Workshop report **Due June 7 I will email dropbox link**

1. Daniels report is on website
2. Don't expect to write it based on listening to one project (we had 6 only 2 was sufficient quality)
3. I suggest writing it on one presentation.
4. Include figures (from a related paper or their presentation)
5. Include references

From email to attendees: There will be a very diverse group of people attending the workshop, including over **60 learning-hungry students**.

Update: We are all set to have your students attend. We will not register them, so they can come and go as needed. food is for the registered participants and please allow them to eat first. Currently we have 70 registered participants and plan to order food for ~100.

4:15-4:30: Bruce Cornuelle, Scripps Institution of Oceanography
“A less grand challenge: How can we merge machine learning with data assimilation? ”

Peter: I propose that if data assimilation is posed “correctly” it is already machine leaning. Anyway looking forward to your talk.

Bruce: I agree, but most machine learning I know about doesn't build in prior known dynamics or let you understand what the machine has learned. If you have examples to the contrary, please give me references. I know about the attempts to "invert" the networks, though.
I also want to know the pdfs that the machine learning technique is optimal for, both in the data and the unknowns, in the way that L2 is optimal for gaussians and L1 is optimal for exponentials.

May 24, **Class HW** Bishop Ch 8/13

MAY 30 CODY (kmeans, Ksvd, Kalman)

May 31, No Class. Workshop, Big Data and The Earth Sciences: Grand Challenges Workshop

June 5, Discuss workshop, Discuss final project. **Spiess Hall open for project discussion 11am-7pm.**

June 7, **Workshop report. No class**

June 12 Spiess Hall open for project discussion 9-11:30am and 2-7pm

June 16 Final report delivered. Beer time

For final project discussion **every** afternoon Mark and I will be available. **Please discuss with Mark or me**

Final Report

June 5

In class on July 5 a status report from each group is mandatory. Maximum 2min/person, (i.e. a 5-member group have 10min), shorter is fine. Have presentation on memory stick or email Mark. Class might run longer, so we could start earlier.

For the Final project (Due 16 June 5Pm). Delivery Dropbox request <2GB (details to follow).:

A) Deliver a code:

1) Assume we have reasonable compilers installed (we use Mac OsX)
2) Give instructions if any additional software should be installed.
3) You can ask us to download a dataset. Or include it in this submission
4) Don't include all developed codes. Just key elements.
5) We should not have to reprogram your code.

B) Report

1) The report should include all the following sections: Summary -> Introduction->Physical and Mathematical framework->Results.
2) Summary is a combination of an abstract and conclusion.
3) Plagiarism is not acceptable! When citing use " " for quotes and citations for relevant papers.
4) Don't write anything you don't understand.
5) Everyone in the group should understand everything that is written. If we do not understand a section during grading we should be able to ask any member of the group to clarify. You can delegate the writing, but not the understanding.
6) Use citations. Any concepts which are not fully explained should have a citation with an explanation.
7) Please be concise. Equations are good. Figures essential. Write as though your report is to be published in a scientific journal.
8) I have attached a sample report from Mark, though shorter is preferred.

Discrete Variables (1)

General joint distribution: K^2-1 parameters

$$p(\mathbf{x}_1, \mathbf{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K}\prod_{l=1}^{K} \mu_{kl}^{x_{1k}x_{2l}}$$

Independent joint distribution: 2(K-1) parameters

$$\hat{p}(\mathbf{x}_1, \mathbf{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_{1k}^{x_{1k}} \prod_{l=1}^{K} \mu_{2l}^{x_{2l}}$$

General joint distribution over M variables: K^M -1parameters

M-node Markov chain: K-1 + (M-1) K(K-1) parameters

K-SVD algorithm

K-SVD [Aharon 2006]: Learn optimal dictionary for sparse representation of data

$$\min_{\mathbf{Q}}\{\min_{\mathbf{X}}\|\mathbf{Y}-\mathbf{Q}\mathbf{X}\|_F^2 \ \text{ subject to } \ \forall m, \ \|\mathbf{x}_m\|_0 \le T\}$$

K-SVD algorithm:

1. Solve for coefficients **X=[x_1…x_i]** for fixed **Q** using OMP
2. Solve (1) for dictionary **Q=[q_1…q_i]**, updating both **Q** and **X** from the SVD of representation error

$$\|\mathbf{Y}-\mathbf{Q}\mathbf{X}\|_F = \left\|\left(\underbrace{\mathbf{Y}-\sum_{j\neq k}\mathbf{q}_j\mathbf{x}_T^j}\right)-\mathbf{q}_k\mathbf{x}_T^k\right\|_F$$

$$= \|\mathbf{E}_k-\mathbf{q}_k\mathbf{x}_T^k\|$$

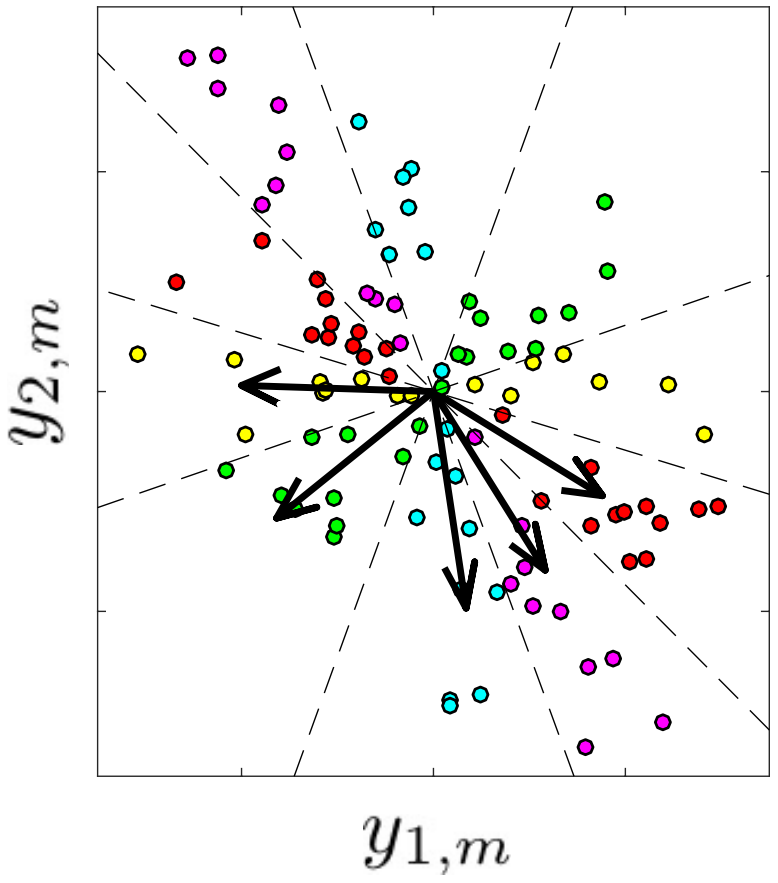
update **q_k, x_k** by SVD

$$\mathbf{E}_k^e = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

$$\mathbf{q}_k = \mathbf{U}(:,1), \mathbf{x}_T^k = \mathbf{V}(:,1)\mathbf{S}(1,1)$$

