Distributed Fusion in Sensor Networks [Çetin et al. 06]

A Graphical Models Perspective

January 21st 2008

Overview

- Introduction
  - Challenges in sensor networks
  - Graphical models? What for?
- Graphical models
  - Definitions
  - Markov property
  - Factorization
  - Sum Product Algorithm
  - Other algorithms
- Mapping a Network Application to a Graphical Model
  - Self localization in Sensor Networks
- Message Censoring
- Trading off Accuracy for Bits in Particle-Based Messaging
- Effects of Message Approximation
- Summary
- Outlook

Introduction – Challenges

Typical Challenges in (Wireless) Sensor Networks

- Distributed nature of computation and deployment
- Communication bandwidth constraints
- Energy constraints
- Information sharing involves approximation
- Traditional measures of distortion not sufficient to describe quality of approximation

Introduction – Graphical models? What for?

- Well suited to capture structure of sensor network which consists of
  - Nodes (for sensing, communication and computation)
  - Connections between nodes (for modelling statistical dependencies and communication links)
- Well developed inference algorithms on graphs already exist
  - Scalable
  - Can deal with “loopy graphs”
  - good convergence and accuracy properties
- Inference Algorithms use parallel message-passing operations
  - well suited to parallel realization of sensor networks via physically distributed processors
- Provide suitable framework for development and analysis of communication-constrained versions of message-passing algorithms

Graphical Models – Definitions

- GM defined on (undirected) graph $G(V,E)$ consisting of
  - Vertex or node set V and
  - Edge set $E \subset V \times V$
- Each node $v \in V$ is associated with a random variable or random vector X_v
- Set of edges E describe conditional dependencies that exist between nodes
- set of random variables/vectors $X = \{X_v : v \in V\}$ has to satisfy Markov property with respect to G

Graphical Models – Conditionally Independence

Definition of conditionally independence:

- two variables X_A and X_B are conditionally independent given a third variable X_C if:

  $P(X_A,X_B| X_C) = P(X_A|X_C)\, P(X_B| X_C)$ … for every X_C

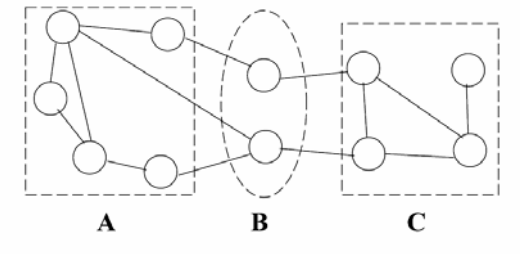
- which is equivalent to saying

  $P(X_A|X_B,X_C) = P(X_A| X_C)$ … for every X_C

Graphical Models – Markov Property

- Assuming V partitioned in disjoint sets A, B and C, in which B separates A and C (Every path between sets A and C must pass through set B)
- The sets of variables $X_A = \{X_v : v \in A\}$ and $X_C = \{X_v : v \in C\}$ must be conditionally independent given the values of $X_B = \{X_v : v \in B\}$
- Thus, the distribution $p(X_A, X_B, X_C)$ can be written in the form:

  $p(X_B)\, p(X_A|X_B)\, p(X_C|X_B)$

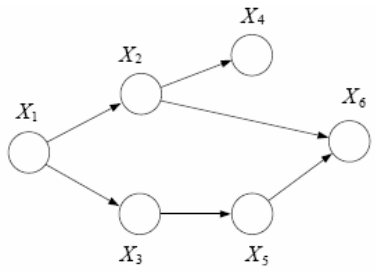


[Ihler et al. 05]

- For a “graph” associated to time series we could say: the past and the future are conditionally independent given the present

Graphical Models

- Graphical models represent large joint distributions compactly using a set of “local" relationships specified by a graph
- Macro language for description particular family of joint distributions
- Edges between the nodes tell us *qualitatively* about the factorisations of the joint probability
- Functions, stored at the nodes tell us *quantitative* details of the pieces into which the joint distribution factors



