# Introduction to Spatio-Temporal Pattern Recognition

Jayson Rome

April 26, 2002

**Abstract**

In many pattern recognition tasks we may be interested in asking questions about the behavior of sequences of observations on a particular random vector. Given that we have observed the sequence of feature vectors $x_1, x_2, ..., x_i$, at $i$ discrete points in time, for example, how likely is it that $x$ will take the value $x_{i+1}$ at time $i+1$. We may be interested in asking questions about the behavior of sequences of observations on a particular random vector. Given that I have observed the sequence of feature vectors $x_1, x_2, ..., x_i$, at $i$ discrete points in time, how likely is it that $x$ will take the value $x_{i+1}$ at time $i+1$.

# Contents

**1 Introduction** **3**

**2 Mathematical Foundations** **4**
2.1 Bayesian Classifier . . . . . 4

**3 Markov Chains: the 1D Markov Property** **5**
3.1 Markov Matrices and Random Walks on Graphs . . . . . 5

**4 Hidden Markov Models** **7**
4.1 Evaluation Problem . . . . . 8
4.2 State Estimation Problem . . . . . 9
4.3 Training Problem . . . . . 9
4.4 Example . . . . . 10
4.5 Applications of HMMs . . . . . 11

**5 Markov Random Fields: the 2D Markov Property** **11**
5.1 Applications of MRFs . . . . . 13

**6 Conclusions** **13**

# 1 Introduction

The recognition task can be described as follows: In general some real world stochastic process (human speech, photons, etc) produces some observable output, which can be characterized by a signal (1D as in the case of speech, or multidimensional as in the case of 2D and 3D imaging). These signals may be discrete or continuous, stationary or non-stationary, pure or corrupted by some other signal, noise or a carrier wave for example [17], [16]. Measurable properties of the signal, which directly or indirectly indicate some information about the state of the signal, are selected and orthogonal features are used to construct a feature vector. It is hoped that this vector will uniquely distinguish a single signal (pattern, object, system) by representing each unique object as a unique point in some feature space. Due to the randomness inherent in measurement the feature vectors must be treated as random variables and analyzed probabilistically [20]. To classify and unknown object some measure is used to compute the "distance" of the unknown vector to vectors representative of known classes of objects. The unknown signal is assigned to the class whose vector is "closest" or "likeliest". Given a classification task involving $M$ classes

$$\omega_1, \omega_2, \ldots, \omega_M$$

and some unknown object with feature vector $x$, we use Bayes Theorem to compute the $M$ conditional probabilities

$$P(\omega_i|x)$$

(i.e. the probability of class $\omega_i$ , given observation $x$) for $i = 1, 2, \ldots, M$ and assign $x$ to the class with the highest probability [20], [5].

We may be interested in asking questions about the behavior of sequences of observations on a particular random vector. Given that I have observed the sequence of feature vectors $x_1, x_2, ..., x_i$, at $i$ discrete points in time, how likely is it that $x$ will take the value $x_{i+1}$ at time $i + 1$. C

Hidden Markov Models (HMMs) are doubly stochastic processes based on finite state machines, in which an observable random variable (symbol) is dependant on an underlying hidden random variable (state). HMMs have proved extremely important and effective in the analysis of one-dimensional stochastic signals. In speech recognition (where HMMs have enjoyed great success and technical refinement) an unknown speech signal is characterized by a series of quantized vectors. Each known word is modelled as a finite hidden state machine capable of emitting some observable symbol during state transitions. For each of the models in the dictionary, the probability that the model generated the unknown sequence is computed. The recognition task involves assigning the unknown signal to the model with the highest probability. The HMM is uniquely suited to model 1D signals. We think of the model as being a sequence generator. In any given state the model has some probability of generating one symbol from a given alphabet. Models can be combined into hierarchies organized around some grammar. In this way HMMs can be extended from recognizing individual words to recognizing sentences. The model does not extend itself very well to multi-dimensional signals so the traditional approach has been to find some 1D representation of the multidimensional signal and recognize the 1D signal with traditional HMM methods.