…. repeat until convergence

2D example

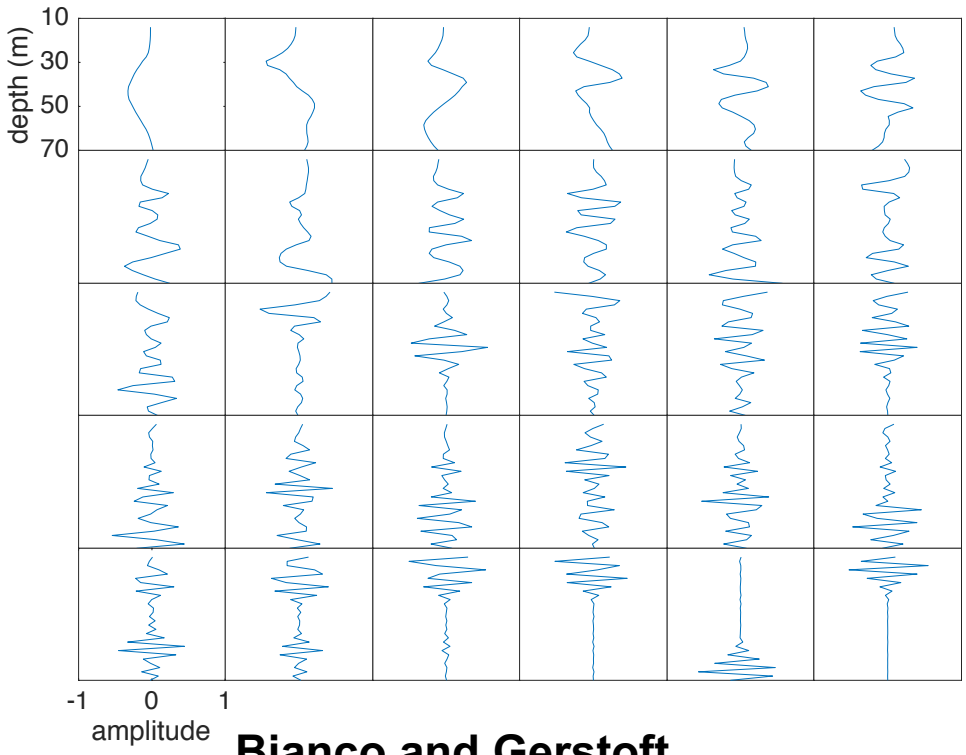


Dictionary learning for SSP

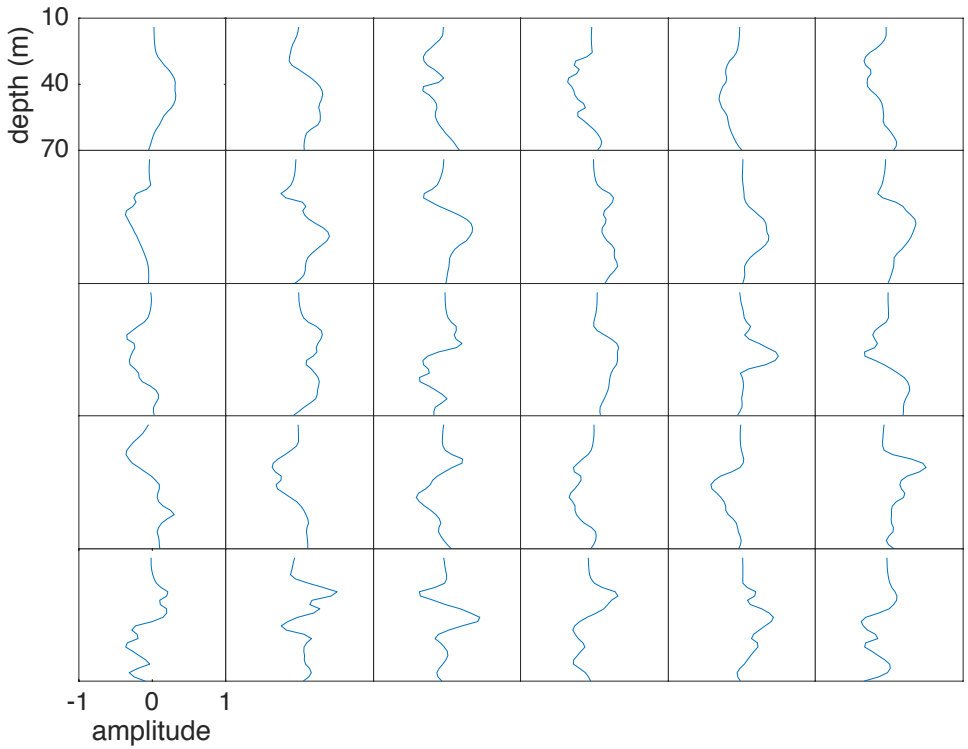
$$D = \arg\max_{\mathbf{D}} \sum_{i=1}^{N} \max_{\mathbf{x}_i} \{P(\mathbf{y}_i, \mathbf{x}_i|\mathbf{D})\}$$

$$= \arg\min_{\mathbf{D}} \sum_{i=1}^{N} \min_{\mathbf{x}_i} \left\{ \|\mathbf{D}\mathbf{x}_i - \mathbf{y}_i\|^2 + \lambda \|\mathbf{x}_i\|_1 \right\}.$$