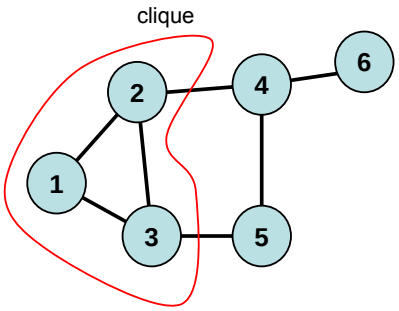
[Roweis 06]

Graphical Models – Factorization (1)

- As long G is relatively sparse, performing factorisation is an efficient method for calculating joint distributions of a large number of random variables

---

- Let $\mathcal{C}$ denote the set of all cliques on G
- A clique C is a subset of nodes out of V that are fully connected



Graphical Models – Factorization (2)

- If a random vector X is Markov with respect to G its distribution p(x) admits factorisation as product of functions ψ of variables restricted to cliques $C \in \mathcal{C}$
- p(x) has to be strictly positive

$$p(x) = \frac{\prod_{C \in \mathcal{C}} \psi_C(x_C)}{Z} \; ; \; Z \triangleq \sum_x \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

- Z is the partition Function
- $\psi_C(x_C)$ are the so-called compatibility functions
- if log is applied to this functions, they are referred to as potentials or potential functions

Graphical Models – Factorization (3)

- Assumption for simplicity: a potential is a function either of
  - variable at single node of graph (node potential) or
  - variables at pair of nodes corresponding to an edge in E (edge potential)

$$p(x) = \frac{\left(\prod_{s \in V} \psi_s(x_s)\right) \left(\prod_{(s,t) \in E} \psi_{s,t}(x_s, x_t)\right)}{Z}$$

- Assumption causes no real loss of generality
- Graphical models with higher-order potential functions may always be converted to models with pair wise potential functions (via variable augmentation)

Graphical Models – Joint Distributions

- Computation of marginal distributions relatively straight forward for graphs without loops
- Node and pair-wise potentials can be expressed in terms of
  - marginal probabilities at nodes: $\psi_s(x_s) = p_s(x_s)$ or $\psi_s(x_s) = p_s(x_s)\, p(y_s|x_s)$ if there is a measurement y_s associated with x_s
  - Joint probabilities of pairs of nodes connected by edges: $\psi(x_s,x_t) = p_s(x_s,x_t) / p(x_s)\, p(x_t)$

$$p(x) = \prod_{s \in V} p_s(x_s) \prod_{(s,t) \in E} \frac{p_{st}(x_s, x_t)}{p_s(x_s) p_t(x_t)}$$

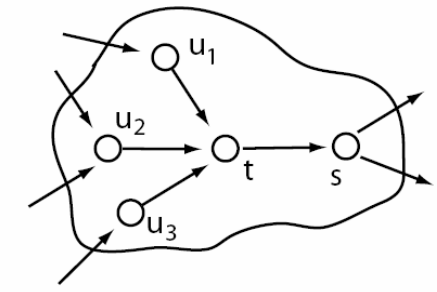
- Marginal probabilities (at each node) can be calculated efficiently by so-called belief propagation algorithms

Graphical Models – Belief Propagation (1)

- Synonym: sum-product algorithm [Ihler et al. 05]
- popular method for solving inference problems on arbitrary graphical models exactly or approximately
  - Tree-structured GMs: optimal (exact) results
  - Loopy graphs: approximate results, may not converge
- Approximate nature of loopy belief propagation acceptable price for performing efficient inference
- Often even additional approximations are performed because
  - Exact message representation is computationally intractable
  - Finite parameterisation for messages is needed
  - Reducing size of messages to decrease their representational costs (energy and bandwidth constraints)
  - …

Graphical Models – Belief Propagation (2)

- Goal: compute marginal distribution $p(x_t)$ at each node t
- Via message-passing algorithm between nodes
- Message is expressed in terms of updating outgoing message at iteration i from each node t to each neighbour s in terms of previous $(i-1)$ iteration's incoming messages from t's neighbours Γ_t