The 2D analogue of the Markov Chain is the Markov Random Field (MRF). MRFs have been applied to image segmentation and restoration, texture segmentation, tumor identification and low-level vision. The MRF is equivalent to the Gibb's distribution, which is mathematically more tractable.

## 2 Mathematical Foundations

*Bayes Theorem,* a fundamental tool in the analysis of stochastic systems, tells us:

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)},$$

where

$$P(x) = \sum_{i=1}^{M} P(x|\omega_i)P(\omega_i).$$

In words

$$A\ Posteriori = \frac{(Likelihood)(A\ \Pr iori)}{(Evidennce)}.$$

### 2.1 Bayesian Classifier

The Bayes Classifier can be shown to be optimal in the sense of minimizing the classification error probability [5]. To classify a given observation vector $x$, independently of prior or future observations, we proceed as follows: Given: $M$ classes $\omega_1, \omega_2, \ldots, \omega_M$ and an unknown pattern with feature vector $x$, assume

$$P(\omega_1), P(\omega_2), \ldots, P(\omega_M)$$

are known,

$$P(x|\omega_1), P(x|\omega_2), \ldots, P(x|\omega_M)$$

are known and find the $M$ conditional probabilities $P(\omega_i|x)$ for $i = 1, 2, \ldots, M$. Assign $x$ to class with the Maximum A Posteriori probability.

We can extend the model to consider the context in which an observation is made by classifying a sequence of observations as follows:
Let:

$$X : x_1, x_2, \ldots, x_N,$$

be a sequence of $N$ observed feature vectors

$$\omega_i : i = 1, 2, \ldots, M$$

be the M possible classes

$$\Omega_i : \omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_N},$$

be one possible sequence of classes corresponding to the observation sequence with $i \in \{1, 2, \ldots M\}$ for $k = 1, 2, \ldots N$

Find: the class sequence $\Omega_i$ that a sequence of observations $X$ belongs to. Applying Bayes theorem again gives [20]:

$$P(\Omega_i|X) = \frac{P(X|\Omega_i)P(\Omega_i)}{P(X)}$$

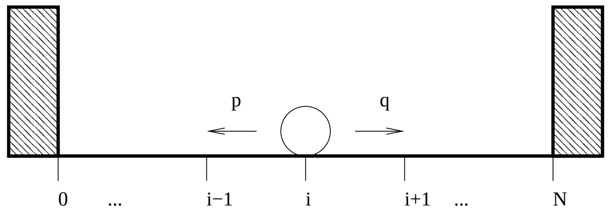


Figure 1: A finite one dimensional walk on the integer lattice. The boundaries of the walk may be absorbing, reflective or some combination of both.

# 3 Markov Chains: the 1D Markov Property

*Markov property* tells us that the state of some random variable depends only on the previous state, and is independent of all other past states, i.e. that the state of the system at time $i$ is dependent only on the state of the system at time $i - 1$.

$$P(F_i = f_i | F_k = f_k \forall k \neq i) = P(F_i = f_i | F_{i-1} = f_{i-1})$$

Assuming the Markov property allows us to model a wide class of stochastic processes, including random walks, birth and death processes, success runs and urn models as Markov chains [6].

## 3.1 Markov Matrices and Random Walks on Graphs

The Markov chains can be formulated in terms of finite state machines or equivalently as a state vector and associated matrix operator. A stochastic matrix is a square matrix with non-negative entries and rows that sum to one. A Markov process can be represented by a stochastic matrix $M$, and the state can be represented by a column vector $x$. The state vector can be updated by:

$$y = Mx$$

A Markov matrix $M$ has the following properties: $\lambda_1 = 1$ is an eigenvalue of $M$, its associated eigenvector $x_1$ is nonnegative, all other eigenvalues satisfy $|\lambda_i| \leq 1$, and if any multiple $M^k$ has all positive entries, then $|\lambda_i| < 1$ [19]. These properties allow us to apply power iteration to well behaved Markov matrices to find the principle eigenvector, which is the eigenvector with eigenvalue $\lambda_1 = 1$, and corresponds to a *steady state distribution.* The assumption of the Markov property greatly simplifies our analysis of some very complicated stochastic processes. Consider a simple stochastic process like a one-dimensional random walk illustrated in figure 1. At regular time intervals the particle moves $+1$ unit with probability $p_i$ or $-1$ unit with probability $q_i$ along the one-dimensional integer lattice.

