the class which maximize $P(Y_t=y_t|X_t=x_t)$.
2. For all pixels t in the image, update $\hat{x}_t$ with the class that maximizes

$$P(Y_t = y_t \mid X_t = x_t)P(X_t = x_t \mid \mathbf{X}_{\partial t} = \hat{x}_{\partial t})$$

3. Repeat 2 n times

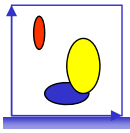
Usually <10 iterations are sufficient



ICM in detail

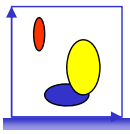
Initilalize x_t, t=1,...N as the non-contextual classification by finding the class which maximize $P(Y_t=y_t|X_t=x_t)$, assign it to classified_image(i,j)

```
For iteration k=1:maxit do
        For i=i:N,j=1:N (all pixels) do
                minimum_energy=High_number;
                For class s=1:S do
                        Udata = -log (P(Yt=yt|Xt=s))
                        Ucontxt=0;
                        nof_similar_neighbors=0;
                        for neighb=1:nof_neighbors
                            if (classified_image(neighb)=s) //neighbor and s of same class
                                ++nof_similar_neighbors;
                        Ucontxt = -beta*nof_similar_neighbors;
                        energy = Udata + Ucontxt;
                        if (energy < minimum_energy)
                                minimum_energy = energy;
                                bestclass = s;
                new_classified_image(i,j) = bestclass;
                if (new_classified_image(i,j)!=classified_image(i,j))
                        ++nof_pixels_changed;
        if nof_pixels_changed<min-limit
                break;
```

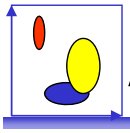
ICM comments

- $P(Y_t=y_t|X_t=x_t)$ can be computed based on various software packages, stored, and used in the ICM algorithm.
- For an image with S classes, this can be stored in a S-band image.
- For each iteration, only the labels x_i change.
  - Why should you use a temporal array to store the updated labels at iteration k, and a separate array for the labels at the next iteration k+1?
    - Hint: try this on a checkerboard image.



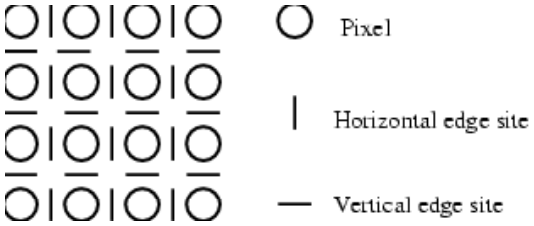
How to choose the smoothing parameter β

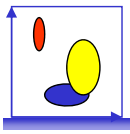
- β controls the degree of spatial smoothing
- β normally lies in the range $1 \leq \beta \leq 2.5$
- The value of β can be estimated based on formal parameter estimation procedures (heavy statistics, but the best way!)
- Another approach is to try different values of β, and choose the one that produces the best classification rate on the training data set.



An energy function for preserving edges

- When β is large, the Ising model tends to smooth the image across edges.
- We can add another energy term to penalize smoothing edges by introducing line processes (Geman and Geman 1984).
- Consider a model where edges can occur between neibhboring pixels and let l(i,j) represent if there is an edge between pixel i and pixel j :





Line processes

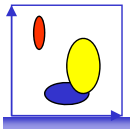
- l(i,j)=0 if there is no edge between pixel i and j, and 1 of there is an edge
- There is an edge if pixels i and j belong to diffent classes, if $c_i \neq c_j$
- We can define an energy function penalizing the number of edges in a neihborhood

$$U_{line}(i) = \beta_l \sum_{k \in N_i} l(i, j)$$

- and let

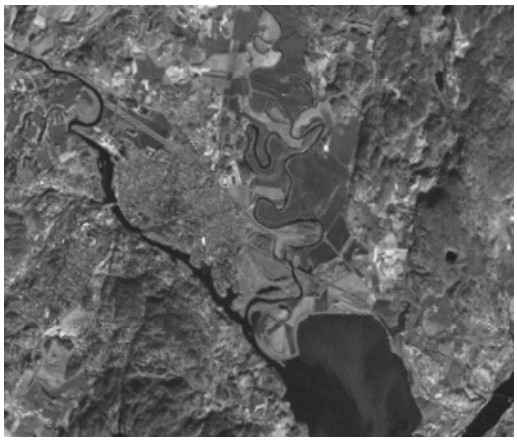
$$U = Udata(X \mid C) + Uspatial(C) + U_{line}(C)$$

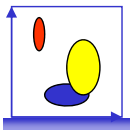
- This will smooth the image, but preserve edges much better.



Test image 1

- A Landsat TM image
- Five classes:
  - Water
  - Urban areas
  - Forest
  - Agricultural fields
  - Vegetation-free areas
- The image is expected to be fairly well approximated by a Gaussian model

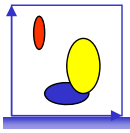




Classification results, Landsat TM image

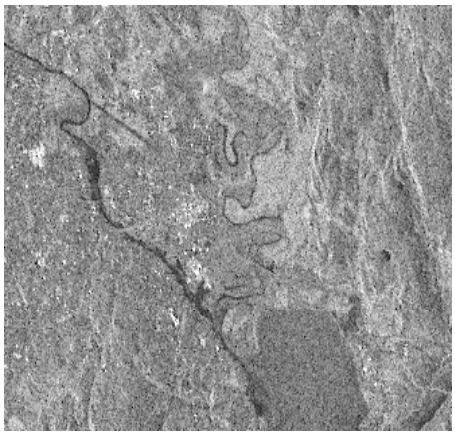
Method	Training data, Noncontextual	Test data, Noncontextual	Test data, contextual
Gaussian	90.1	90.5	96.3
Multilayer perceptron (neural net classifier)	89.7	90.0	95.5

Could we use a SVM classifier?