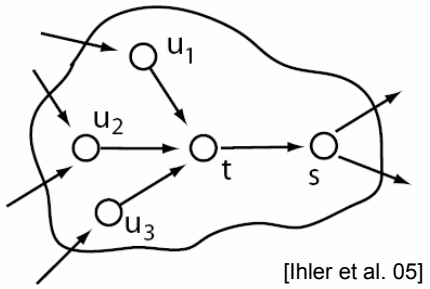
[Ihler et al. 05]

Graphical Models – Belief Propagation (3)

- Message $m_{t,s}$ sent from node t to subsequent node s

$$m_{ts}^{i}(x_s) \propto \int \psi_{ts}(x_t, x_s)\psi_t(x_t) \prod_{u \in \Gamma_t \setminus s} m_{ut}^{i-1}(x_t)dx_t$$

  - Γ_t … Neighborhood nodes of t
  - Each message is normalized so as to integrate to unity



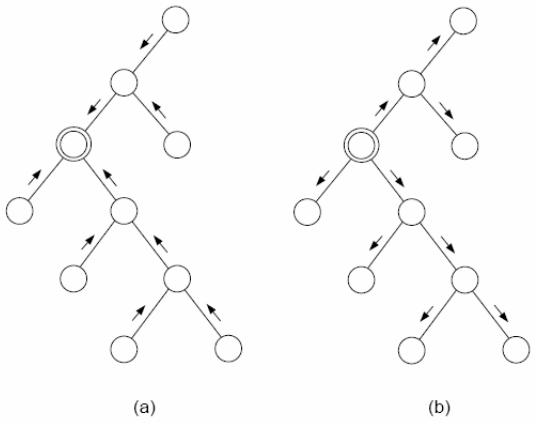
[Ihler et al. 05]

Graphical Models – Belief Propagation (4)

- Marginal distribution at any node is calculated by

$$M_t^i(x_t) \propto \psi_t(x_t) \prod_{u \in \Gamma_t} m_{ut}^i(x_t).$$

- Belief propagation algorithm:
  - Choose an arbitrary root node
  - Calculate messages and pass them from leaves up to root (a) and then back down (b)
  - Given all messages, compute marginals by upper formula



[Roweis 06]

Other methods

- Max-product algorithm
  - alternative factorisation of p(x) in terms of max-marginals

$$p(x) \propto \prod_{s \in V} q_s(x_s) \prod_{(s,t) \in E} \frac{q_{st}(x_s, x_t)}{q_s(x_s) q_t(x_t)}$$

$$q_s(x_s) := \max_{x_u, u \in V \setminus s} p(x_1, \ldots, x_n)$$

$$q_{st}(x_s, x_t) := \max_{x_u, u \in V \setminus \{s,t\}} p(x_1, \ldots, x_n)$$

  - For loop free graphs

- Tree-reweighted max-product algorithm (TRMP)
- Nonparametric belief propagation (NBP)

Mapping Network Applications to GM (1)

Self localisation in sensor networks

- Well-recognized problem for many sensor network applications
- We consider case in which available information for sensor location consists of
  - Uncertain prior information about location of a sensor subset (e.g. GPS data)
  - Ability of sensors to communicate with each other and measure their intersensor distance
  - Any distance measurements obtained

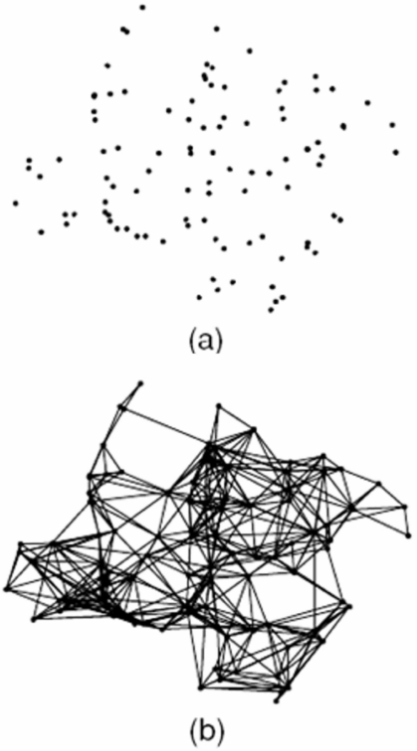
Mapping Network Applications to GM (2)

