02457 Non-linear signal processing

2017 - Lecture 11

Technical University of Denmark
DTU Compute, Kgs Lyngby, Denmark

Outline lecture 11

Kernel methods
- Dual representation, kernel matrices

$$(\mathbf{x}_m \approx \mathbf{x}_n) \Rightarrow (\mathbf{t}_m \approx \mathbf{t}_n)$$

Gaussian process prior
- Smoothness ~ output correlation between input neighbors

Support Vector Machines
- Sparse classifier

Sequence data – Ex Audio signals, Music and Speech, Speech production, Symbolic representations

Markov models
- Definition, properties, stationary distribution
- Sequencial Likelihood function and ML, MAP estimation

Dynamic estimators for non-stationary data
- Stochastic gradient
- Evaluation with non-stationarity

Kernel representation of linear model d > N

Assume we have a high-dimensional data set with input variables **x** in d-dimensional space and d > N

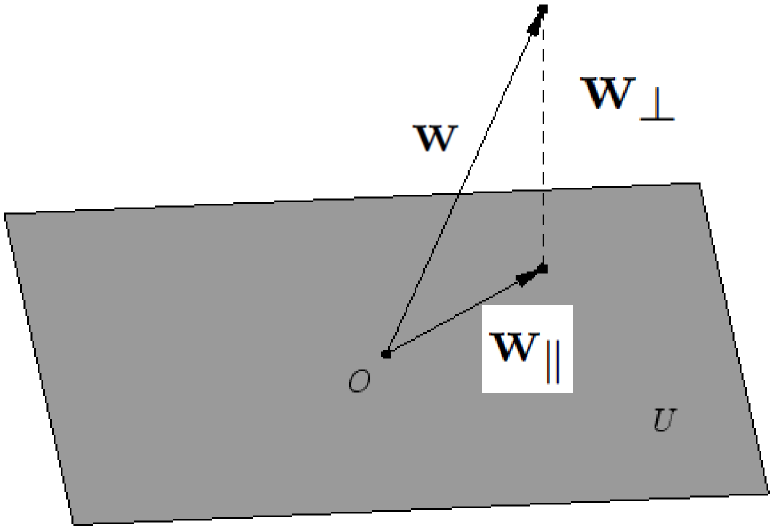
$$D = \{(t_1, \mathbf{x}_1), (t_2, \mathbf{x}_2), ..., (t_N, \mathbf{x}_N)\}$$

Linear model fitted from least squares

$$E(\mathbf{w}) = \sum_{n=1}^{N} (t_n - \sum_{j=1}^{d+1} w_j x_{j,n})^2$$

$$\mathbf{w} = \mathbf{w}_\perp + \mathbf{w}_{||}$$

$$\mathbf{w}_{||} = \sum_{n=1}^{N} a_n \mathbf{x}_n, \qquad \mathbf{w}_\perp^\top \mathbf{x}_n = 0$$



Kernel representation in the linear model

High-dimensional data d > N $D = \{(t_1, \mathbf{x}_1), (t_2, \mathbf{x}_2), ..., (t_N, \mathbf{x}_N)\}$

$$\mathbf{w} = \mathbf{w}_\perp + \mathbf{w}_\parallel \qquad \mathbf{w}_\parallel = \sum_{n=1}^{N} a_n \mathbf{x}_n, \quad \mathbf{w}_\perp^\top \mathbf{x}_n = 0$$

Linear model fitted from least squares

$$\begin{aligned} E(\mathbf{w}) &= \sum_{n=1}^{N} (t_n - \mathbf{w}^\top \mathbf{x}_n)^2 = \sum_{n=1}^{N} (t_n - \mathbf{w}_\parallel^\top \mathbf{x}_n)^2 = \sum_{n=1}^{N} (t_n - \sum_{m=1}^{N} a_m \mathbf{x}_m^\top \mathbf{x}_n)^2 \\ &= \sum_{n=1}^{N} (t_n - \sum_{m=1}^{N} a_m K_{m,n})^2 = \sum_{n=1}^{N} (t_n - (\mathbf{a}^\top \mathbf{K})_n)^2. \end{aligned}$$

$$(\mathbf{K})_{m,n} = K_{m,n} = \mathbf{x}_m^\top \mathbf{x}_n$$

Kernel methods

$$(\mathbf{K})_{m,n} = K_{m,n} = \mathbf{x}_m^\top \mathbf{x}_n$$

Note, we can ignore the component of the weight vector which is orthogonal to the subspace spanned by the data

Costfunction (likelihood function) is ”blind” to this subspace

We can reduce the fitting problem to estimation of an N-dimensional vector (**a**) forget about high-dim data vectors **x** and just keep the kernel matrix **K**

Kernel methods: Implict features

Assume our model is based on a feature representation –

$$\mathbf{x} \mapsto \phi(\mathbf{x})$$

with the kernel trick there is no limitations on the dimensionality

$$E(\mathbf{w}) = \sum_{n=1}^{N}(t_n - \mathbf{w}^\top \phi(\mathbf{x}_n))^2 = \sum_{n=1}^{N}(t_n - (\mathbf{a}^\top \mathbf{K})_n)^2$$

$$(\mathbf{K})_{m,n} = K_{m,n} = \phi(\mathbf{x}_m)^\top \phi(\mathbf{x}_n)$$

Kernel methods: Implict features

Now if we postulate a kernel function K(**x**,**x**')
then we implicitly define a such feature representation

$$(\mathbf{K})_{m,n} = K_{m,n} = \phi(\mathbf{x}_m)^\top \phi(\mathbf{x}_n)$$

When can a symmetric matrix generated from a function k($\mathbf{x}_m$,$\mathbf{x}_n$) be reconstructed as the inner products?

If any subset of points give rise to a positive definite matrix.

Techniques for Constructing New Kernels.

Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, the following new kernels will also be valid:

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= c k_1(\mathbf{x}, \mathbf{x}') && (6.13) \\
k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') && (6.14) \\
k(\mathbf{x}, \mathbf{x}') &= q\left(k_1(\mathbf{x}, \mathbf{x}')\right) && (6.15) \\
k(\mathbf{x}, \mathbf{x}') &= \exp\left(k_1(\mathbf{x}, \mathbf{x}')\right) && (6.16) \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') && (6.17) \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}') && (6.18) \\
k(\mathbf{x}, \mathbf{x}') &= k_3\left(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}')\right) && (6.19) \\
k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x}' && (6.20) \\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) && (6.21) \\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}'_a) k_b(\mathbf{x}_b, \mathbf{x}'_b) && (6.22)
\end{aligned}
$$

where $c > 0$ is a constant, $f(\cdot)$ is any function, $q(\cdot)$ is a polynomial with nonnegative coefficients, $\boldsymbol{\phi}(\mathbf{x})$ is a function from $\mathbf{x}$ to $\mathbb{R}^M$, $k_3(\cdot,\cdot)$ is a valid kernel in $\mathbb{R}^M$, $\mathbf{A}$ is a symmetric positive semidefinite matrix, $\mathbf{x}_a$ and $\mathbf{x}_b$ are variables (not necessarily disjoint) with $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$, and k_a and k_b are valid kernel functions over their respective spaces.

Gaussian kernel, aka squared exponentil aka radial basis function kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2\right)$$

Is it an ok kernel?

$$\|\mathbf{x} - \mathbf{x}'\|^2 = \mathbf{x}^{\mathrm{T}}\mathbf{x} + (\mathbf{x}')^{\mathrm{T}}\mathbf{x}' - 2\mathbf{x}^{\mathrm{T}}\mathbf{x}'$$

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\mathbf{x}^{\mathrm{T}}\mathbf{x}/2\sigma^2\right) \exp\left(\mathbf{x}^{\mathrm{T}}\mathbf{x}'/\sigma^2\right) \exp\left(-(\mathbf{x}')^{\mathrm{T}}\mathbf{x}'/2\sigma^2\right)$$

$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \tag{6.14}$$

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(k_1(\mathbf{x}, \mathbf{x}')\right) \tag{6.16}$$

Kernel methods for supervised learning

The general idea of kernel representations for supervised learning is we can implement smoothness as:

Similarity in input => similarity in output

$$\boxed{(\mathbf{x}_m \approx \mathbf{x}_n) \Rightarrow (\mathbf{t}_m \approx \mathbf{t}_n)}$$

Gaussian processes for function approximation

$$\boxed{(\mathbf{x}_m \approx \mathbf{x}_n) \Rightarrow (\mathbf{t}_m \approx \mathbf{t}_n)}$$

In the Gaussian process model the similarity is implemented in a probabilistic setting

For additive noise model t(**x**) = y(**x**) + e, we can represent the similarity as an assumed Gaussian distribution of the function values for a general set of inputs

$$\mathrm{cov}(y_1, \ldots, y_N) = \mathbf{K}, \quad p(\mathbf{y}|\mathbf{K}) = \frac{1}{|2\pi\mathbf{K}|^{\frac{1}{2}}} \exp(-\tfrac{1}{2}\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y})$$

Kernel methods for supervised learning

The Gaussian distribution of the target then follows

$$\mathrm{cov}(t_1, ..., t_N) \equiv \mathbf{C} = \mathbf{K} + \beta^{-1}\mathbf{1}$$

Predictive distribution

$$p(\mathbf{t}_{\mathrm{test}} | \mathbf{t}_{\mathrm{train}}, \mathbf{C}_{\mathrm{train,test}}) = \frac{p(\mathbf{t}_{\mathrm{test}}, \mathbf{t}_{\mathrm{train}} | \mathbf{C}_{\mathrm{train,test}})}{p(\mathbf{t}_{\mathrm{train}} | \mathbf{C}_{\mathrm{train}})}$$