30 EOF (mean subtracted)



30 K-SVD (mean subtracted)



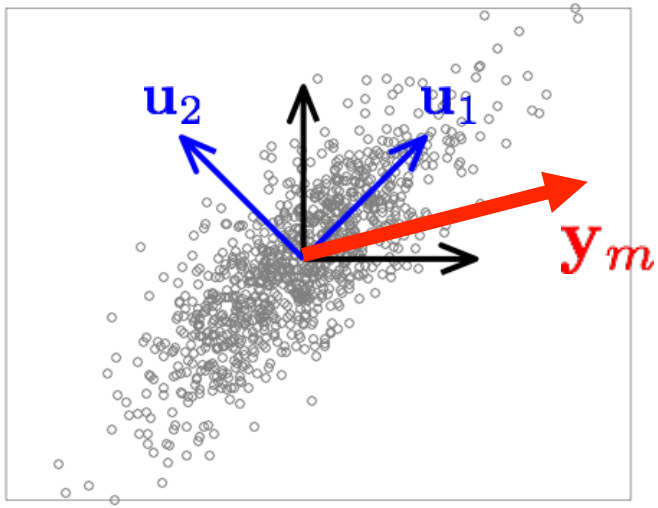
Bianco and Gerstoft …

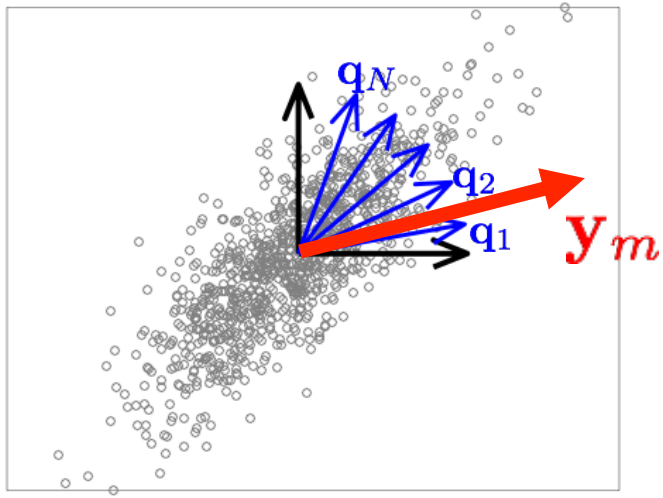
Learned dictionaries and sparsity

EOFs:
Orthogonal, fill $\mathbb{R}^K$

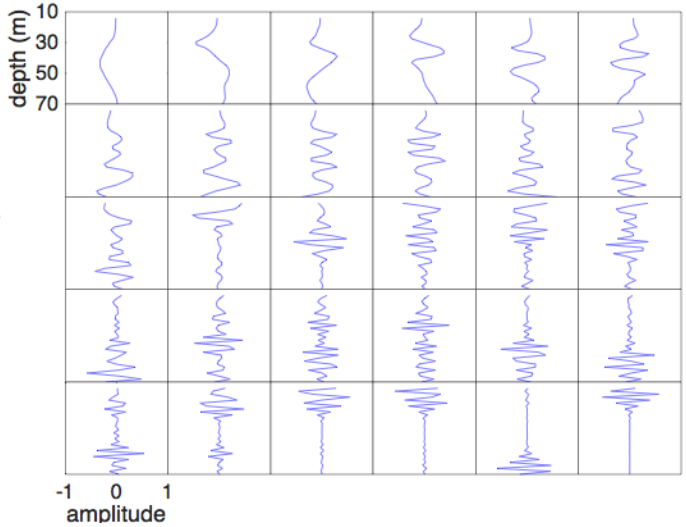
Learned dictionary:
Non-orthogonal, fill feature space

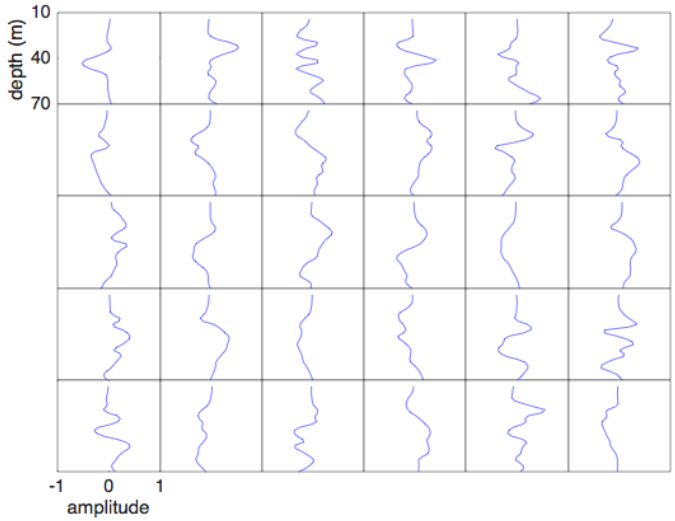
2D Example





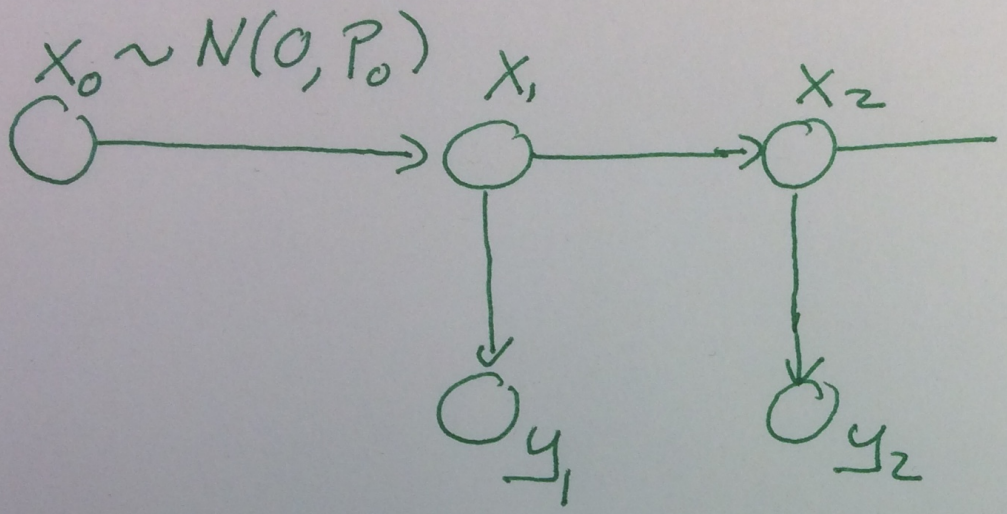
K= 30D shape functions





Learned dictionary: Spanning SSP feature space likelihood that few shapes functions explain a given SSP

State space model



State Eq.