Sensor localisation

a) Physical location of sensors

b) Graphical Model:

- Nodes correspond to sensor node location variables
- Edges correspond to observed information (intersensor distance measurements) from a subset of node pairs



Mapping Network Applications to GM (3)

Definitions:

- $p_s(x_s)$ … prior location probability distribution
- $Pr(x_s,x_t)$ … probability of obtaining a distance measurement between two sensors s and t (located at x_s and x_t)
- $p_L(l_{st}|x_s,x_t)$ … probability distribution of measuring a distance l_{st} (given, that the true sensor positions are x_s and x_t)

Mapping Network Applications to GM (4)

- Calculating potentials

$$\psi_s(x_s) = \rho_s(x_s)$$

$$\psi_{st}(x_s, x_t) = \begin{cases} Pr(x_s,\ x_t)\,\rho_L(l_{st}); & l_{st} \text{ is observed} \\ 1 - Pr(x_s,\ x_t); & \text{otherwise.} \end{cases}$$

- Finally we can apply the belief propagation algorithm to this network to gather marginal distributions of all nodes
- Marginals refer to inferenced sensor position

Overview

Message Censoring (1)

Self localization in sensor networks

Multiobject data association

Different approaches

- Sum-product algorithm (Belief-propagation)
- Max-product
- TRMP algorithm

For all these messages

- Message transmitted over whole network

Message Censoring (2)

Message schedule

- Creation, transmission, processing
- Convergence behaviour
  - Centralized coordination
  - Local rules

Simple local rule

- Compute Kullback-Leibler divergence (KLD)

$$D\left(M_{ts}^{k} \| M_{ts}^{k-1}\right) = \sum_{x_s} M_{ts}^{k}(x_s) \log \frac{M_{ts}^{k}(x_s)}{M_{ts}^{k-1}(x_s)}$$

Message censoring (3)

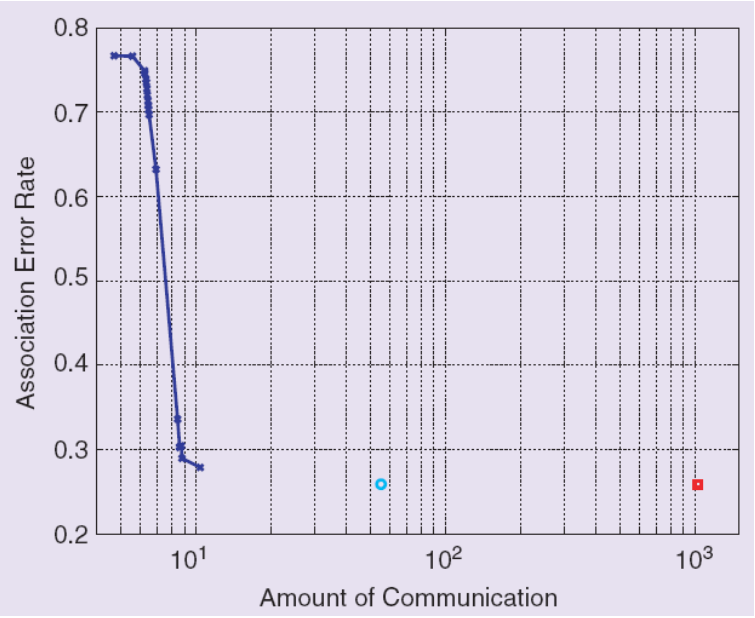
Kullback-Leibler divergence

- Completely local
- Data dependent message
- Nodes can become silent
- Restart sending when sufficiently new information reaches from “elsewhere”

Message censoring (4)

Modified sum-product algorithm

- Tracking targets
- Varying threshold
- Number of wrong matches / number of measurements
- Compared to
  - Max-product (cyan)
  - TRMP (blue)



Performance-communication tradeoff

Message censoring (5)

Side-effect

- Better performance with message censoring (?)
- “Rumor propagation” of algorithm
  - Repeated propagation of messages in loops
  - Leads to incorrect corroboration of hypothesis

Trading off accuracy for Bits (1)