Gaussian conditioning

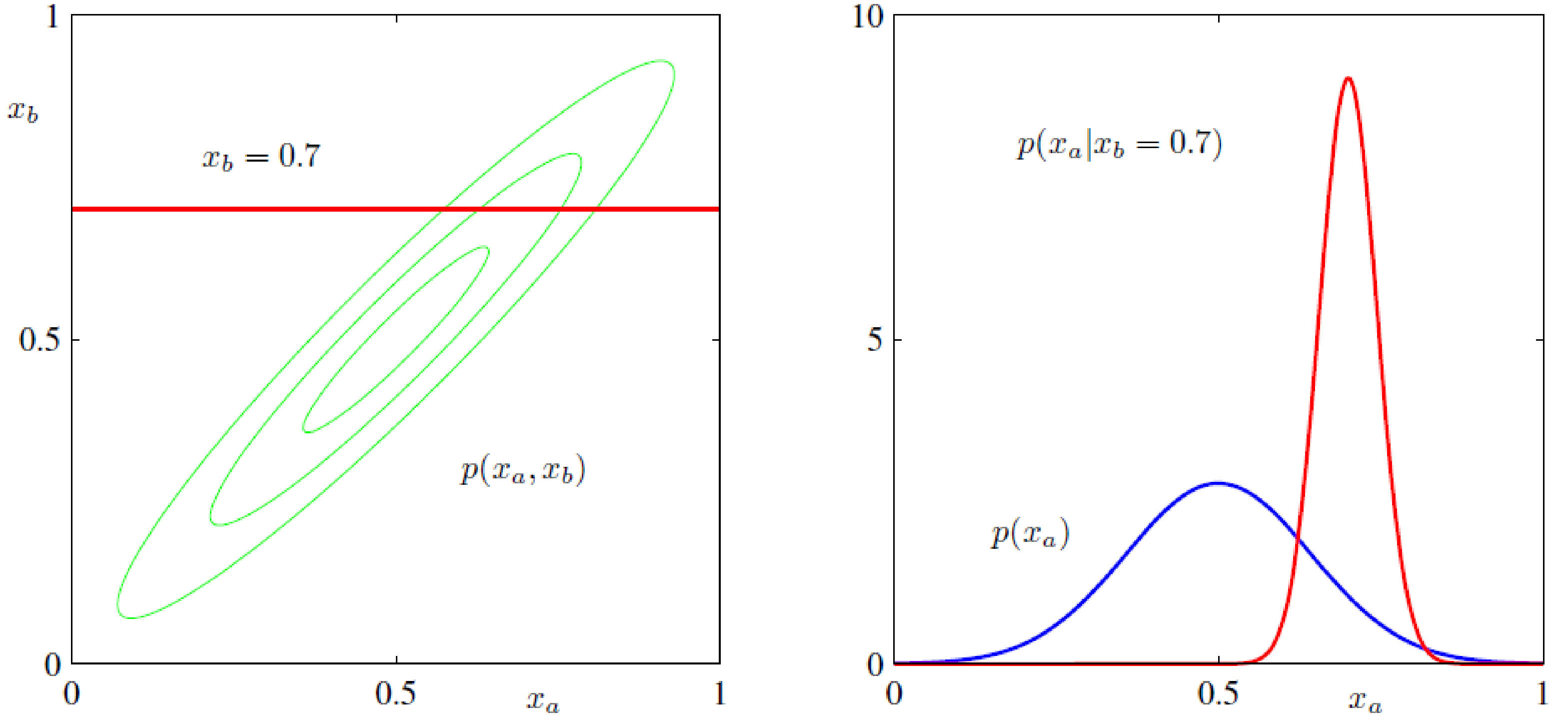


Figure 2.9 The plot on the left shows the contours of a Gaussian distribution $p(x_a, x_b)$ over two variables, and the plot on the right shows the marginal distribution $p(x_a)$ (blue curve) and the conditional distribution $p(x_a|x_b)$ for $x_b = 0.7$ (red curve).

Kernel methods for supervised learning

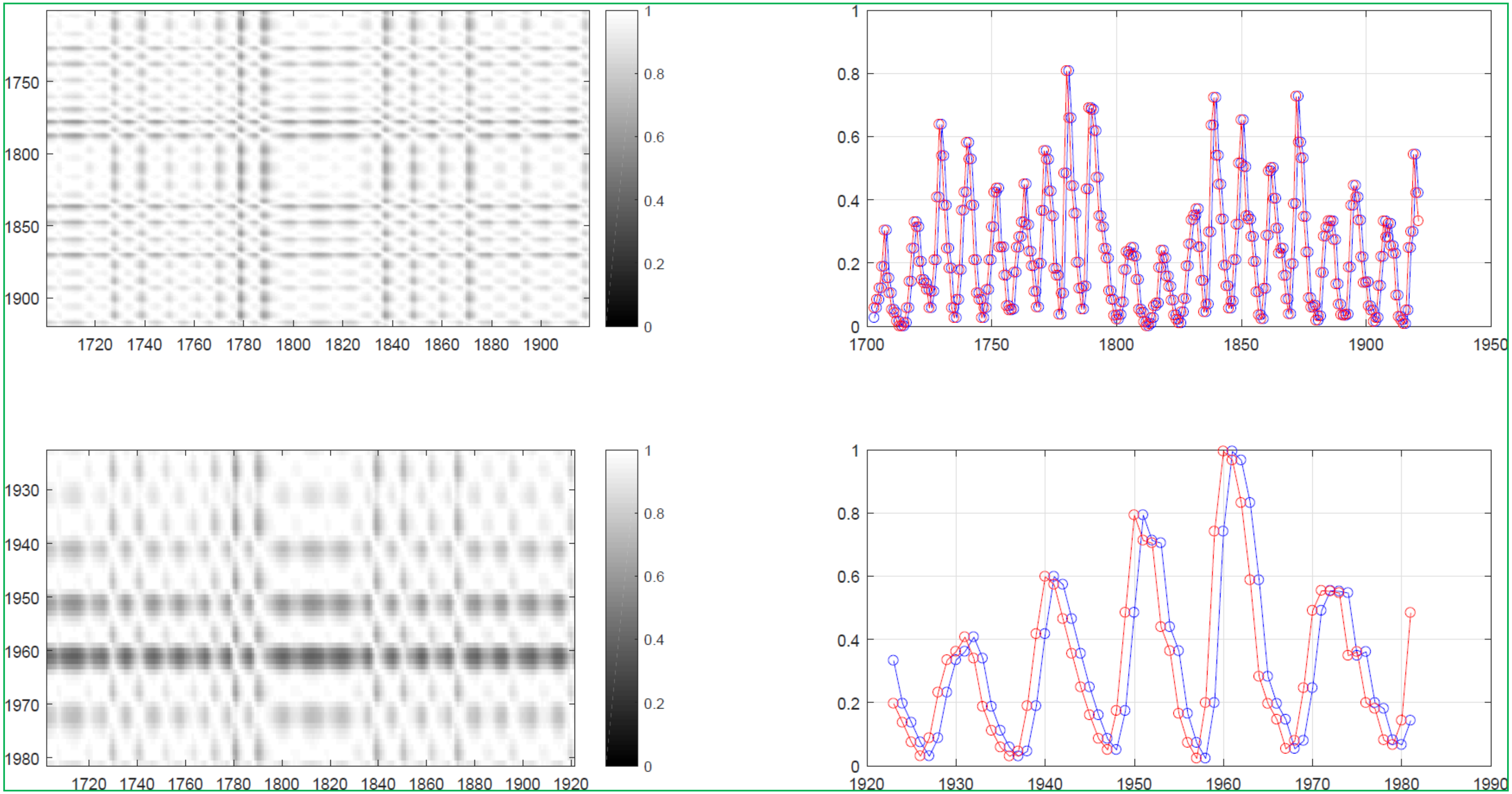
Rules for conditioning (Bishop Eq. (281-82))

$$\boxed{\begin{aligned} \boldsymbol{\mu}_{\text{test}|\text{train}} &= \mathbf{C}_{\text{test,train}} \mathbf{C}_{\text{train}}^{-1} \mathbf{t}_{\text{train}} \\ \mathbf{C}_{\text{test}|\text{train}} &= \mathbf{C}_{\text{test}} - \mathbf{C}_{\text{test,train}} \mathbf{C}_{\text{train}}^{-1} \mathbf{C}_{\text{train,test}} \end{aligned}}$$

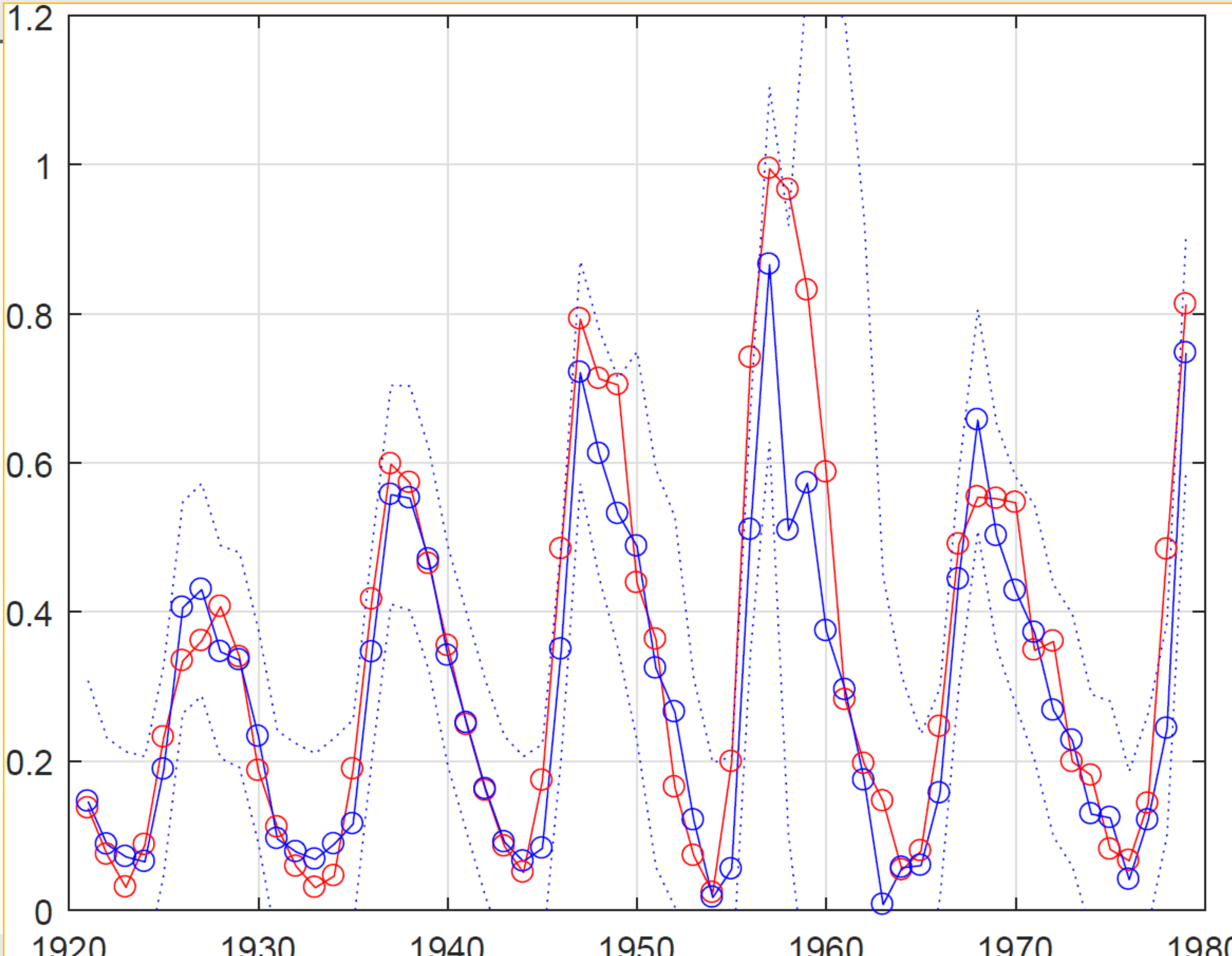
$$\hat{t}_m = (\boldsymbol{\mu}_{\text{test}|\text{train}})_m$$

$$\text{std}(t_m) = \sqrt{(\mathbf{C}_{\text{test}|\text{train}})_{m,m}} = \sqrt{(\mathbf{C}_{\text{test}})_{m,m} - (\mathbf{C}_{\text{test,train}} \mathbf{C}_{\text{train}}^{-1} \mathbf{C}_{\text{train,test}})_{m,m}}$$

Sun spot C_{train}, $C_{test,train}$



Confidence interval (2*σ)



A simple alternative derivation of the GP prediction (Bishop sec. 6.1)

$$\boxed{J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}$$

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}\phi(\mathbf{x}_n) = \sum_{n=1}^{N} a_n\phi(\mathbf{x}_n) = \mathbf{\Phi}^{\mathrm{T}}\mathbf{a}$$

$$\boxed{J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{a}}$$

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a}. \qquad \Rightarrow \qquad \mathbf{a} = (\mathbf{K} + \lambda\mathbf{I}_N)^{-1}\mathbf{t}.$$

$$\boxed{y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\phi(\mathbf{x}) = \mathbf{k}(\mathbf{x})^{\mathrm{T}}\left(\mathbf{K} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{t}}$$