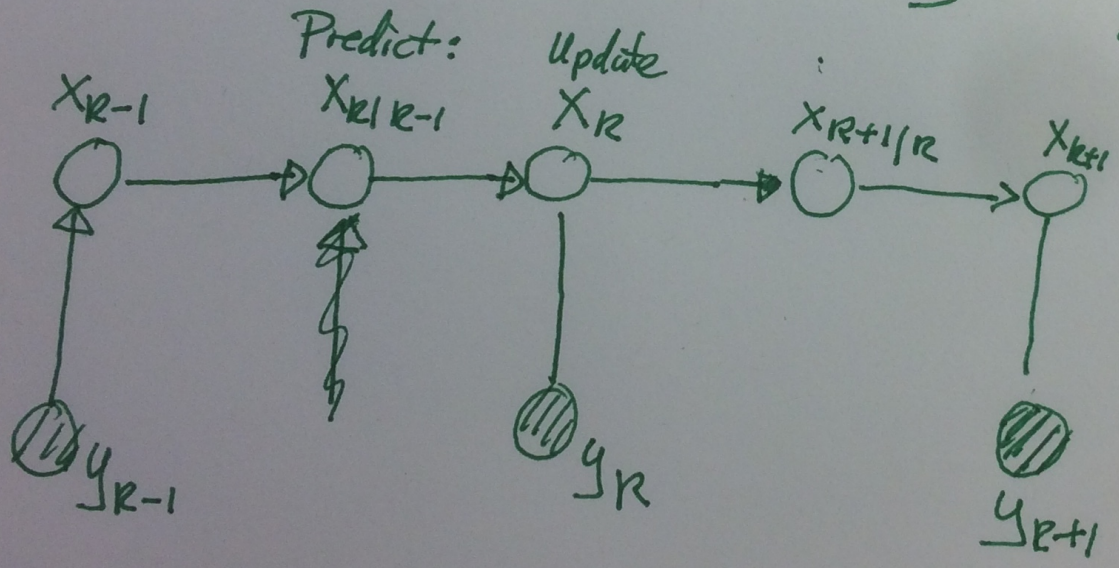
$$x_{k+1} = M_k x_k + \delta_k$$

Measurement Eq

$$y_k = H_k x_k + v_k$$

$$\delta_k \sim N(0, Q_k)$$

$$v_k \sim N(0, R_k)$$



The Model

Consider the discrete, linear system,

$$\mathbf{x}_{k+1} = \mathbf{M}_k \mathbf{x}_k + \mathbf{w}_k, \quad k = 0, 1, 2, \ldots, \tag{1}$$

where

- $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector at time t_k
- $\mathbf{M}_k \in \mathbb{R}^{n \times n}$ is the state transition matrix (mapping from time t_k to t_{k+1}) or model
- $\{\mathbf{w}_k \in \mathbb{R}^n; k = 0, 1, 2, \ldots\}$ is a white, Gaussian sequence, with $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$, often referred to as model error
- $\mathbf{Q}_k \in \mathbb{R}^{n \times n}$ is a symmetric positive definite covariance matrix (known as the model error covariance matrix).

Some of the following slides are from: Sarah Dance, University of Reading

The Observations

We also have discrete, linear observations that satisfy

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \quad k = 1, 2, 3, \ldots, \tag{2}$$

where

- $\mathbf{y}_k \in \mathbb{R}^p$ is the vector of actual measurements or observations at time t_k
- $\mathbf{H}_k \in \mathbb{R}^{n \times p}$ is the observation operator. Note that this is not in general a square matrix.
- $\{\mathbf{v}_k \in \mathbb{R}^p; k = 1, 2, \ldots\}$ is a white, Gaussian sequence, with $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$, often referred to as observation error.
- $\mathbf{R}_k \in \mathbb{R}^{p \times p}$ is a symmetric positive definite covariance matrix (known as the observation error covariance matrix).

We assume that the initial state, $\mathbf{x}_0$ and the noise vectors at each step, $\{\mathbf{w}_k\}$, $\{\mathbf{v}_k\}$, are assumed mutually independent.

The Prediction and Filtering Problems

We suppose that there is some uncertainty in the initial state, i.e.,

$$\mathbf{x}_0 \sim N(0, \mathbf{P}_0) \tag{3}$$

with $\mathbf{P}_0 \in \mathbb{R}^{n \times n}$ a symmetric positive definite covariance matrix.

The problem is now to compute an improved estimate of the stochastic variable $\mathbf{x}_k$, provided $\mathbf{y}_1, \ldots \mathbf{y}_j$ have been measured:

$$\widehat{\mathbf{x}}_{k|j} = \widehat{\mathbf{x}}_{k|y_1,\ldots,y_j}. \tag{4}$$

- When $j = k$ this is called the filtered estimate.
- When $j = k - 1$ this is the one-step predicted, or (here) the predicted estimate.

- The Kalman filter (Kalman, 1960) provides estimates for the linear discrete prediction and filtering problem.
- We will take a minimum variance approach to deriving the filter.
- We assume that all the relevant probability densities are Gaussian so that we can simply consider the mean and covariance.
- Rigorous justifcation and other approaches to deriving the filter are discussed by Jazwinski (1970), Chapter 7.

Prediction step

We first derive the equation for one-step prediction of the mean using the state propagation model (1).

$$\begin{aligned} \widehat{\mathbf{x}}_{k+1|k} &= \mathbb{E}\left[\mathbf{x}_{k+1}|\mathbf{y}_1, \ldots \mathbf{y}_k\right], \\ &= \mathbb{E}\left[\mathbf{M}_k \mathbf{x}_k + \mathbf{w}_k\right], \\ &= \mathbf{M}_k \widehat{\mathbf{x}}_{k|k} \end{aligned} \tag{5}$$