Particle based representations of messages

- Efficient transmission for such messages
- How many particles do we need
- How accurate a representation is required

Trading off accuracy for Bits (2)

The task is to transmit a probability distribution q(x) represented by a set of particles $\{x_i\}$

- Standard problem in transmission
- Big difference: Neither order nor the accuracy is important for individual particles
- Only the accuracy of the reconstruction of q(x) is important
- Transmission protocol can order the particles and encode only differences $x_{i+1} - x_i$

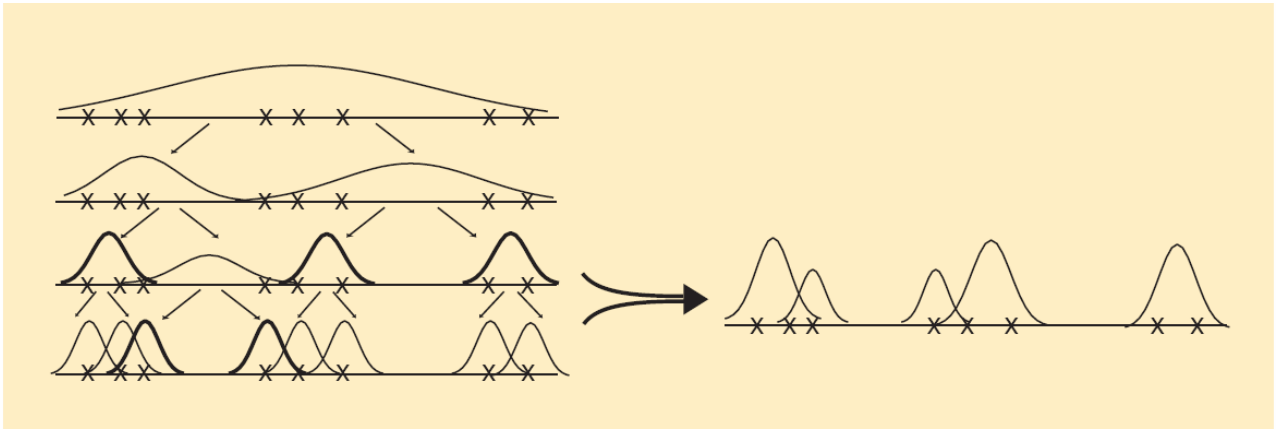
Trading off accuracy for Bits (3)

KD-tree

- The idea is to group particles in a binary tree structure
- Starting from a root node where all particles are grouped together
- Split the set and refine the clustering
- Make an easy approximation on each node of the tree such as Gaussian
- Every cut through this tree corresponds to an approximation of the full, fine scale distribution q(x)

Trading off accuracy for Bits (4)

KD-tree conceptional illustration



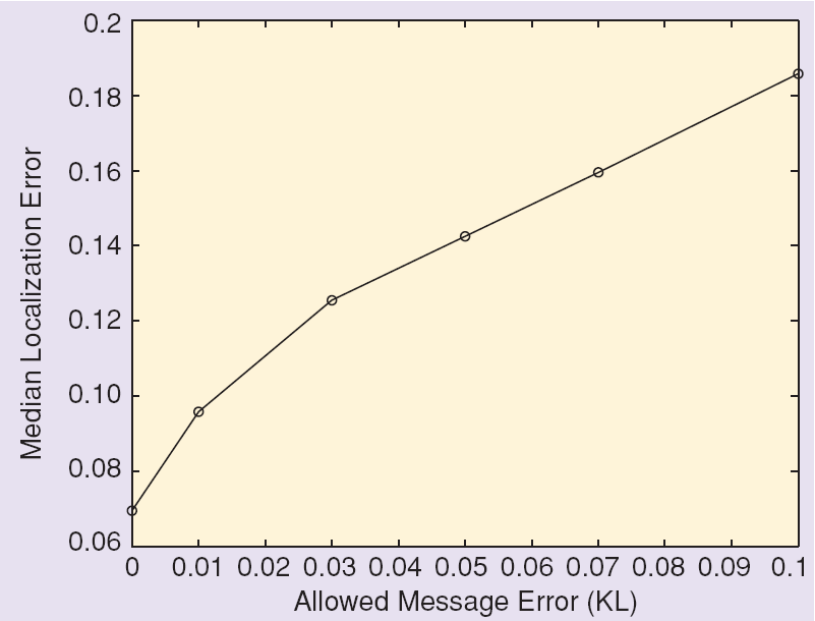
Trading off accuracy for Bits (5)