Support vector machines

The general idea of kernel representations for supervised learning is that similarity in input should lead to similarity in output

For support vectors this is assumed for labels

$$\boxed{(\mathbf{x}_m \approx \mathbf{x}_n) \Rightarrow (\mathbf{t}_m \approx \mathbf{t}_n)}$$

Use kernel to implement similarity and linear discriminant to make decision

Maximum margin principle

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b$$

Margin = $|y(\mathbf{x})| / \|\mathbf{w}\|$

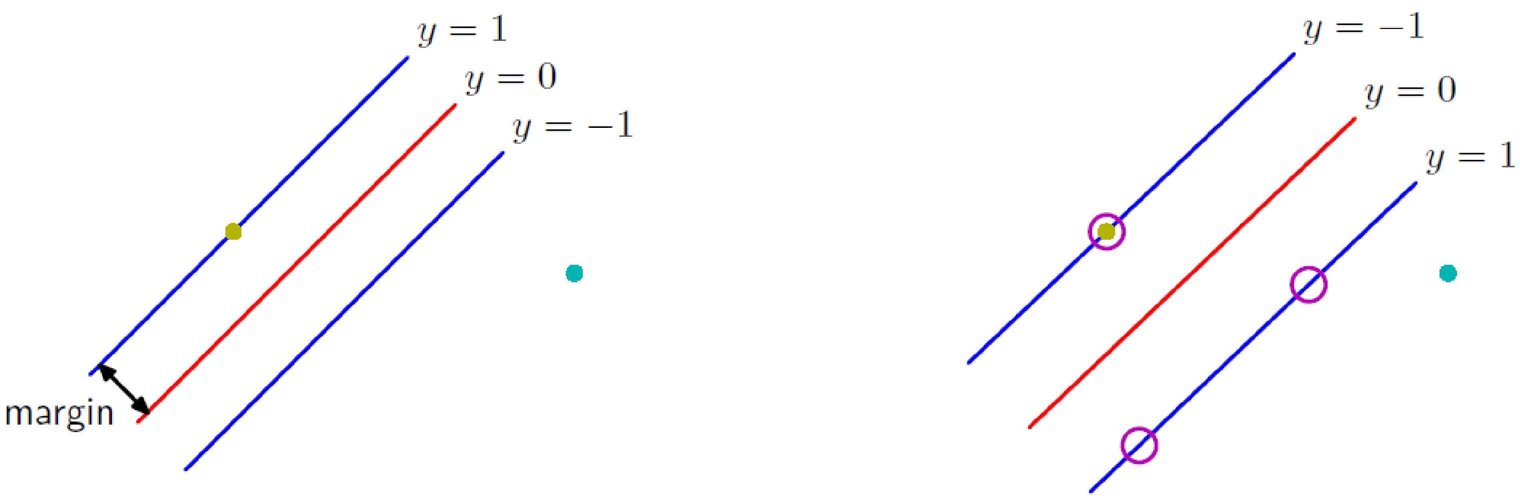
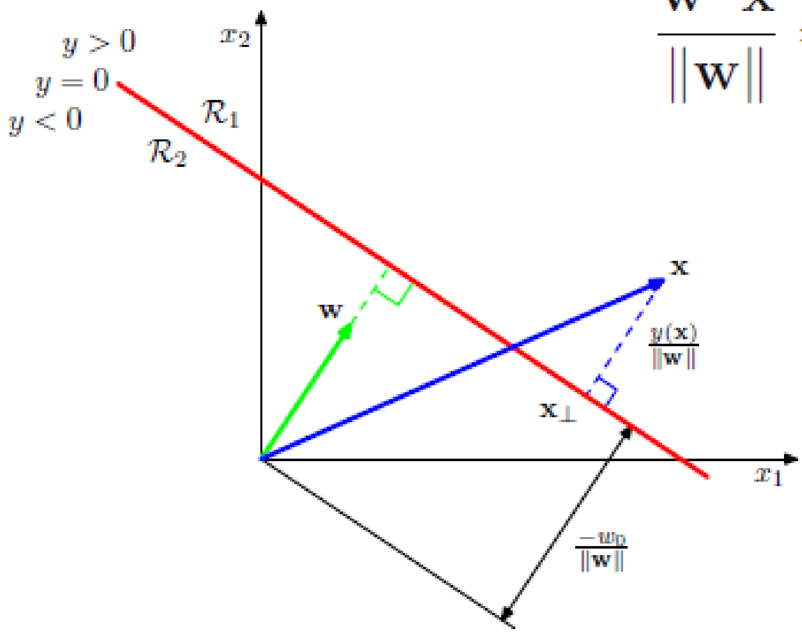


Figure 7.1 The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by a subset of the data points, known as support vectors, which are indicated by the circles.

Geometry of linear discriminant

Figure 4.1 Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to $\mathbf{w}$, and its displacement from the origin is controlled by the bias parameter w_0. Also, the signed orthogonal distance of a general point $\mathbf{x}$ from the decision surface is given by $y(\mathbf{x})/\|\mathbf{w}\|$.



$$\frac{\mathbf{w}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|}$$

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|}.$$

$$\boxed{y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) + b}$$

Maximum margin principle

Margin = $|y(\mathbf{x})| / \|\mathbf{w}\|$

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}$$

$$\underset{\mathbf{w},b}{\arg\max}\left\{\frac{1}{\|\mathbf{w}\|}\min_n\left[t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right)\right]\right\}$$

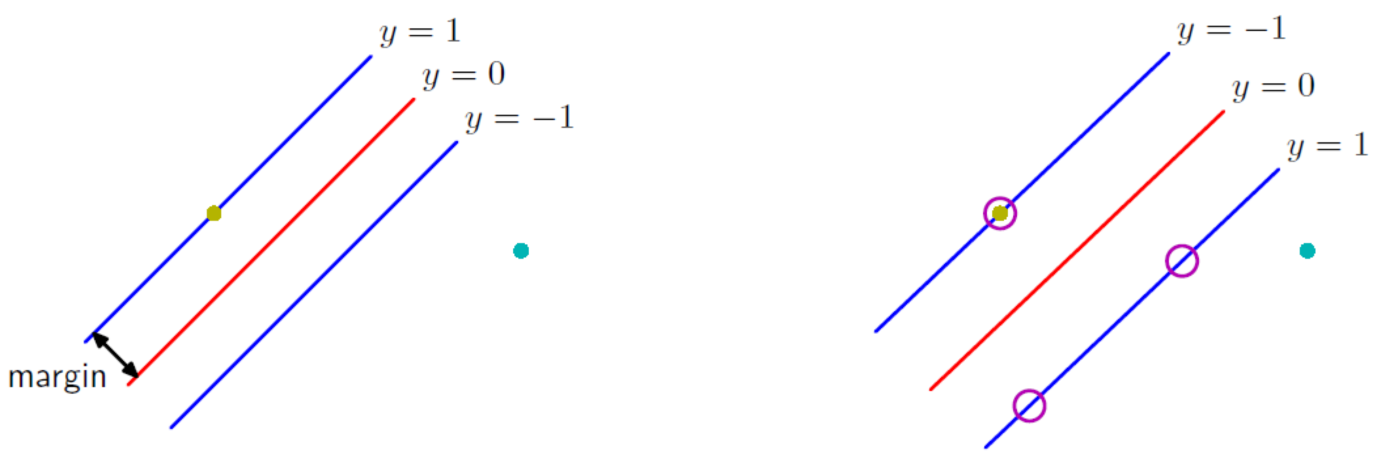


Figure 7.1 The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by a subset of the data points, known as support vectors, which are indicated by the circles.

Convex criteria

rescaling $\mathbf{w} \to \kappa\mathbf{w}$ and $b \to \kappa b$

$$t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right)=1$$

For points closest to y=0

$$t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right)\geqslant 1$$

For all points

$$\arg\max_{\mathbf{w},b}\left\{\frac{1}{\|\mathbf{w}\|}\min_n\left[t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right)\right]\right\}$$

$$\arg\min_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2$$

with the above constraint

Convex optimization

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b) - 1 \right\}$$

$$\begin{aligned} \mathbf{w} &= \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \\ 0 &= \sum_{n=1}^{N} a_n t_n. \end{aligned}$$

Convex optimization = quadratic optimization w. constraint

Eliminating $\mathbf{w}$ and b from $L(\mathbf{w}, b, \mathbf{a})$ using these conditions then gives the *dual representation* of the maximum margin problem in which we maximize

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{7.10}$$

with respect to $\mathbf{a}$ subject to the constraints

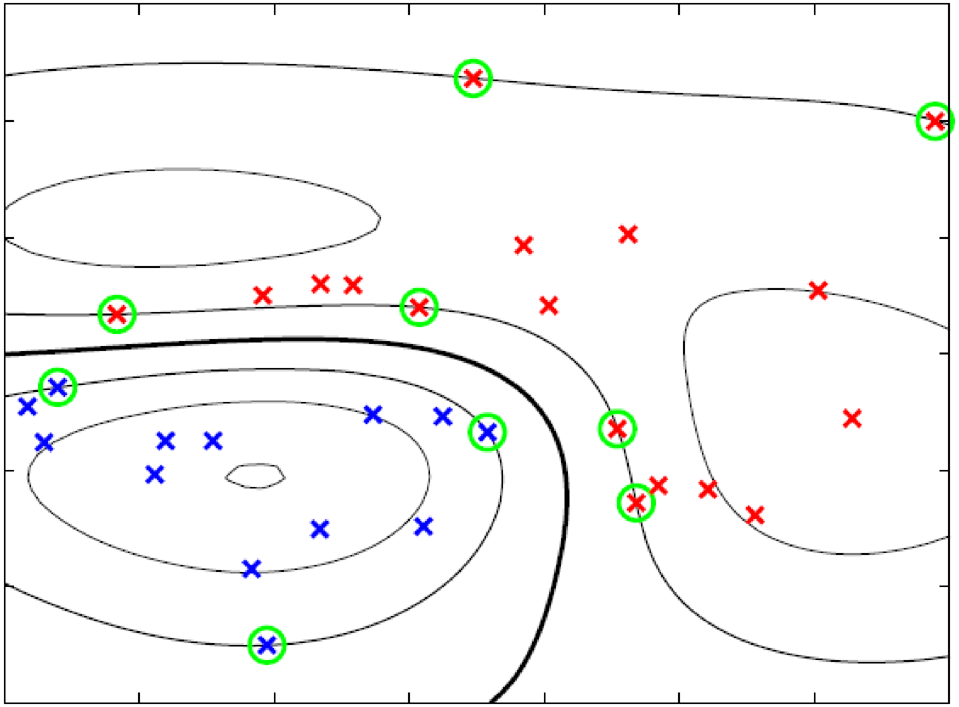
$$a_n \geqslant 0, \qquad n = 1, \ldots, N, \tag{7.11}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{7.12}$$

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{x}')$$

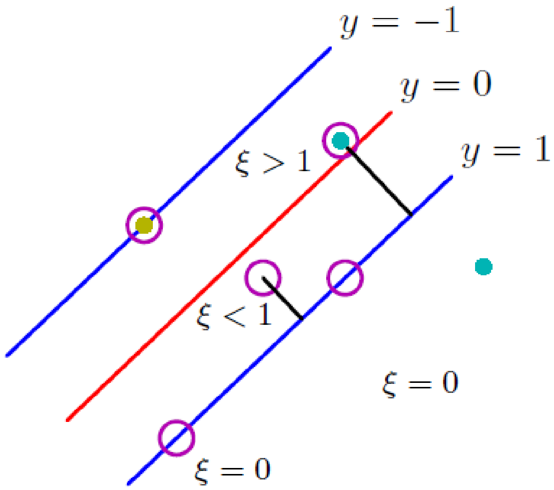
$$\boxed{\hat{t}_m = \operatorname{sign}\left(\sum_{n=1}^{N} a_n t_n K(\mathbf{x}_m, \mathbf{x}_n) + b\right)}$$

Figure 7.2 Example of synthetic data from two classes in two dimensions showing contours of constant $y(\mathbf{x})$ obtained from a support vector machine having a Gaussian kernel function. Also shown are the decision boundary, the margin boundaries, and the support vectors.