Data set 2

- ERS SAR image
- 5 texture features from a lognormal texture model used
- 5 classes:
  - Water
  - Urban areas
  - Forest
  - Agricultural fields
  - Vegetation-free areas

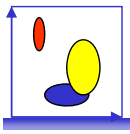


Classification results, SAR image

Method	Training data, Noncontextual	Test data, Noncontextual	Test data, contextual
Gaussian	63.7	63.4	67.1
Multilayer perceptron (neural net)	66.6	66.9	70.8
Tree classifier	70.3	65.0	76.1

More on different energy functions

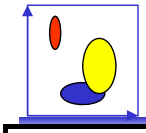
- MRF local energy terms can be used to model other types of context to (see Solberg 1996)
  - Multitemporal classification
  - Consistency with an existing map or previous classification
  - Consistency with other types of GIS data



An energy function for fusion with a thematic map

- Assume that a map or previous classification of the scene exists.
- This map can be partly inaccurate and needs to be updated.
- Let $C^g=\{c^g_1,...,c^g_N\}$ be an old map of the area.
- Consider a set of S different classes. The probability for a change from class s_1 to s_2 can be specified as a table of transitions (next page) $\Pr(x_i|c^g_i)$.
- An additional energy term can be

$$U_G = -\beta_g \sum_{neighborhood} \Pr(x_i \mid c_i^g)$$



Example of allowed transitions

	Urban	Forest	Agricultural	Bare soil	Water
Urban	1.0	0.0	0.0	0.0	0.0
Forest	0.1	0.7	0.1	0.1	0.0
Agricultral	0.1	0.1	0.7	0.1	0.0
Bare soil	0.1	0.1	0.1	0.7	0.0
Water	0.0	0.0	0.0	0.0	1.0

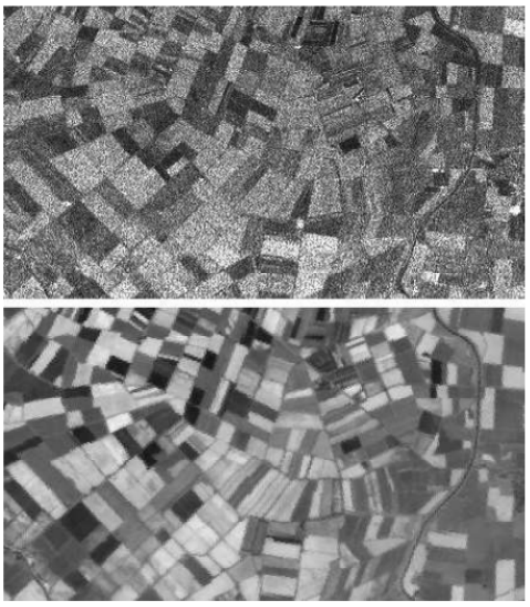
An energy term for crop ownership data

- På norsk/In norwegian: jordskiftekart eller bestandskart av grenser regioner som er en naturlig enhet og som ofte drives likt.
- Let a line process l(i,j) define if pixels i and j are assigned to the same class (l(i,j)=0) or not (l(i,j)=1) in the class label image.
- Let the crop ownership map be represented by a line process.
- An edge site in this map indicates if the two pixels (i,j) it involves are on the same region (l_g(i,j)=0) or not (l(i,j)=1).
- An energy term seeking consistency with the crop ownership map is:
- $$U_{map} = -\beta_{map} \sum_{neighborhood} W(x_i, l(i,j))$$

  where $W(x_i, l(i,j)) = 0$ if $l(i,j) = l_g(i,j)$ and 1 otherwise

Example agricultural classification

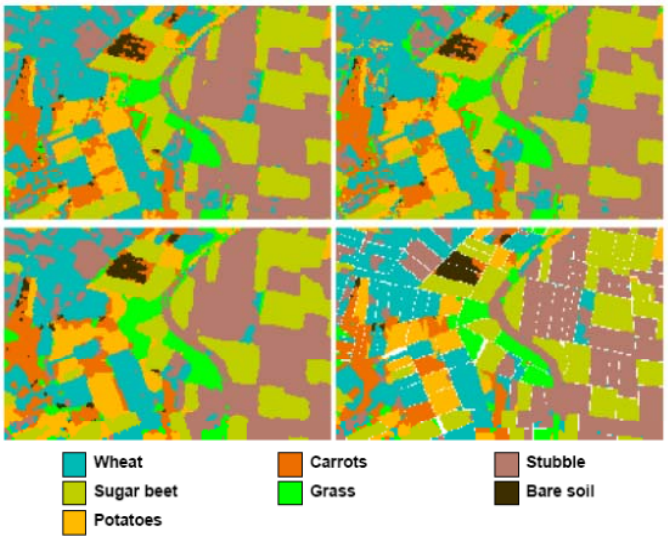
- Optical (Landsat) and SAR image of agricultural site.
- Classes: wheat, sugar beet, potatoes, carrots, grass, stubble, bare soil.
- Field border map also available.



SAR on top, Landsat bottom

Example agricultural classification

Result based on only optical



Result based on optical+SAR No context

Result based on optical+SAR MRF

Result based on optical+SAR MRF with field borders Field borders overlaid in white

Wheat
Sugar beet
Potatoes
Carrots
Grass
Stubble
Bare soil

SAR	Optical	Combined – noncontextual	Combined – MRF	Combined – MRF with field border map
59.9	70.3	71.3	73.0	79.6

Learning goals

- Understand the energy function combining the data term and the contextual term.
- Understand the Ising model
- Understand the ICM algorithm
- Realize that other types of spatial constrains can be added by modifying the energy function.