KD-tree benefits

- The tree structure allows an easy estimation of KLD
- Transmission protocol can take advantage of the structure of parent/child mean/covariances
- Adaption of message transmission
  - Specify a desired message accuracy ➔ leads to a cut through in the tree/message approximation
  - Specify a upper bound of bits for communication ➔ determines the most accurate approximation

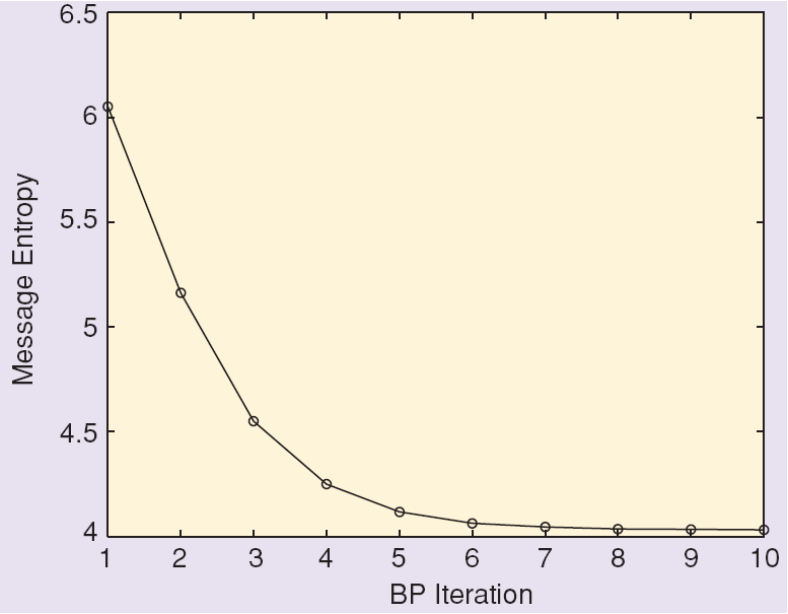
Trading off accuracy for Bits (6)

Trade off between message approximation error and localization error



Trading off accuracy for Bits (7)

Entropy behaviour in successive messages over iterations proceed



Effects of Message Approximation (1)

The preceding methods

- Message approximation due to censoring or to particle based approximation
- A relation between communication resources and the accuracy of the fused estimates

Difference between an exact and approximate message

- Kullback Leibler divergence
- ***Dynamic range measure***

Reminder of BP

Dynamic range (1)

- measure for $e_{ts}(x_s)$ $$d(e_{ts}) = \sup_{a,b} \sqrt{e_{ts}(a)/e_{ts}(b)}$$
- point wise equality condition $d(e_{ts}) = 1$ or $\log d(e_{ts}) = 0$
- Equivalently definition

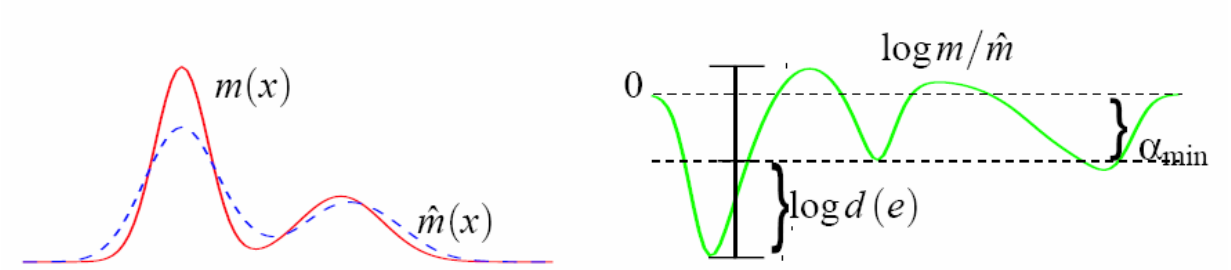
$$\log d(e_{ts}) = \inf_{\alpha} \sup_{x} |\log \alpha m_{ts}(x) - \log \hat{m}_{ts}(x)| = \inf_{\alpha} \sup_{x} |\log \alpha - \log e_{ts}(x)|$$