Slack variables for noisy data (non-separable)

Figure 7.3 Illustration of the slack variables $\xi_n \geqslant 0$. Data points with circles around them are support vectors.



$$t_n y(\mathbf{x}_n) \geqslant 1 - \xi_n,$$

Cost associated with slack

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\mathbf{w}\|^2$$

Augmented Lagrange function (a, μ parameters positive)

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N} a_n \{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n$$

$$\boxed{\frac{\partial L}{\partial \xi_n} = 0 \quad \Rightarrow \quad a_n = C - \mu_n} \quad => \quad \boxed{0 \leqslant a_n \leqslant C}$$

Slack variables for noisy data (non-separable)

Modified optimization problem

$$
\begin{aligned}
\tilde{L}(\mathbf{a}) &= \sum_{n=1}^{N} a_n - \sum_{n,m=1}^{N} a_n a_m t_n t_m K(\mathbf{x}_n, \mathbf{x}_m), \\
&\quad 0 \leq a_n \leq C, \\
&\quad \sum_{n=1}^{N} a_n t_n = 0, \\
b &= \frac{1}{|M|} \sum_{n \in M} \left(t_n - \sum_{m \in S} a_m t_m K(\mathbf{x}_n, \mathbf{x}_m) \right).
\end{aligned}
$$

Interpretation of coefficients:

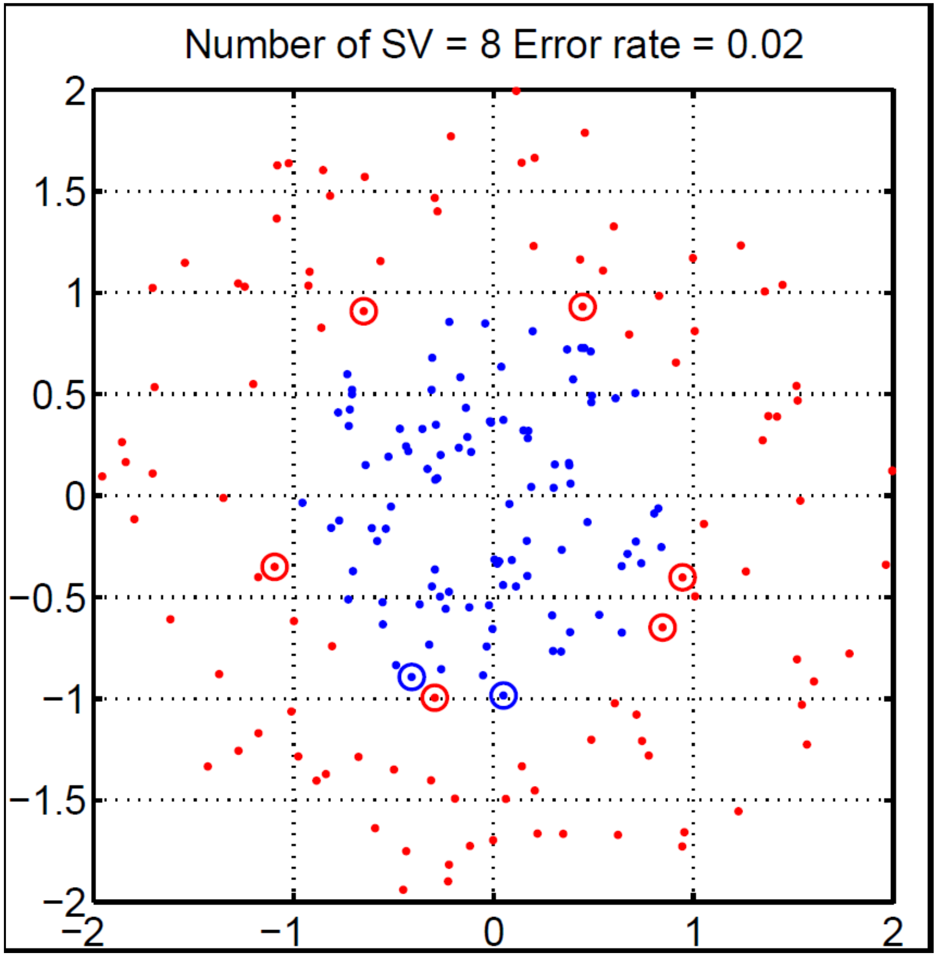
a > 0: support vectors, enters classification

0< a < C: on the boundary

a = C: slack “activated” => inside margin; misclass if ξ > 1

$$
a_n \left(t_n y(\mathbf{x}_n) - 1 + \xi_n \right) \quad = \quad 0
$$

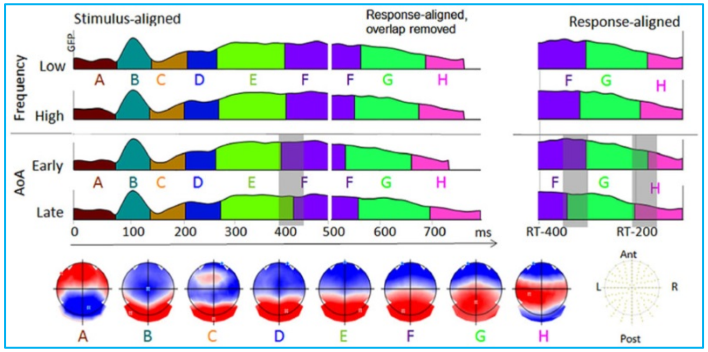
Simple 2D synthetic data

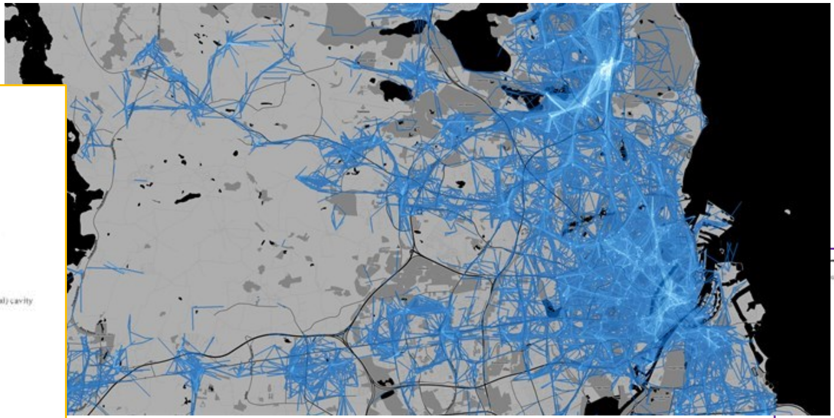


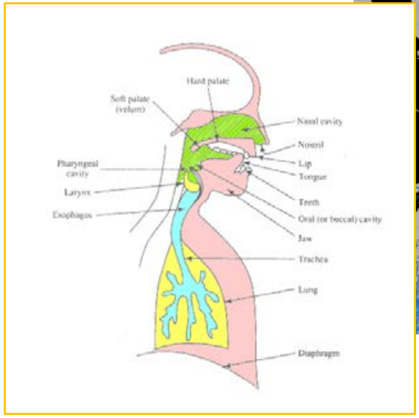
Dynamical systems represented as symbolic sequences

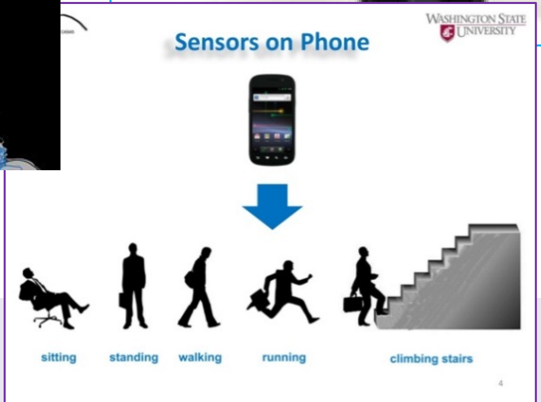
Many real life applications are based on symbolic, label sequence based representations

Speech, phonemes
Gesture, activity recognition
Text, words, topics
Mobility, stop locations
EEG, micro-states, meso-states)









Spotify

2012/11/10/spotify-is-having-a-good-2012-revenues-could-reach-500m-as-it-expands-the-digital-music-market/

nsen@gmail... × | Spotify Is Having A Good 2012: Reven... ×

Spotify Is Having A Good 2012: Revenues Could Reach $500M As It Expands The Digital Music Market

Posted Nov 10, 2012 by Eric Eldon (@eldon)

37 Like 0 Tweet 931 Share 0 Next Story

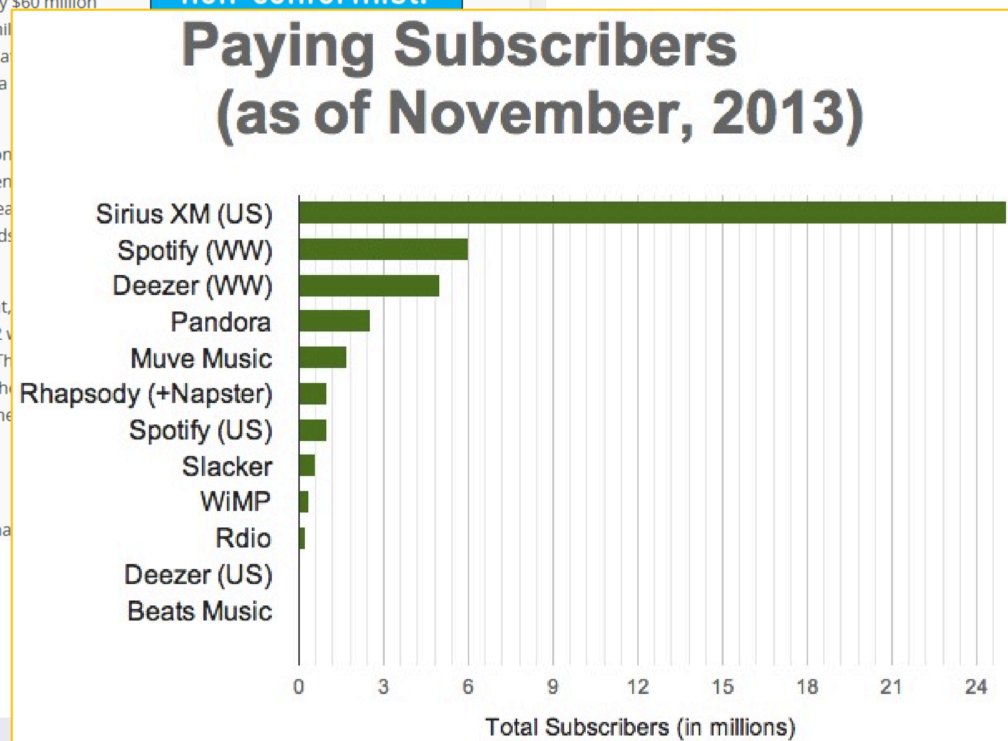
Spotify, the streaming music startup, was having serious trouble paying its bills, if you believed reports from earlier this year. Its 2011 financials showed a loss of nearly $60 million on revenues of $244 mil information is out of da the company has had a strong 2012.