Consider a finite segment of the lattice beginning at the origin, which has length $N$. Conceptually it might be easier to model the system as some finite N-state machine. Each state represents a position on the walk. Individual states $s$ can be considered absorbing if a particle does not leave the state $s$ after entering it, reflecting if the particle immediately returns to the previous state after entering $s$, or partially absorbing / partially reflecting, if there is some probability of being absorbed or reflected by $s$. At regular time intervals the machine is able to make random transitions to adjacent states according to some probability distribution. If the machine is in state $i$ at time $t$, it moves to state $i + 1$ with probability $p_i$ and moves to state $i - 1$ with probability $q_i$.

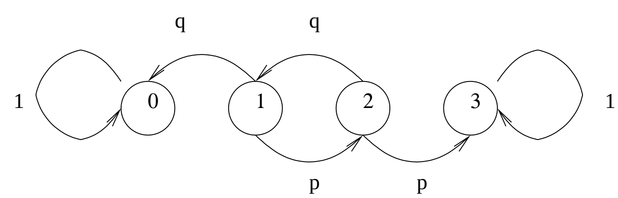


Figure 2: A finite state machine representing a one dimensional random walk with absorbing boundaries.

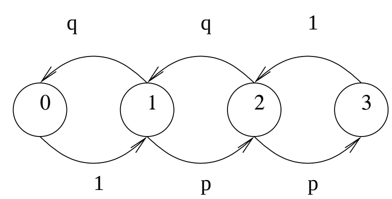


Figure 3: A finite state machine representing a one dimensional random walk with reflecting boundaries.

If the particle moves to position $(-1)$ or $(N+1)$ (the machine moves into state $(-1)$ or state $N$, it is said to have been absorbed or to have fallen off in the case of absorbing boundaries, or in the case of reflecting boundaries is returned to its previous state. Figures 2, 3, 4 show finite state machine representations of $1-D$ random walks with various boundary conditions.

If we know the probability of moving from one state to the next and we assume the Markov Property holds, we can create a state transition matrix and define a vector which represents the probability of finding the particle in any of the given states after some number of random moves. Let $y^T = (y_1, y_2, \ldots, y_N)$ be the *Current State Distribution*, and let $x^T = (x_1, x_2, \ldots, x_N)$ be the *Initial State Distribution*. Let

$$M = \begin{bmatrix} m_{11} & \cdots & m_{1N} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NN} \end{bmatrix}$$

be the *State Transition Matrix*, where $m_{ij}$ is the *State Transition Probability* of going from

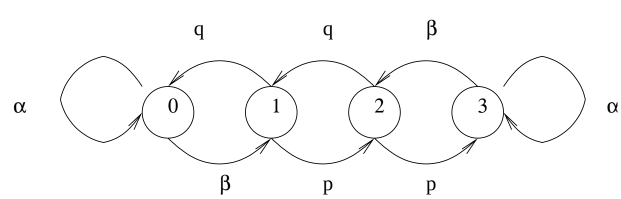


Figure 4: A finite state machine representing a one dimensional random walk with mixed boundaries.

state $i$ to state $j$, and $\sum_{j=1}^{N} m_{ij} = 1, \forall i$. Let $y = Mx$ be the state after one random transition and $y = M^k x$ be the state after $k$ random transitions. Given this model of the system we can simulate the system and ask questions about its behavior. To find the *steady state distribution* of the particle, for example, we would find the eigenvalues of the equation $Mx = \lambda x$ , where $\lambda = 1$ is the principle eigenvalue and hence the steady state distribution. For example with $N = 5$, $p_i = q_i = \frac{1}{2}$, and initial distribution