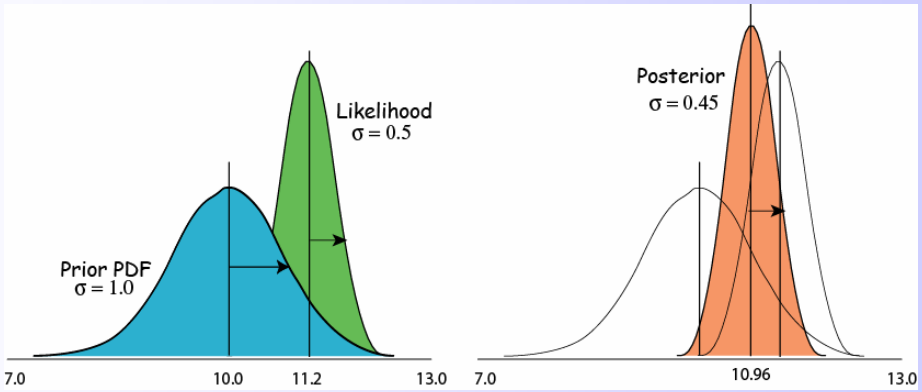
The one step prediction of the covariance is defined by,

$$\mathbf{P}_{k+1|k} = \mathbb{E}\left[(\mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1|k})(\mathbf{x}_{k+1} - \widehat{\mathbf{x}}_{k+1|k})^T | \mathbf{y}_1, \ldots \mathbf{y}_k\right]. \qquad (6)$$

Exercise: Using the state propagation model, (1), and one-step prediction of the mean, (5), show that

$$\mathbf{P}_{k+1|k} = \mathbf{M}_k \mathbf{P}_{k|k} \mathbf{M}_k^T + \mathbf{Q}_k. \qquad (7)$$

Product of Gaussians=Gaussian:



One data point problem

For the general linear inverse problem we would have

Prior: $$p(\boldsymbol{m}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{m}-\boldsymbol{m}_o)^T C_m^{-1}(\boldsymbol{m}-\boldsymbol{m}_o)\right\}$$

Likelihood: $$p(\boldsymbol{d}|\boldsymbol{m}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{d}-G\boldsymbol{m})^T C_d^{-1}(\boldsymbol{d}-G\boldsymbol{m})\right\}$$

Posterior PDF

$$\propto \exp\left\{-\frac{1}{2}[(\boldsymbol{d}-G\boldsymbol{m})^T C_d^{-1}(\boldsymbol{d}-G\boldsymbol{m})+(\boldsymbol{m}-\boldsymbol{m}_o)^T C_m^{-1}(\boldsymbol{m}-\boldsymbol{m}_o)]\right\}$$

$$\propto \exp\left\{-\frac{1}{2}\left[\mathbf{m}-\hat{\mathbf{m}}\right]^T \mathbf{S}^{-1}\left[\mathbf{m}-\hat{\mathbf{m}}\right]\right\}$$

$$\mathbf{S}^{-1} = \mathbf{G}^T\mathbf{C}_d^{-1}\mathbf{G}+\mathbf{C}_m^{-1}$$

$$\hat{\mathbf{m}} = \left(\mathbf{G}^T\mathbf{C}_d^{-1}\mathbf{G}+\mathbf{C}_m^{-1}\right)^{-1}\left(\mathbf{G}^T\mathbf{C}_d^{-1}\mathbf{d}+\mathbf{C}_m^{-1}\mathbf{m}_0\right)$$

$$= \mathbf{m}_0+\left(\mathbf{G}^T\mathbf{C}_d^{-1}\mathbf{G}+\mathbf{C}_m^{-1}\right)^{-1}\mathbf{G}^T\mathbf{C}_d^{-1}\left(\mathbf{d}-\mathbf{G}\mathbf{m}_0\right)$$

Filtering Step

At the time of an observation, we assume that the update to the mean may be written as a linear combination of the observation and the previous estimate:

$$\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}), \tag{8}$$

where $\mathbf{K}_k \in \mathbb{R}^{n\times p}$ is known as the Kalman gain and will be derived shortly.

But first we consider the covariance associated with this estimate:

$$\mathbf{P}_{k|k} = \mathbb{E}\left[(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k})^T | \mathbf{y}_1, \ldots \mathbf{y}_k\right]. \tag{9}$$

Using the observation update for the mean (8) we have,

$$\begin{aligned}
\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k} &= \mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1} - \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}) \\
&= \mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1} - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}), \\
&\quad \text{replacing the observations with their model equivalent,} \\
&= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k\mathbf{v}_k. \qquad (10)
\end{aligned}$$

Thus, since the error in the prior estimate, $\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1}$ is uncorrelated with the measurement noise we find

$$\begin{aligned}
\mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbb{E}\left[(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \widehat{\mathbf{x}}_{k|k-1})^T\right](\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T \\
&\quad + \mathbf{K}_k\mathbb{E}\left[\mathbf{v}_k\mathbf{v}_k^T\right]\mathbf{K}_k^T. \qquad (11)
\end{aligned}$$

Simplification of the a posteriori error covariance formula

Using this value of the Kalman gain we are in a position to simplify the Joseph form as

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1}. \tag{15}$$

Exercise: Show this.

Note that the covariance update equation is independent of the actual measurements: so $\mathbf{P}^{k|k}$ could be computed in advance.

Summary of the Kalman filter

Prediction step

Mean update: $\widehat{\mathbf{x}}_{k+1|k} = \mathbf{M}_k \widehat{\mathbf{x}}_{k|k}$

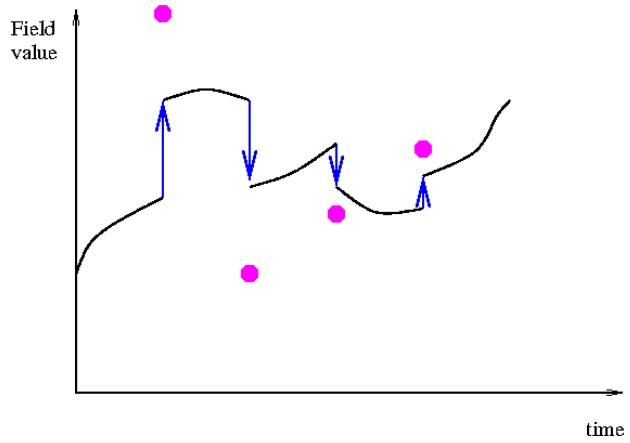
Covariance update: $\mathbf{P}_{k+1|k} = \mathbf{M}_k \mathbf{P}_{k|k} \mathbf{M}_k^T + \mathbf{Q}_k.$

Observation update step

Mean update: $\widehat{\mathbf{x}}_{k|k} = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \widehat{\mathbf{x}}_{k|k-1})$

Kalman gain: $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}^T + \mathbf{R}_k)^{-1}$