It made $200 million in total revenue over the first six months of 2012, and is on run-rate that could put it around $500 million by January, according to confiden information leaked to me by industry sources. Despite another net loss this yea business model — free streaming music with ads, $5 for web access with no ads month for ad-free plus mobile access — is going in the right direction.

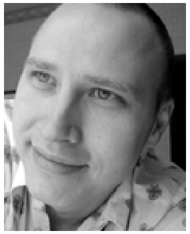
Its deal structure with the labels requires either a $200 million annual payment, had to do last year, or around 75% of total revenue (whichever is higher). 2012 v first year that revenue is high enough for the percentage structure to kick in. Th is projecting profit after cost-of-sales to be around $60 million. It still has anoth million in engineering, marketing, sales and other operating costs, on top of the burden, so it'll likely post an annual loss of roughly $40 million.

The company didn't comment on these numbers when we asked.

The situation still sounds painful, but it's not bad at all. The record labels are ha



WinAmp demo project



Lehn-Schiøler, Garcias-Arena, Petersen, Hansen: ISMIR 2006 (Oct 9, 2006)

CASTSEARCH - CONTEXT BASED SPEECH DOCUMENT RETRIEVAL

Lasse Lohilahti Mølgaard, Kasper Winther Jørgensen, and Lars Kai Hansen

Informatics and Mathematical Modelling
Technical University of Denmark Richard Petersens Plads
Building 321, DK-2800 Kongens Lyngby, Denmark

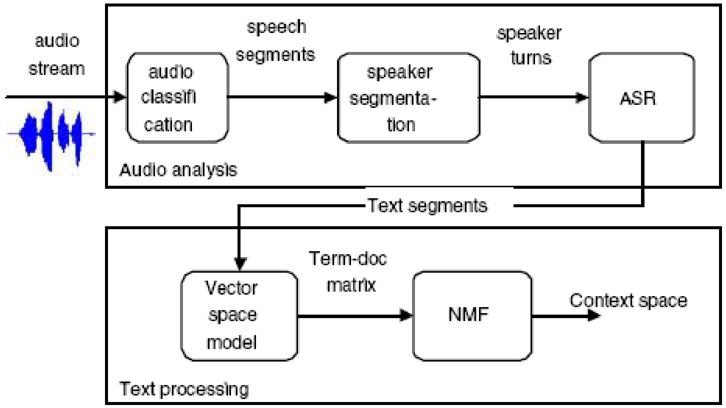
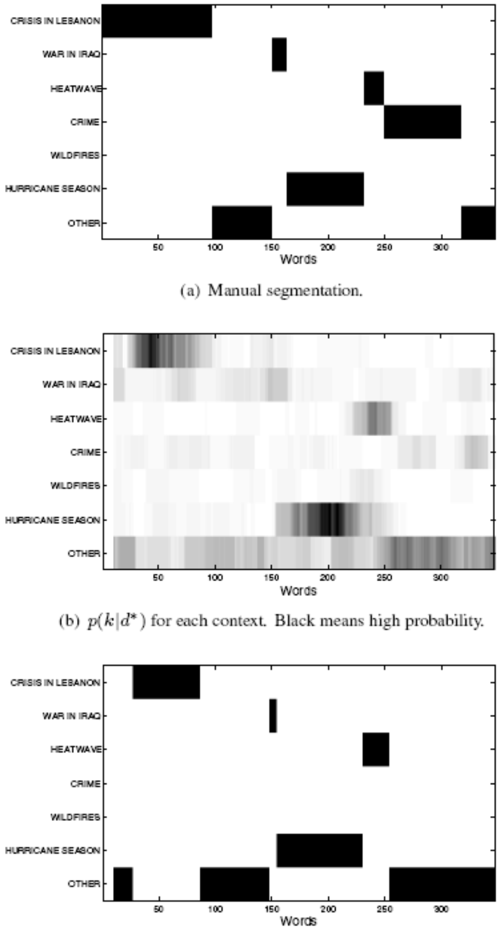


Fig. 1. The system setup. The audio stream is first processed using audio segmentation. Segments are then using an automatic speech recognition (ASR) system to produce text segments. The text is then processed using a vector representation of text and apply non-negative matrix factorization (NMF) to find a topic space.

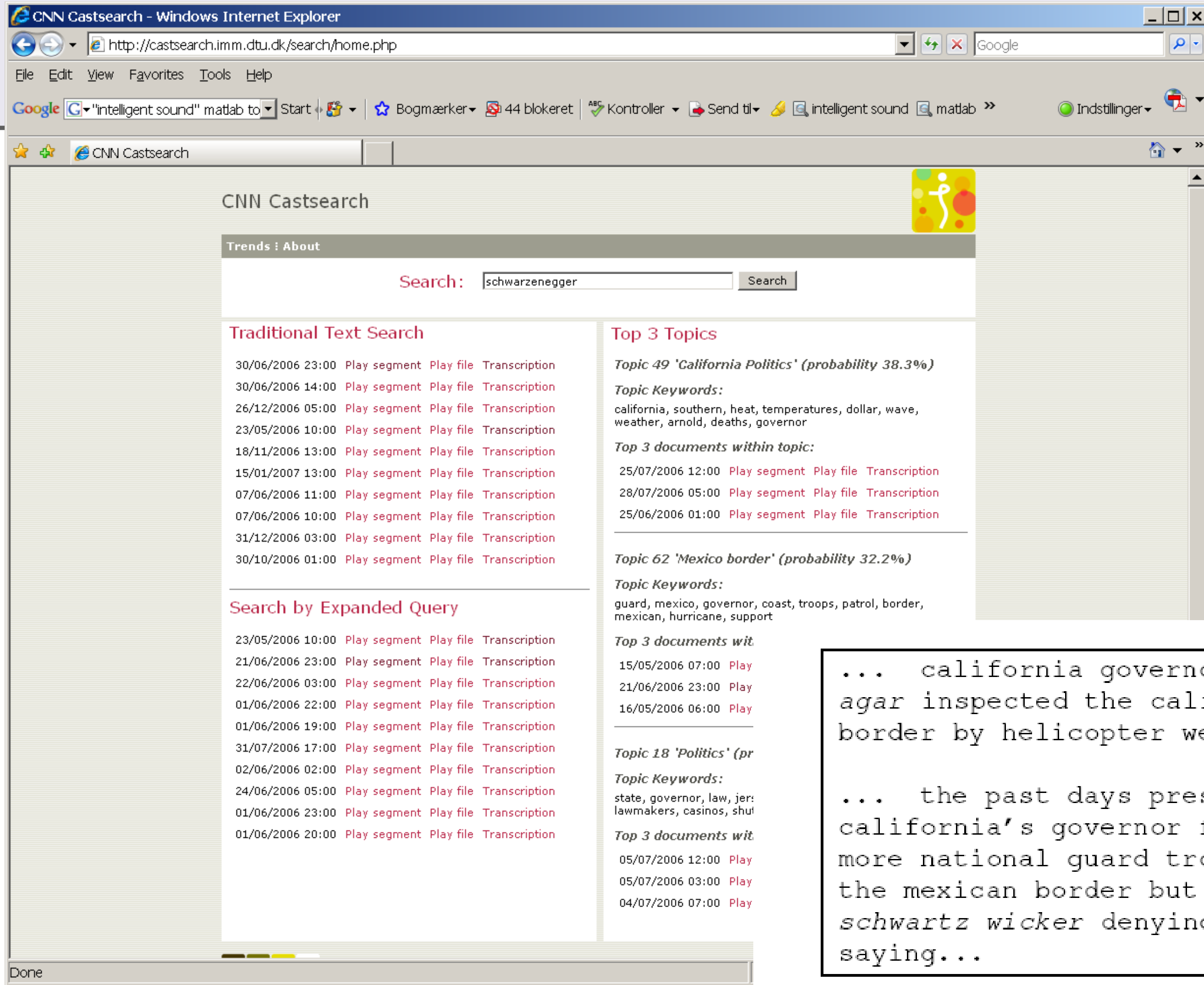


(a) Manual segmentation.

(b) $p(k|d^*)$ for each context. Black means high probability.

(c) The segmentation based on $p(k|d^*)$.

Fig. 3. Figure 3(a) shows the manual segmentation of the news show into 7 classes. Figure 3(b) shows the distribution $p(k|d^*)$ used to do the actual segmentation shown in figure 3(c). The NMF-segmentation is in general consistent with the manual segmentation. Though, the segment that is manually segmented as 'crime' is labeled 'other' by the NMF-segmentation



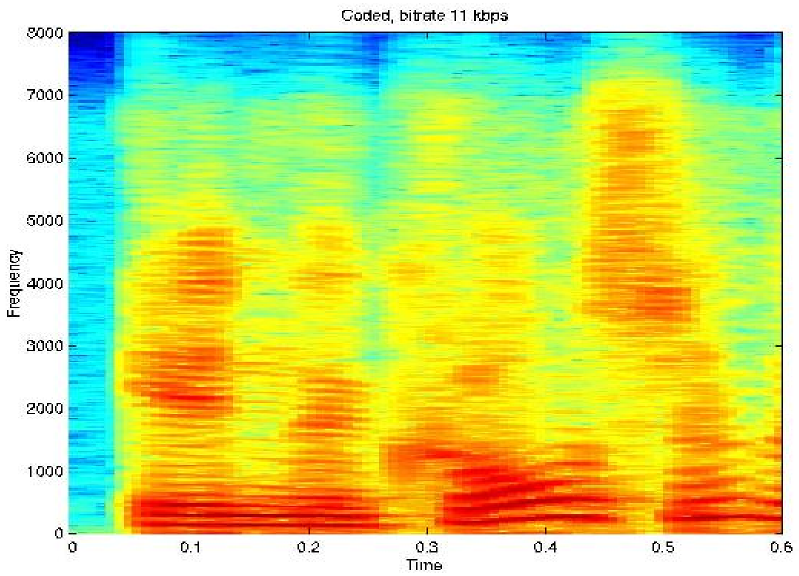
... california governor arnold's *fortson agar* inspected the california mexico border by helicopter wednesday to see ...

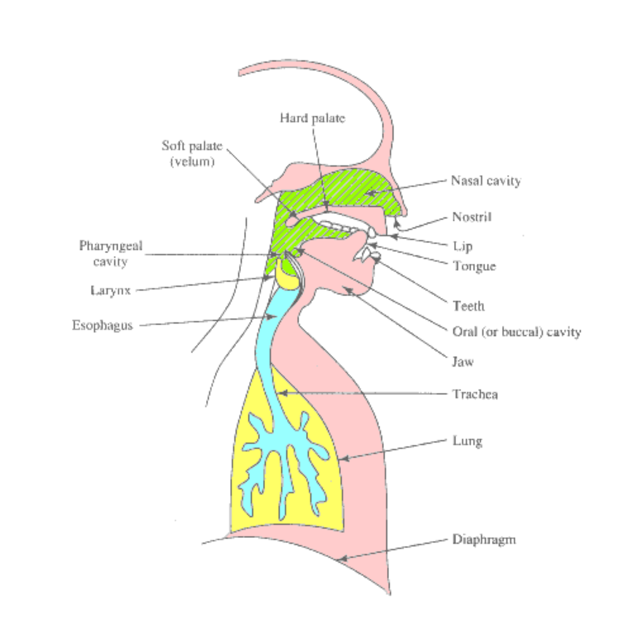
... the past days president bush asking california's governor for fifteen hundred more national guard troops to help patrol the mexican border but governor orville *schwartz wicker* denying the request saying...