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

and state transition matrix

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ q_2 & 0 & p_2 & 0 & 0 \\ 0 & q_3 & 0 & p_3 & 0 \\ 0 & 0 & q_4 & 0 & p_4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

we find

$$x_\infty = \begin{bmatrix} \frac{1}{2} \\ 0 \\ 0 \\ 0 \\ \frac{1}{2} \end{bmatrix}$$

We see that in the steady state (i.e. as $t \to \infty$ ) that the particle will be absorbed, and will be found at either end with equal probability. We could again use this model to answer questions about the probabilistic behavior of the system. To find out how likely it was that a given sequence was generated by a given model, for example, we merely compute:

$$P(o_0 o_1 o_2 \ldots o_M) = \prod_{i=0}^{M} r_i$$

where

$$r_i = \begin{cases} p_i & if \quad F_i(n) = h \\ q_i & if \quad F_i(n) = t \end{cases}$$

and $F_i(n)$ is the result of flipping i$^{th}$ coin at the n$^{th}$ time step.

# 4 Hidden Markov Models

We can improve on the usefulness of the Markov Chain model by introducing a second observable stochastic process: the ability for the machine to emit some symbol dependant on the state of the machine. For the Hidden Markov Model we assume that the state of the machine is not directly observable. On the transition from one hidden state to the next the model emits some observable symbol from a finite alphabet according to some probability distribution. Following Rabiner [17], [16], we define a HMM by:

- $N$: number of states.

  The individual states are given by $s = \{s_1, s_2, \dots, s_N\}$

  The state at time $t$ is given by $q_t$

- M: Number of distinct observation symbols per state

  The individual symbols are given by $v = \{v_1, v_2, \dots, v_M\}$

- $A = \{a_{ij}\}$: State transition probability matrix

  $a_{ij} = P(q_{t+1} = s_j | q_t = s_i) : 1 \le i, j \le N$

- $B = \{b_j(k)\}$ : Observation symbol probability distribution in state j

  $b_j(k) = P(v_k : at \quad t | q_t = s_j) : 1 \le j \le N, \quad 1 \le k \le M$

- $\Pi = \{\pi_i\}$ : Initial state distribution

  $\pi_i = P(q_1 = s_i) \quad : 1 \le i \le N$

A model can be expressed in shorthand notation by $\lambda = (A, B, \Pi)$. The following algorithm generates a sequence of observations from a given model:

1. Choose initial state from $\Pi$,
2. Choose observed symbol from $b_j(k)$,
3. Move to new state according to $a_{ij}$.

The three basic problems for HMM as outlined by Rabiner [17], [16]:

1. Evaluation Problem,
2. State Estimation Problem,
3. Training Problem.

## 4.1 Evaluation Problem

Given an observed sequence

$$O = \{o_1, o_2, \dots, o_T\},$$

and a model

$$\lambda = (A, B, \Pi),$$

we would like to find

$$P(O|\lambda),$$

the probability that the observed sequence $O$ was generated by the model $\lambda$. This problem is solved with the Forward-Backward Algorithm. We begin by defining the forward variable

$$\alpha_t(i) = P\left(O_1, O_2, \dots, O_t, i_t = q_i | \lambda\right),$$

with

$$\alpha_1(i) = \pi_i b_i\left(O_1\right), \quad 1 \le i \le N.$$

Having defined the forward variable, we perform the following iteration:
for t = 1 to Tmax

$$\alpha_{t+1}(i) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}).$$

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_{T\max}(i).$$

Similarly we define the backward variable

$$\beta_{T\max}(i) = 1,$$

and perform the following iteration:
for t =Tmax . . . 1

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

## 4.2 State Estimation Problem

Given an observed sequence

$$O = \left\{o_1, o_2, \ldots, o_T\right\},$$

and a model

$$\lambda = (A, B, \Pi),$$

we would like to find the Optimal state sequence

$$Q = \left\{q_1, q_2, \ldots, q_T\right\}$$

to "explain" the observation. We define

$$\gamma_t(i) = P(i_t = q_i|O, \lambda),$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}$$

$$i_t = arg \max_{1 \le i \le N} \left[\gamma_t(i)\right], \quad 1 \le t \le T\max$$

Also Solved with Viterbi Algorithm.