- It is directly related to the point wise error between two distributions

Dynamic range (2)

Scalar α

- „purpose of zero-centering“ – invariance of simple rescaling
- Acts as a scale factor defining a class of equivalent messages
- The closest message will be chosen in log-error sense



Dynamic range (3)

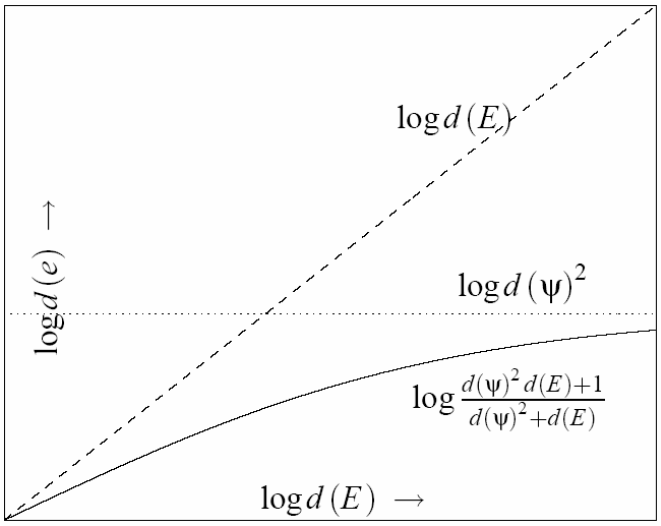
Other interesting properties

- Bounding of the error
- Error propagation

$$\log d\left(E_{ts}^{i}\right) \leq \sum_{u \in \Gamma_t \setminus s} \log d\left(e_{ut}^{i}\right)$$

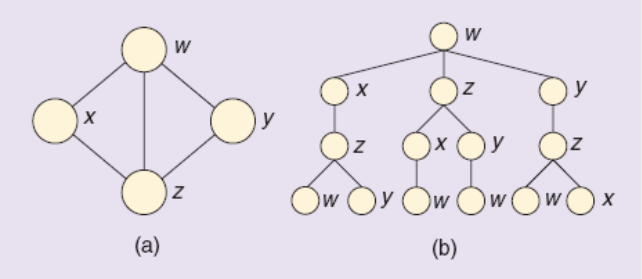
- Mixing of potentials
- Satisfies a rate of contraction

$$d\left(e_{ts}^{i+1}\right) \leq \frac{d\left(\psi_{ts}\right)^2 d\left(E_{ts}^{i}\right)+1}{d\left(\psi_{ts}\right)^2 + d\left(E_{ts}^{i}\right)}$$



Effects of Message Approximation (2)

- These properties provide a basis for bounding
- For loopy graph it is often visualized through „unwrapping“



- It is possible to compute a bound on the dynamic range after any number of iterations (without message approximation)
- Provide a strict bound for error