Fig. 2. Two examples of the retrieved text for a query on 'schwarzenegger'.

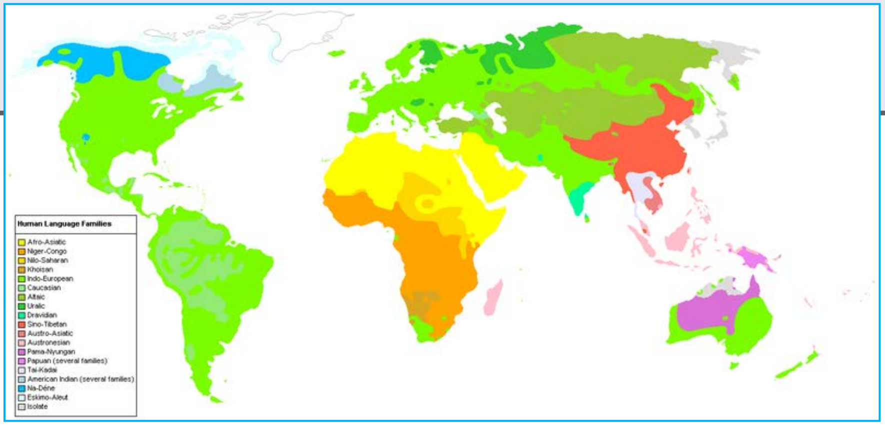
castsearch.imm.dtu.dk

Sequence model example: Audio DSP



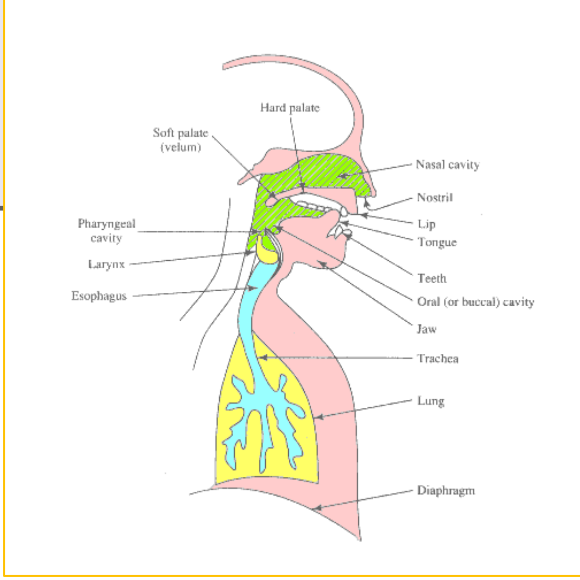


SPEECH –some definitions



- Speech signals are sequences of sounds
- The basic sounds and the transitions between them serve as symbolic representation of information - **semantics**
- The arrangement of these sounds (symbols) is governed by the rules of ***language***
- The study of these rules and their implications in human communication is called ***linguistics***
- The study of and classification of the sounds of speech is called ***phonetics***

SPEECH PRODUCTION



Speech is produced by the human vocal tract

The vocal tract is excited either by short burst of periodic stimulus or white noise

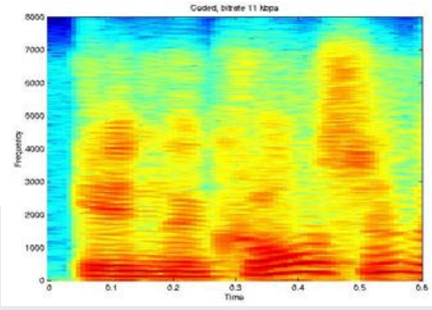
Voiced sounds are produced by an airflow through tight vocal cords.

Unvoiced sounds are produced by turbulent flow.

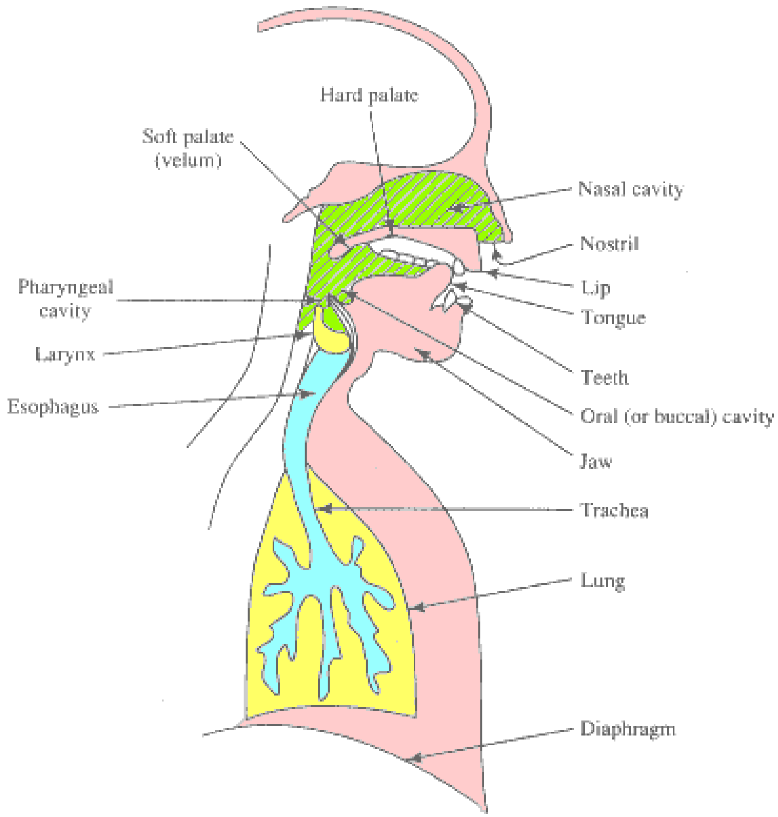
Sounds are classified broadly into phonemes: Vowels and consonants
More complex structures include diphthongs, semi-vowels.

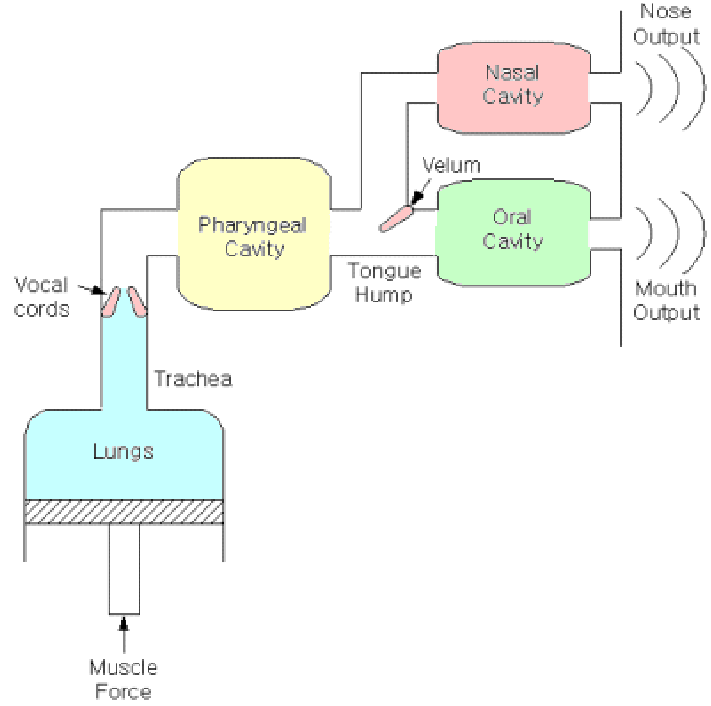
Formants are peaks in the power spectrum modulation (envelopes).

Frequencies in the range:
F1 (270-730 Hz), F2 (840-2290 Hz), and F3 (1690-3010 Hz).



Modeling speech

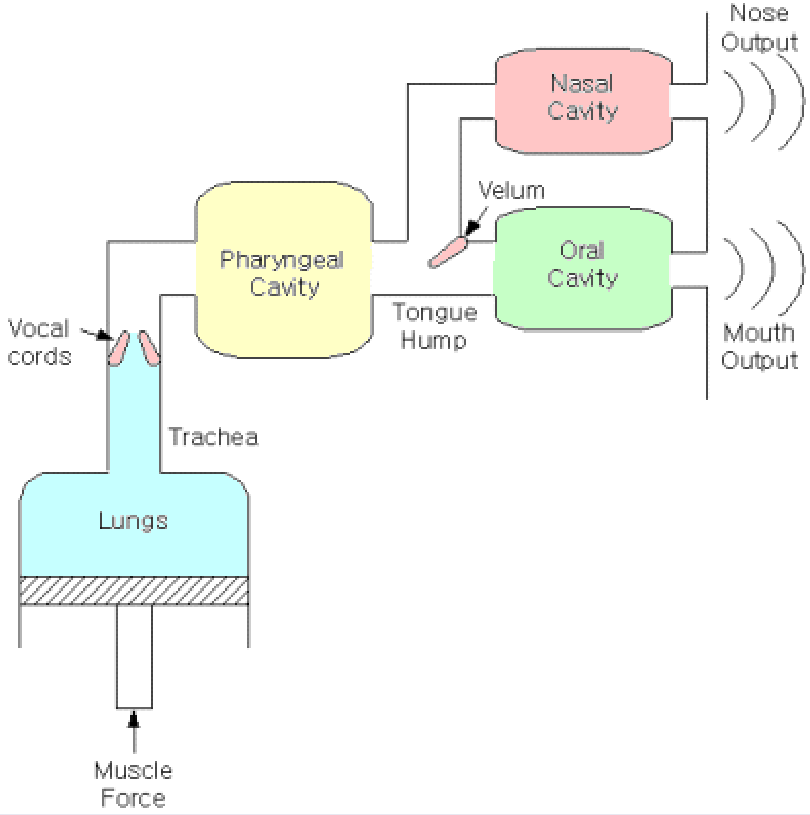




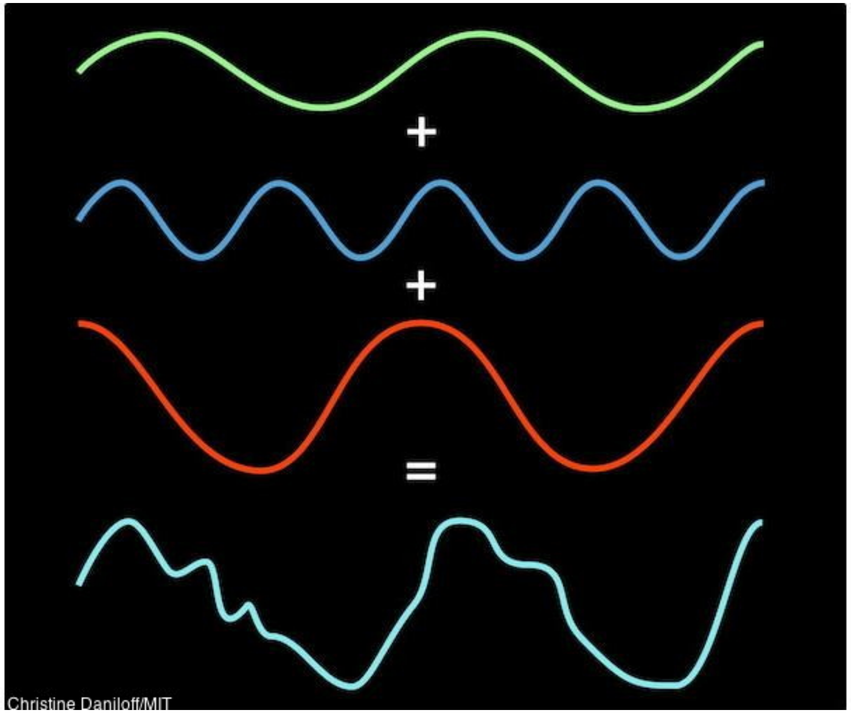
The speech signal is modelled as a linear time variant system excited by a high-frequency signal (periodic og random)