## 4.3 Training Problem

Given a training sequence

$$O = \left\{o_1, o_2, \ldots, o_T\right\}$$

we would like to adjust the model parameters

$$\lambda = (A, B, \Pi)$$

to maximize

$$P(O|\lambda),$$

the probability of the observation $O$, given the model $\lambda$. Solved with Baum-Welch Re-estimation Algorithm [17], [16].

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)}$$

$$\bar{\pi} = \gamma_1(i), \quad 1 \leq i \leq N$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T\max -1} \xi_t(i,j)}{\sum_{t=1}^{T\max -1} \gamma_t(i)}$$

$$\bar{b}_j(k) \frac{\sum_{t=1, O_t=k}^{T\max} \gamma_t(j)}{\sum_{t=1}^{T\max} \gamma_t(j)}$$

## 4.4 Example

For example, given a model with Observation symbols

$$v = \{start, stop, A, B\};$$

Observation symbol density

$$B = \left\{ \begin{array}{ccc} p(A) = p_1 & p(A) = p_2 & p(A) = p_3 \\ p(B) = q_1 & p(B) = q_2 & p(B) = q_3 \end{array} \right\};$$

State transition matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & S_1 & R_1 & 0 & 0 \\ 0 & 0 & S_2 & R_2 & 0 \\ 0 & 0 & 0 & S_3 & R_3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

And Initial state distribution

$$\Pi = \{1, 0, 0, 0, 0\}$$

We can compute the probability of a given state sequence given the observed sequence

$$startABBAstop.$$

We enumerate all possible state sequences that could have generated the observation:

$$p(0,1,1,2,3,4|startABBAstop) = S_1p_1S_1q_1R_1q_2R_2p_3R_3,$$

$$p(0,1,2,2,3,4|startABBAstop) = R_1p_2S_2q_2S_2q_2R_2p_3R_3,$$

$$p(0,1,2,3,3,4|startABBAstop) = R_1p_2R_2q_3S_3q_3S_3p_3R_3,$$

as well as computing the probability of generating the sequence $startABBAstop$ , given the model parameters:

$$p(startABBAstop|\lambda) = S_1p_1S_1q_1R_1q_2R_2p_3R_3 + R_1p_2S_2q_2S_2q_2R_2p_3R_3 + R_1p_2R_2q_3S_3q_3S_3p_3R_3$$

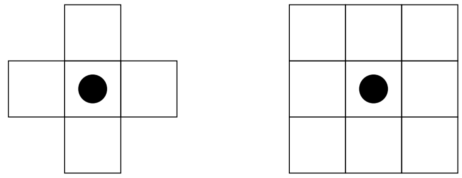

Figure 5: $\eta^1$ and $\eta^2$ neighborhood systems

### 4.5 Applications of HMMs

- Speech Recognition: [17], [16].
- 2D Shape Recognition: [10]
- Spatio-Temporal Pattern Recognition: [7]
- Color Image Retrieval: [12]
- Lip Reading: [14]
- Gesture Recognition [21]
- Sign Language Recognition [18]
- 2D Texture Recognition and Restoration [3]

## 5 Markov Random Fields: the 2D Markov Property

The 2D Markov property tells us that the state of some 2D probability distribution at point $(i,j)$ is dependant only on the state of the distribution in a neighborhood specified around point $(i,j)$.

$$P(F_{ij} = f_{ij} | F_{kl} = f_{kl} : (k,l) \neq (i,j)) = P(F_{ij} = f_{ij} | F_{kl} = f_{kl}, (k,l) \in \Im_{ij}),$$

where $\Im_{ij}$ is some neighborhood of $(i,j)$. In the Markov Random Field model (MRF) we assume that the image is a sample from some random field with correlations among neighboring pixels. We represent the intensity at each pixel as a linear combination of pixel intensities in a neighborhood with added noise [15]. The Markov random field is equivalent to the Gibb's Distribution. Following Geman and Geman's seminal work [9], we are given the $M \times M$ integer lattice

$$Z_M = \{(i,j) : 1 \leq i,j \leq M\},$$

define an image to be a two dimensional array of random variables:

$$\{F = f\} \equiv \{F_{ij} = f_{ij}, (i,j) \in Z_M\}$$

we define the original image as a two dimensional array composed of two processes: $X = (F, L)$ , is the actual image, where $F = \{f_{ij}\}$ is the Intensity process that represents the observable pixel intensities and $L = \{l_{ij}\}$ is the line process and is made up of unobservable edge elements.