Covariance update: $\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}.$



Kalman smoother

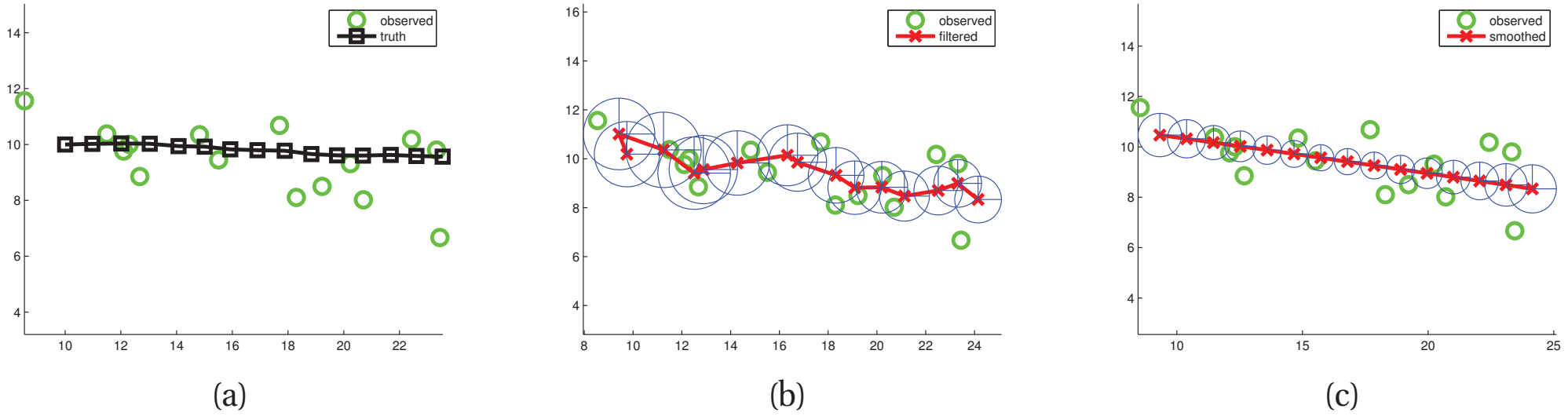
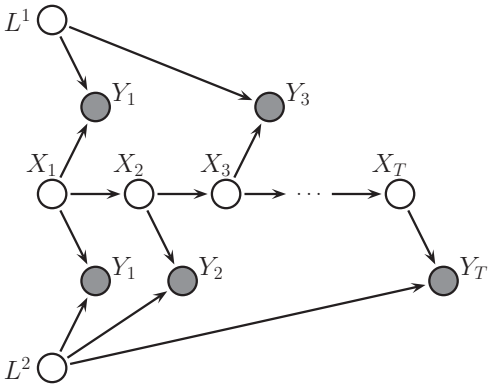


Figure 18.1 Kalman filtering and smoothing. (a) Observations (green cirles) are generated by an object moving to the right (true location denoted by black squares). (b) Filtered estimated is shown by dotted red line. Red cross is the posterior mean, blue circles are 95% confidence ellipses derived from the posterior covariance. For clarity, we only plot the ellipses every other time step. (c) Same as (b), but using offline Kalman smoothing. Figure generated by kalmanTrackingDemo.



Graphical model underlying SLAM. L^i is the fixed location of landmark i, x_t is the robot location, and y_t is the observation. In this trace, the robot sees landmarks 1 and 2 at time 1, then just landmark 2, then just landmark 1, etc.

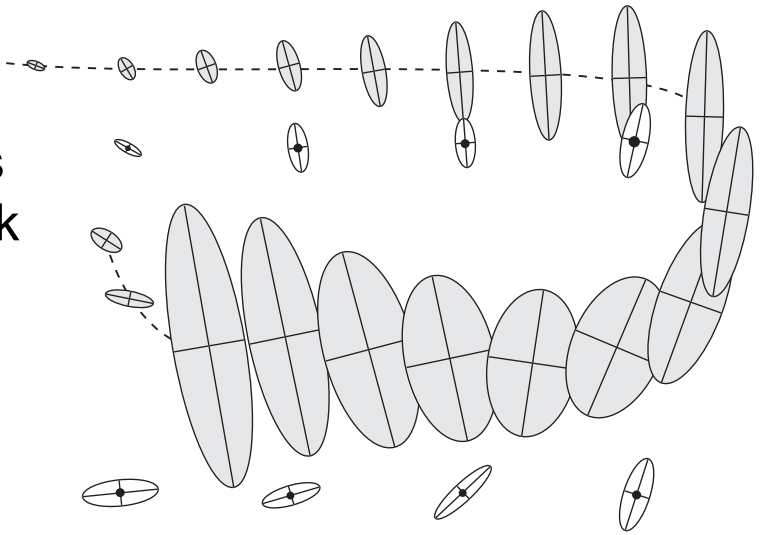


Illustration of the SLAM problem. (a) A robot starts at the top left and moves clockwise in a circle back to where it started. We see how the posterior uncertainty about the robot's location increases and then decreases as it returns to a familar location, closing the loop. If we performed smoothing, this new information would propagate backwards in time to disambiguate the entire trajectory.

Predict N steps ahead

SLAM (Simultaneous Location and Mapping)

Kalman smoother

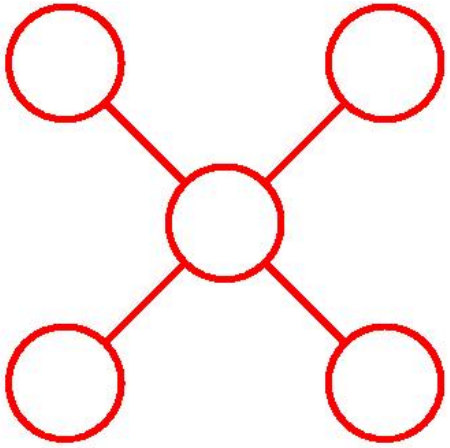
RLS (Recursive least squares)

Advanced KF:

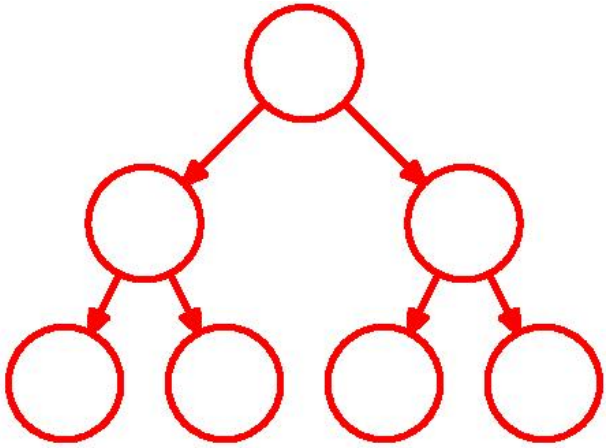
- Ensample KF (EnKF) non Gaussian
- Extended KF (EKF) non-linear
- Unscented KF (UKF) well chosen control points
- … Particle Filter Nonlinear, non Gaussian