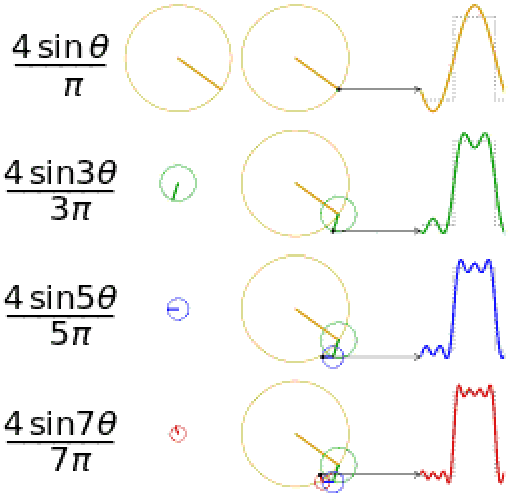
- Linear system model (IIR model)

$$x(n) \; = \; \sum_{j=1}^{p} w_j x(n-j) + \epsilon(n)$$



Fourier transform

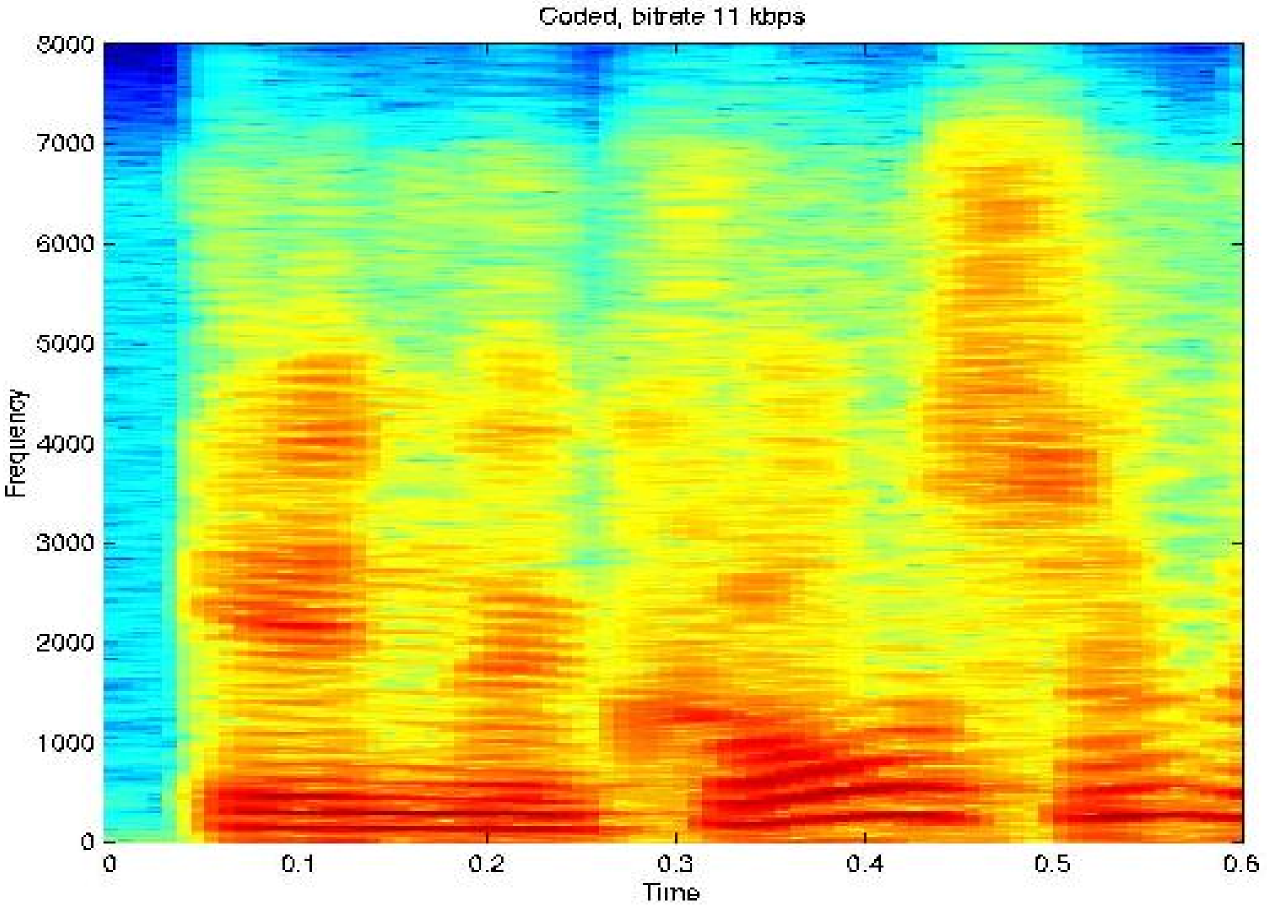




https://en.wikipedia.org/wiki/Fourier_series

http://en.wikipedia.org/wiki/Discrete_Fourier_transform

Speech spectrogram



CEPSTRAL COEFFICIENTS

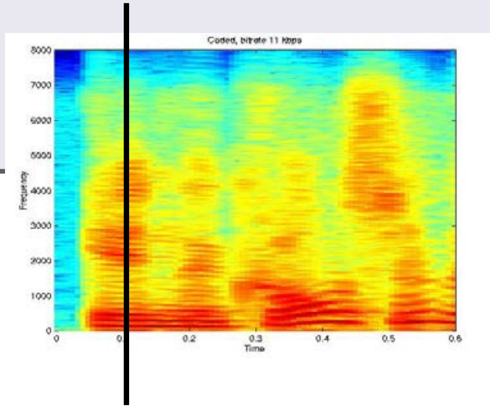
- Linear system model (IIR model)

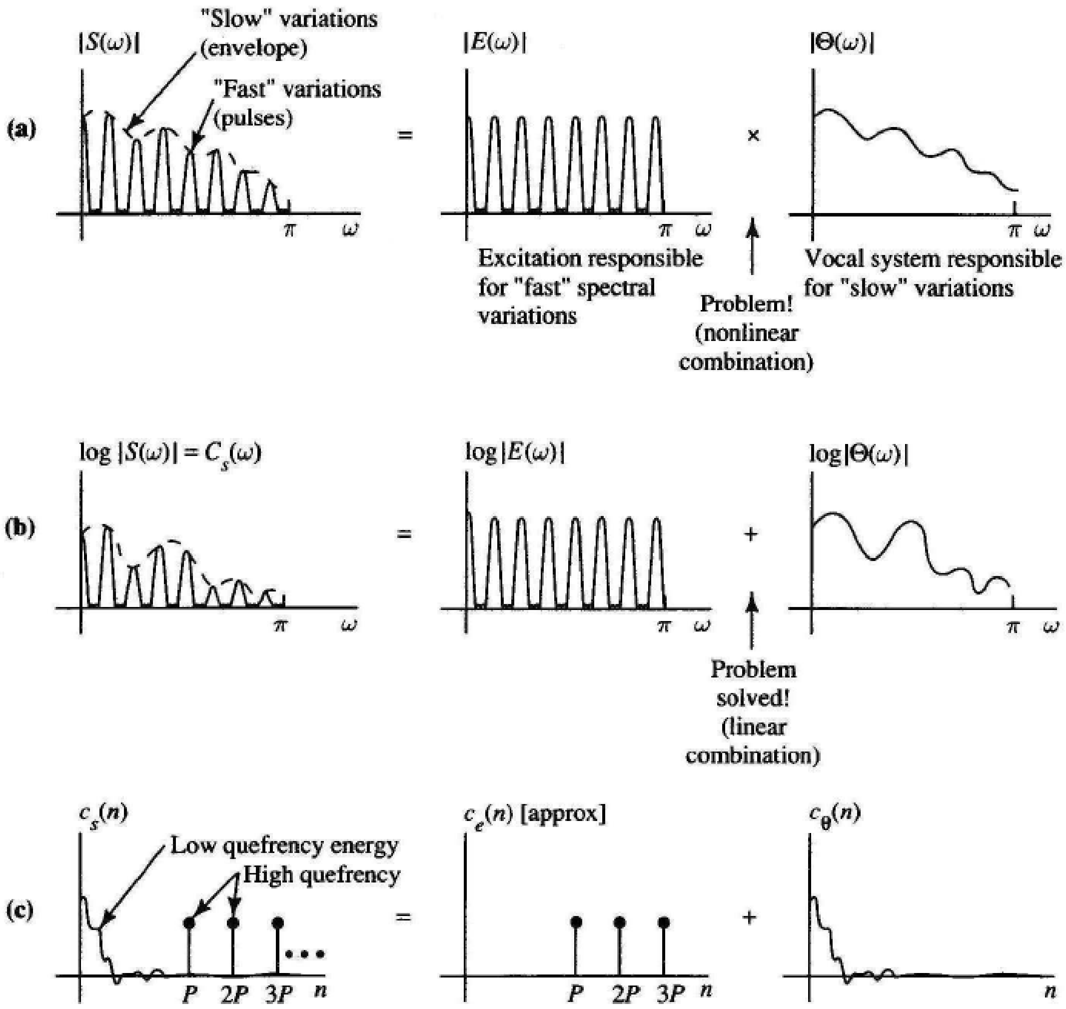
$$
\begin{aligned}
s(n) &= \sum_{j=1}^{p} w(j) \cdot s(n-j) + \epsilon(n) \\
s(n) &= \sum_{i} \theta(i) \cdot \epsilon(n-i)
\end{aligned}
$$

- In the Fourier domain

$$
\begin{aligned}
S(\omega) &= \Theta(\omega) \cdot E(\omega) \\
\log|S(\omega)| &= \log|\Theta(\omega)| + \log|E(\omega)|
\end{aligned}
$$

Cepstral liftering





Phoneme grouping and sequence recognition

Provides discrete symbols for identification of word.

The audio stream is segmented into a string of symbols (~phonemes), by assigning windows to most likely phoneme using K-means clustering.