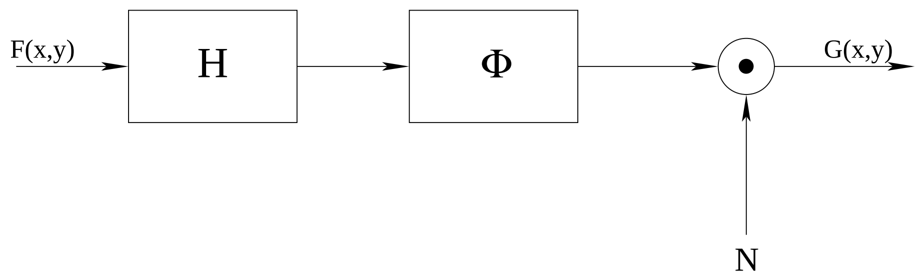


Figure 6: Mathematical model of image formation with degradation.

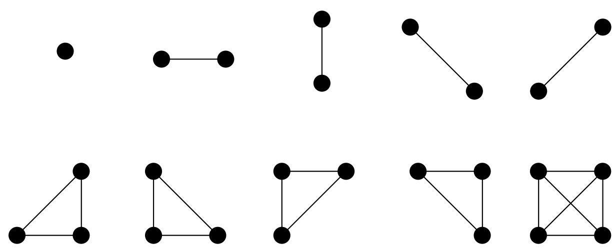

Figure 7: $\eta^2$ cliques

$F = \{f_{ij}\}$ is a sample realization of a usually isotropic and homogeneous random field with correlations beyond nearest neighbors. The degraded image is modelled as a blur, some non-linear transformation and some noise process: $G = \Phi(H(F)) \circ N$ where $H$ is a blurring matrix represented by a shift invariant point spread function; $\Phi$ is a Nonlinear transform; $N$ is an independent Noise field, $\circ$ is some invertible operation (like addition or multiplication).

Model $F = \{f_{ij}\}$ is a MRF, which is equivalent to a Gibb's distribution (GD) by the Hammersley-Clifford theorem: The neighborhood systems $\eta^1, \eta^2, \ldots, \eta^m$ are collections of subsets. In a *clique* every pair of distinct sites are neighbors. The $\eta^2$ clique consists of ten sets of neighbors:

A random field $X = \{x_{ij}\}$ defined on the integer lattice is a Gibb's Random Field with respect to $\eta$ if and only if its joint distribution is given by:

$$P(X_{ij} = x) = \frac{1}{Z} e^{-U(x)} = \frac{1}{Z} e^{\frac{-U'(x)}{T}},$$

where

$$U(x) = \sum\nolimits_{c \in C} V_c(x),$$

is an energy function over clique C,

$$V_c(x),$$

is a potential associated with clique C,

$$Z = \sum\nolimits_{x} e^{-U(x)},$$

is a partition function, and $T$ is a global temperature.