Trees

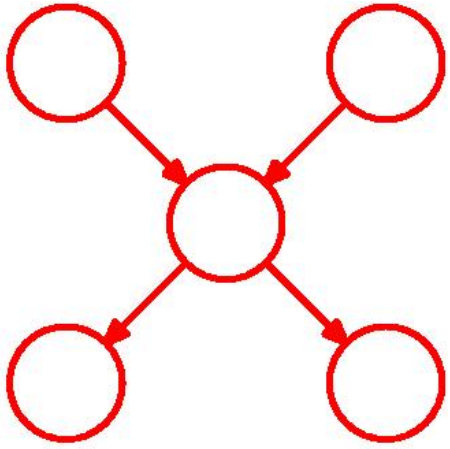
Undirected Tree



Directed Tree



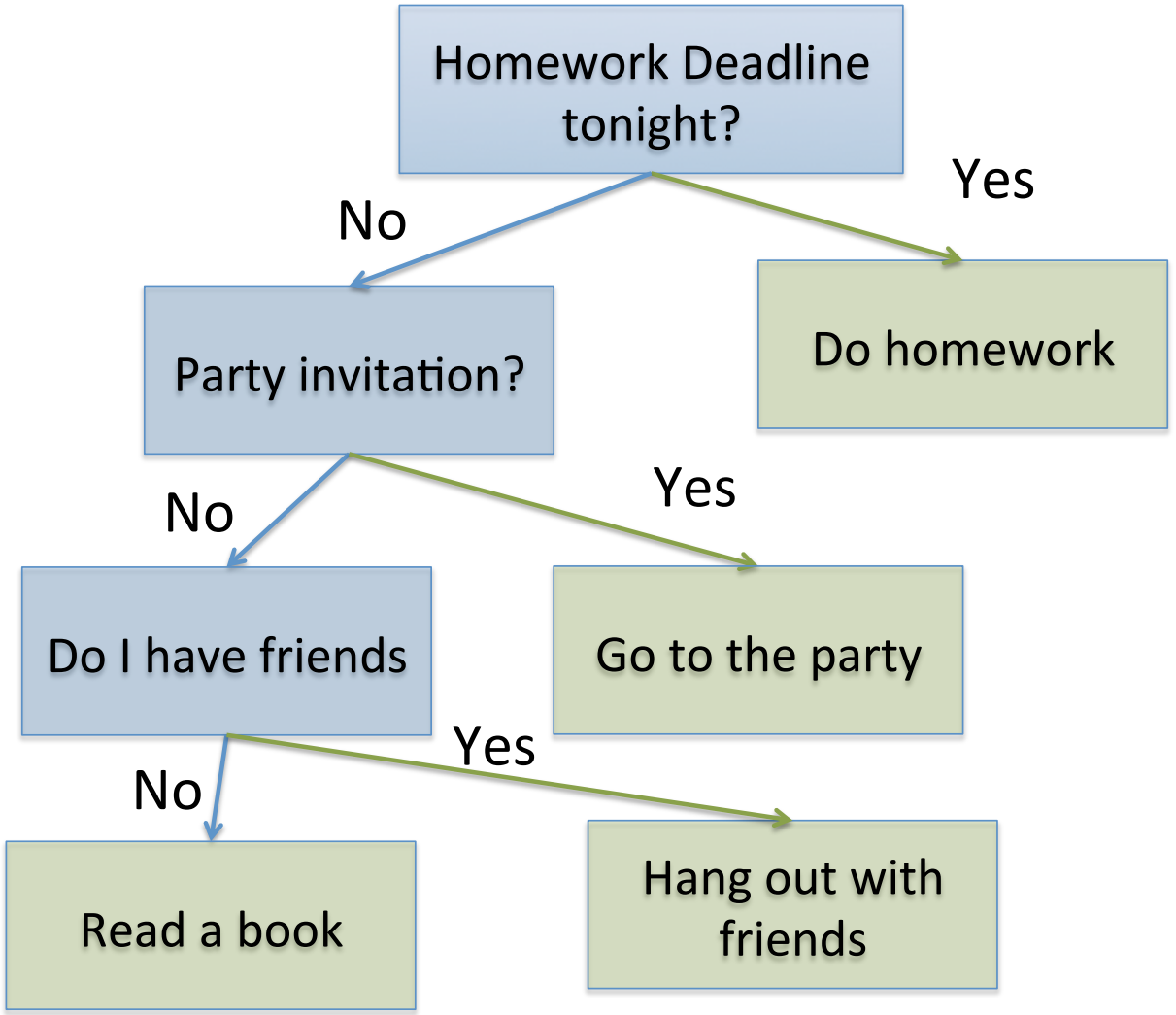
Polytree



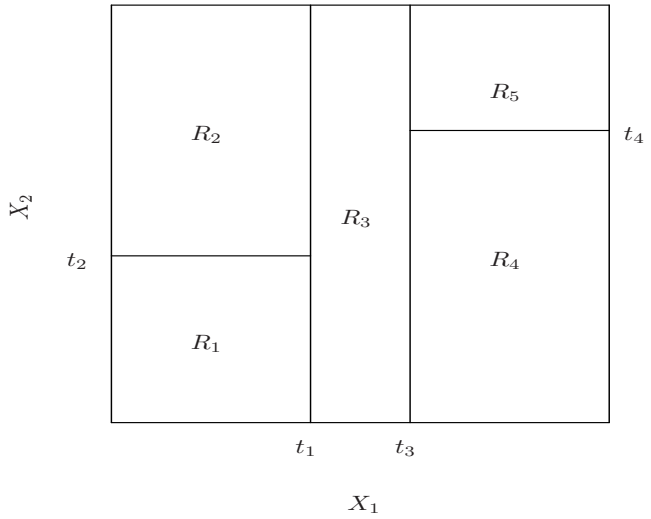
Trees

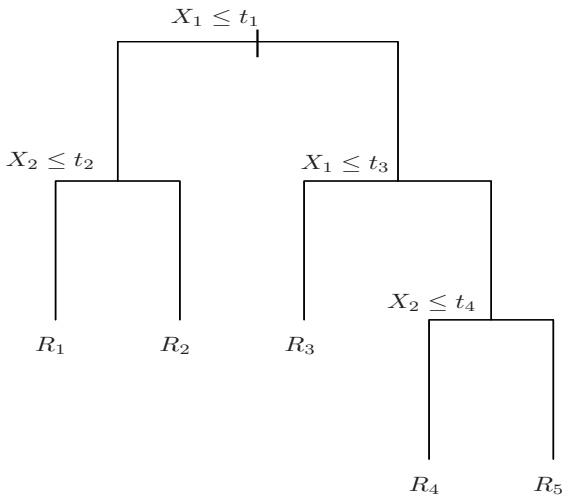
What would you do tonight? Decide amongst the following:

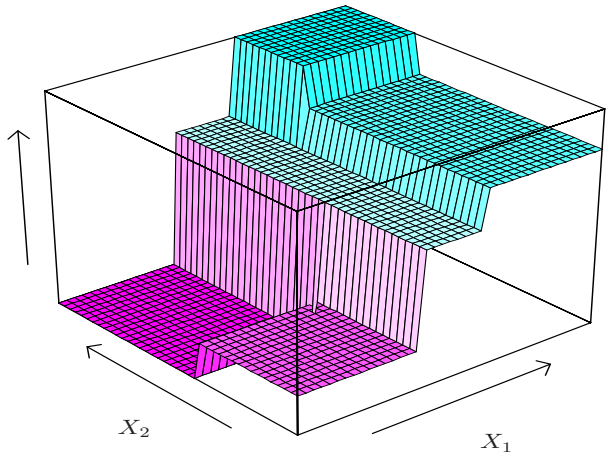
- Finish homework • Go to a party • Read a book • Hang out with friends



Regression Trees (Fig 9.2 in Hastie)







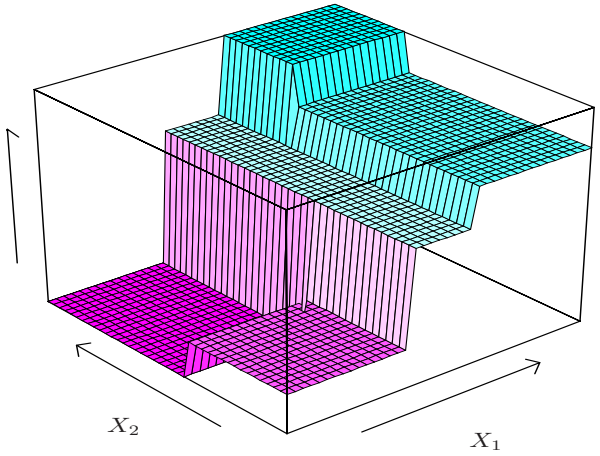
Details of the tree-building process

1. Divide the predictor space, the set of possible values for X1,X2,...,Xp, into J distinct and non-overlapping regions, R1, R2, . . . , RJ .
2. For every observation that falls into the region Rj, we make the same prediction, which is simply the mean of the response values for the training observations in Rj.

The goal is to find boxes R1,...,RJ that minimize the RSS (residual sum square), given by

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

where $\hat{y}_{Rj}$ is the mean response for the training observations within the jth box.



Bagging

Bootstrap aggregation, or **bagging**, is a general-purpose procedure for reducing the variance of a statistical learning method; it is particularly useful and frequently used in the context of decision trees.

we generate B bootstrapped training data sets. We train our method on the bth bootstrapped training set in order to get $f^{*b}(x)$, the prediction at a point x. We then average all the predictions to obtain

$$\hat{f}_{\mathrm{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

Random Forrest

Random forests provide an improvement over **bagged trees** by way of a small tweak that **decorrelates** the trees. This reduces the variance when averaging the trees.

As in bagging, we build a number of decision trees on bootstrapped training samples.

But when building these decision trees, each split in a tree is based on a random selection of m predictors. is chosen split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors

But when building these decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors.

Carrying On…

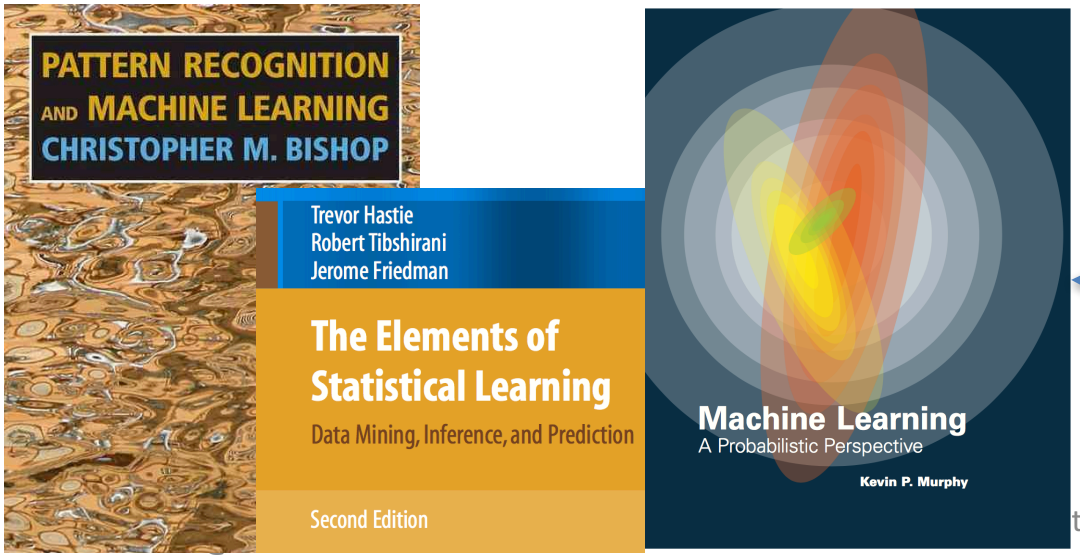
The book by Murphy has more details on ML.

Many interesting courses online and at UCSD.

Lots of opportunities also outside CS.

For next course, more class interaction (phone questions), more cody home work, **physics** better integrated.

Graphical models better integrated, Gaussian processes, sequential state models.



Murphy: "This books adopts the view that the best way to make machines that can learn from data is to use the *tools of probability theory*, which has been the mainstay of statistics and engineering for centuries. "