Cepstral analysis → Clustering of features → Recognition of word sequences

Markov process / chain

Andrey Markov (1906)

A string of symbols ‘y_n’ is a Markov chain if

$$p(y_n \mid y_1, y_2, \ldots y_{n-2}\, y_{n-1}) = p(y_n \mid y_{n-L}, \ldots y_{n-2}\, y_{n-1})$$

A string of symbols is 1’st order Markovian if

$$p(y_n \mid y_1, y_2, \ldots y_{n-2}\, y_{n-1}) = p(y_n \mid y_{n-1})$$

SIMPLE MARKOV MODEL

Let y_n be a sequence of symbols with K states

Let $a_{j,j'}$ be the probability of jumping from j to j'

i.e., $a_{j,j'} = p(y_n = j' | y_{n-1} = j)$

Matrix $a_{j,j'}$ is a stochastic matrix $\Sigma_{j'} a_{j,j'} = 1$

a can be estimated by maximum likelihood

Markov chain estimation by maximum likelihood

The likelihood can be written, introducing the observed number of transitions j => j' : $n_{j,j'}$

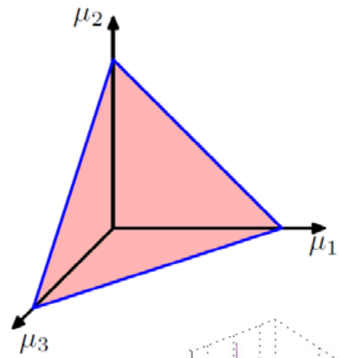
$$
\begin{aligned}
P(\{y_n\}|a) &= P(y_1)\prod_{n=2}^{N} P(y_n|y_{n-1},a) \\
&= P(y_1)\prod_{j,j'} (a_{j,j'})^{\mathrm{n}_{\mathrm{j,j'}}}
\end{aligned}
$$

Estimator

$$
\widehat{a}_{j,j'} = \frac{n_{j,j'}}{\sum_{j''} n_{j,j''}}
$$

MAP estimate with Dirichlet priors on $a_{j,:}$

The Dirichlet distribution over three variables μ_1, μ_2, μ_3 is confined to a simplex (a bounded linear manifold) of the form shown, as a consequence of the constraints $0 \leqslant \mu_k \leqslant 1$ and $\sum_k \mu_k = 1$.



$$\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$$

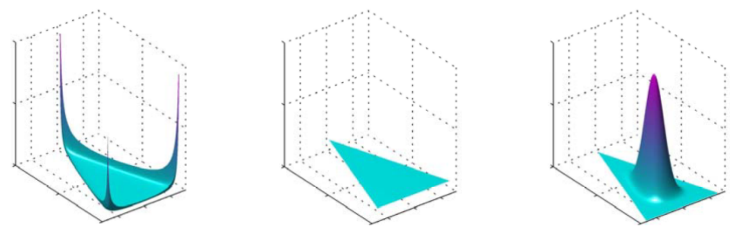


Figure 2.5 Plots of the Dirichlet distribution over three variables, where the two horizontal axes are coordinates in the plane of the simplex and the vertical axis corresponds to the value of the density. Here $\{\alpha_k\} = 0.1$ on the left plot, $\{\alpha_k\} = 1$ in the centre plot, and $\{\alpha_k\} = 10$ in the right plot.

$$\begin{aligned} \widehat{a}_{j,j'} &= \frac{n_{j,j'} + \alpha_{j,j'} - 1}{\sum_{j''} n_{j,j''} + \alpha_{j,j''} - 1} \\ &= \frac{n_{j,j'} + \alpha_{j,j'} - 1}{\sum_{j''} n_{j,j''} + \alpha_{j,j''} - 1} \end{aligned}$$

Analysis of Markov chains - The ensemble picture

Consider a large number of 'parallel' Markov chains $(y^i)_n$
Each chain makes random moves according to a common a-matrix (stationarity)
The probability over states, at time step n, obeys the Kolmogorov Chapman equation

$$P_{n+1}(j') = \sum_{j=1}^{K} P_n(j) a_{j,j'} \qquad \mathbf{P}_{n+1} = \mathbf{P}_n \mathbf{A}$$

The stationary distribution (Perron–Frobenius theorem, criterion for uniqueness) is a (left) eigenvector, with unit eigenvalue

$$P_*(j') = \sum_{j=1}^{K} P_*(j) a_{j,j'}. \qquad \mathbf{P}_* = \mathbf{P}_* \mathbf{A}$$

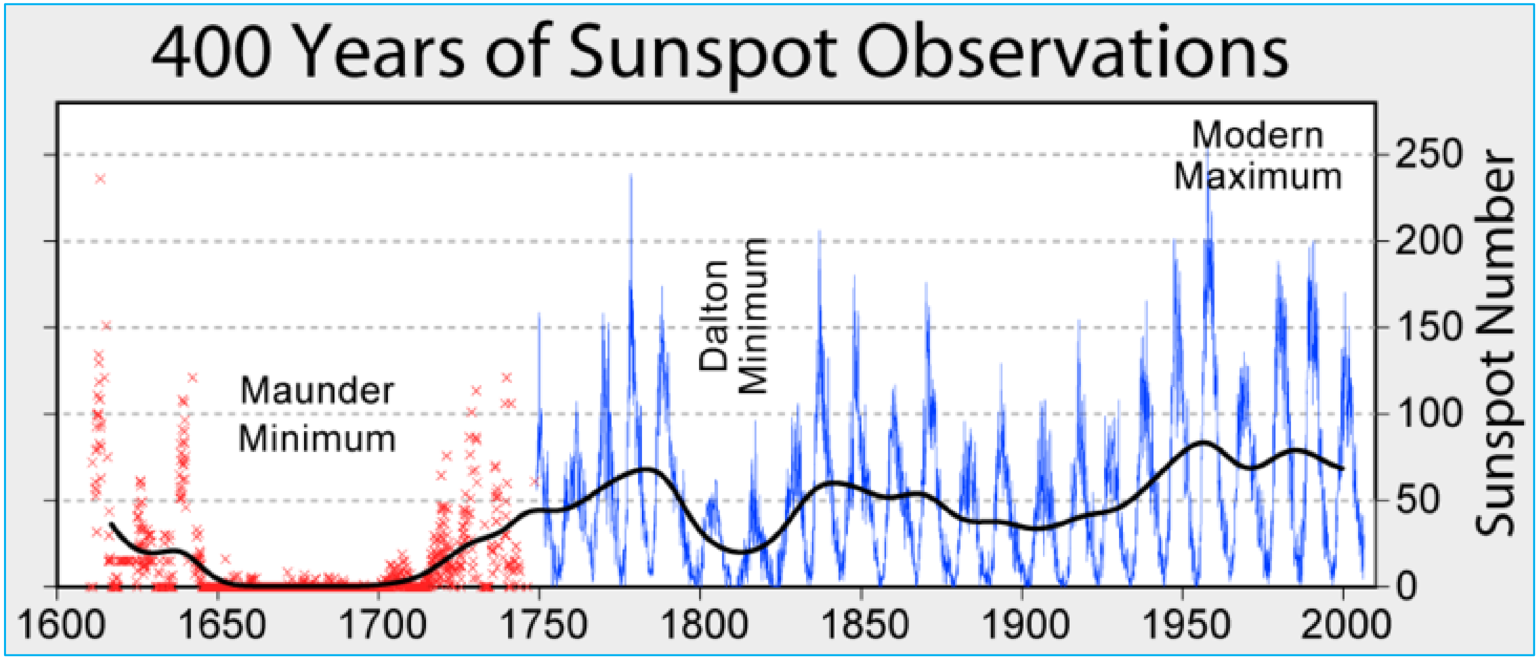
Detection based on sequences

$$P(\mathrm{B}|\{y_n\}) = \frac{P(\{y_n\}|\mathrm{B})P(\mathrm{B})}{P(\{y_n\})}.$$

$$P(\mathbf{A}_\mathrm{B}|\{y_n\}) = \frac{P(\{y_n\}|\mathbf{A}_\mathrm{B})P(\mathbf{A}_\mathrm{B})}{P(\{y_n\})}.$$

$$\begin{aligned} P(\{y_n\}|a) &= P(y_1)\prod_{n=2}^{N} P(y_n|y_{n-1}, a) \\ &= P(y_1)\prod_{j,j'} (a_{j,j'})^{\mathrm{n_{j,j'}}} \end{aligned}$$

What if data is non-stationary?



Issues

How to track the changes in the model?

Window based - assuming local stationarity

Tracking slow changes

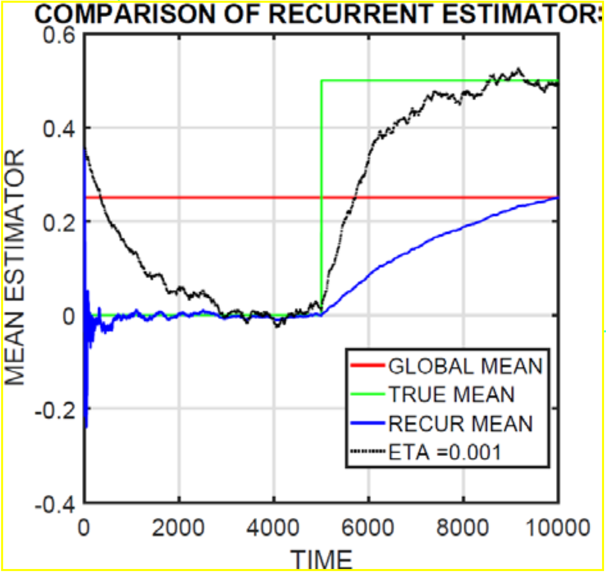
How to evaluate the error?

"Next sample error" - window

Dynamic estimator of the mean

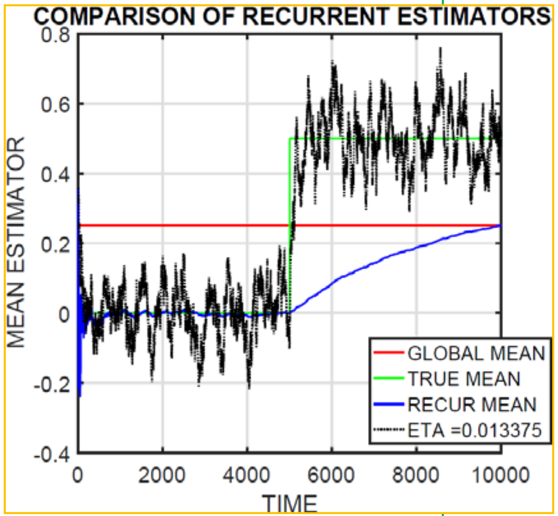
Dynamic updates for stream of data $\{x_1, x_2, ..., x_N\}$, $\mu = \frac{1}{N}\sum_{n=1}^{N} x_n$

$$
\begin{aligned}
\mu_N &= \frac{1}{N}x_N + \frac{1}{N}\sum_{n=1}^{N-1} x_n \\
&= \frac{1}{N}x_N + \frac{N-1}{N}\frac{1}{N-1}\sum_{n=1}^{N-1} x_n \\
&= \frac{1}{N}x_N + \frac{N-1}{N}\mu_{N-1} \\
&= \mu_{N-1} + \frac{1}{N}(x_N - \mu_{N-1})
\end{aligned}
$$



Non-stationarity, stochastic gradient

$$
\begin{aligned}
\mu_N &= \mu_{N-1} - \eta \frac{\partial}{\partial \mu}\left[\frac{1}{2}(x_N - \mu)^2\right]_{N-1} \\
&= \mu_{N-1} + \eta(x_N - \mu_{N-1})
\end{aligned}
$$



Difference equation has explicit solution

$$
\begin{aligned}
\mu_N &= \sum_{n=1}^{N} \eta(1-\eta)^{N-n} x_n \\
&= \eta \sum_{n=1}^{N} \exp\left((N-n)\log(1-\eta)\right) x_n \\
&\approx \eta \sum_{n=1}^{N} \exp\left(-(N-n)\eta\right) x_n \\
&\approx \eta \sum_{q=1}^{N} \exp(-q\eta) x_{N-q} \\
&\approx \frac{1}{W} \sum_{q=1}^{W} x_{N-q}
\end{aligned}
$$

Compare dynamic estimators in non-stationary data