Choice of potential functions can give binary, binomial, Poisson, and Gaussian Random Fields. To define a GD it is necessary to specify some neighborhood system, to define cliques and to specify clique potentials. Given some degraded sample G we wish to estimate the original image X by again utilizing Bayes theorem:

$$P(X = \omega : \omega = (f, l)|G = g) = \frac{P(G = g|X = \omega)P(X = \omega)}{P(G = g)}$$

Assuming X has a Gibbs distribution

$$P(X = \omega) = \frac{1}{Z} e^{\frac{-U(\omega)}{T}}$$

$$P(G = g|X = \omega)$$

is also Gibbsian:

$$P(G = g|X = \omega) = P(\Phi(H(F)) \circ N = g|X = \omega) = \frac{1}{Z} e^{\frac{-U^P(x)}{T}}$$

The restoration (estimate of $X$) is found by a process of simulated annealing, in which the temperature $T$ is changed according to some cooling schedule $T(k) = \frac{c}{\log(1+k)}$ , which gives the temperature at the $k^{th}$ iteration. The process generates a sequence of images that converges to an estimate of the Maximum A Priori (MAP) probability of the original image given the degraded sample [9].

### 5.1 Applications of MRFs

- Segmentation and Restoration: [9], [1], [2]
- Texture Models [4]
- Tumor Detection [11]
- Color Image Segmentation [13]
- Color Texture Segmentation [15]
- Low Level Vision [8]

## 6 Conclusions

Temporal pattern recognition is concerned with changes in a signal over time. The Markov assumption allows us to model a wide variety of stochastic processes. Markov models provide a framework for modelling the time dependent behavior of random vectors.

## References


[1] J. Besag, *Spatial Interaction and the Statistical Analysis of Lattice Systems*, Journal of the Royal Statistical Society, Series B **36** (1974).

[2] ———, *On the Statistical Analysis of Dirty Pictures*, Journal of the Royal Statistical Society, Series B **48** (1986).

[3] V. Breuer and G. Radons, *Classification of time series with hidden Markov models: Unsupervised learning and self-organization*, Physical Review E **53** (1996), no. 4.

[4] G. R. Cross and A.K. Jain, *Markov Random Field Texture Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-5** (1983), no. 1.

[5] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[6] W. Feller, *An Introduction to Probability and Its Applications*, third ed., John Wiley and Sons, New York, 1971.

[7] K. H. Fielding and D. W. Ruck, *Spatio-Temporal Pattern Recognition Using Hidden Markov Models*, IEEE Transactions on Aerospace and Electronic Systems **31** (1995), no. 4.

[8] W. T. Freeman and E. C. Pasztor, *Learning low-level vision*, IEEE International Conference On Computer Vision, 1999.

[9] D. Geman and S. Geman, *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-6** (1984).

[10] Y. He and A. Kundu, *2D Shape Classification Using Hidden Markov Model*, IEEE Transactions on Pattern Analysis and Machine Intelligence **13** (1991), no. 11, 1172–1184.

[11] H. D. Li, M. Kallegri, L. P. Clarke, V. K. Jain, and R. A. Clark, *Markov Random Field for Tumor Detection in Digital Mammography*, IEEE Transactions on Medical Imaging **14** (1995), no. 3.

[12] H. Lin, L. Wang, and S. Yang, *Color Image Retrieval Based on Hidden Markov Models*, IEEE Transactions On Image Processing **6** (1997), no. 2.

[13] J. Liu and Y. Yang, *Multiresolution Color Image Segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994), no. 7.

[14] J. Luettin, N. A. Thacker, and S. W. Beet, *Visual Speech Recognition Using Active Shape Models and Hidden Markov Models*, IEEE ICASSP (1996).

[15] D. K. Panjwani and G. Healey, *Markov Random Field Models for Unsupervised Segmentation of Textured Color Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995), no. 10.

[16] L. R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE **77** (1989), no. 2.

[17] L. R. Rabiner and B. Juang, *An Introduction to Hidden Markov Models*, IEEE Acoustics, Speech and Signal Processing (1986).

[18] T. Starner and A. Pentland, *Real-Time American Sign Language Recognition From Video Using Hidden Markov Models*, ISCV95, 1995.

[19] G. Strang, *Linear Algebra and its Applications, Third Edition*, Harcourt Brace Jovanovich, Orlando, Florida, 1986.

[20] S. Theodoridis and K. Koutroumbas, *Pattern recogniton*, Academic Press, San Diego, 1999.

[21] A.D. Wilson and A. F. Bobick, *Parametric Hidden Markov Models for Gesture Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999), no. 9.