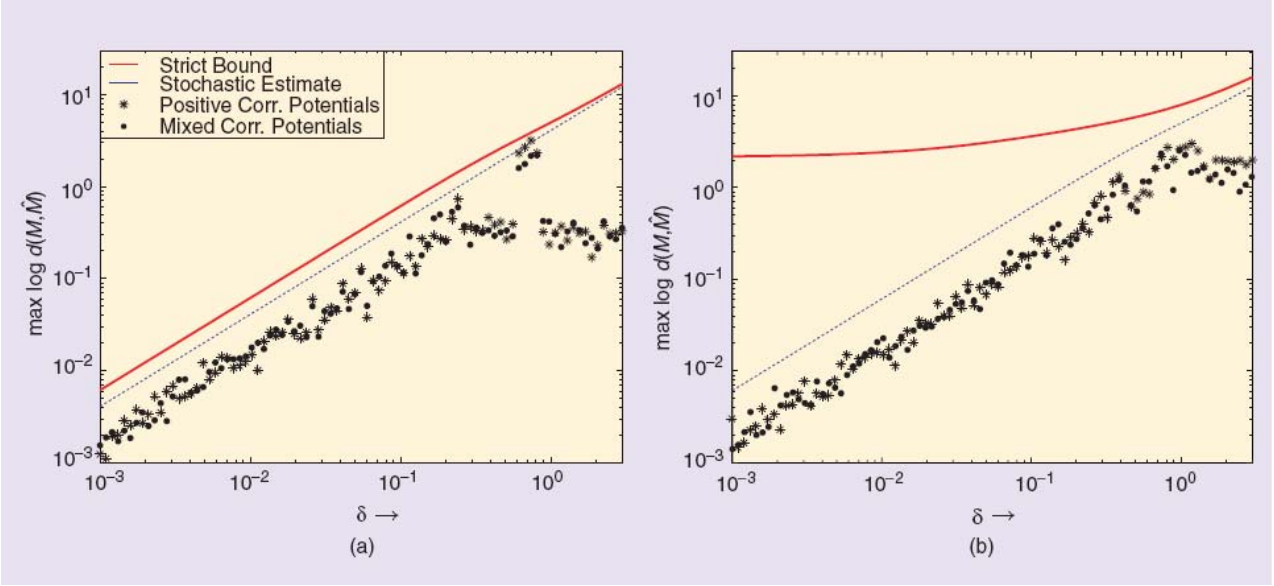
Effects of Message Approximation (3)

- This bounds seems to be to pessimistic to actual performance
- Assumption of uncorrelated error between nodes and iterations (crazy?)
- Empirically the system behave similarly to the predictions made by such a assumption
- However such a assumption is unsuitable for deriving strict convergence guarantees

Effects of Message Approximation (4)

Experimental results

- Maximum errors incurred over quantization error



Effects of Message Approximation (5)

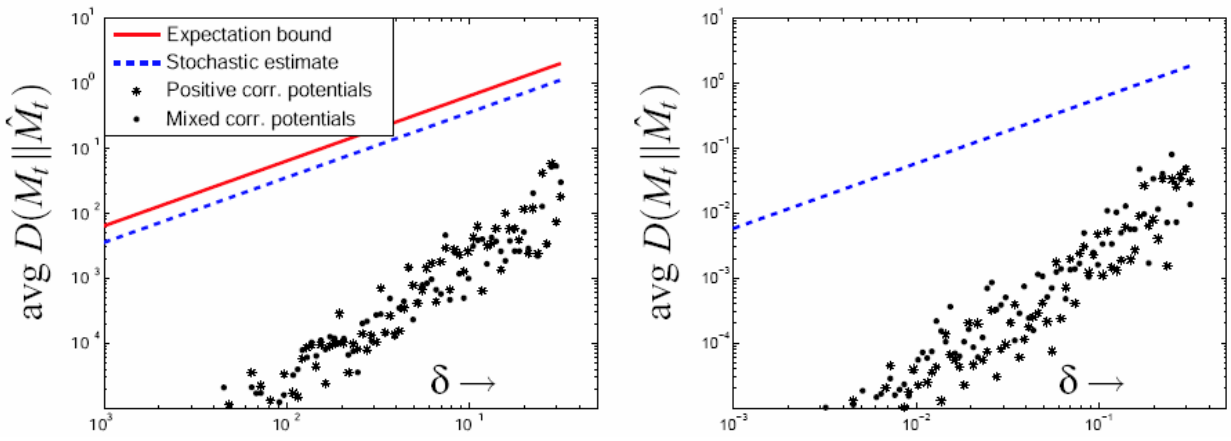
KL-divergence measures

- Dynamic range measure is to strict (?)
- KLD address to a weighted average
- Can such a measure fail (?)
- Many properties can not be fulfilled
  - Result is more an approximation than a bound
  - Error accumulation bounding, additive error, contraction rates
- Experiments

Effects of Message Approximation (6)

Again the same setup than before

- Different power of potentials
- Different quantization level



Summary (1)

Introduction

- Challenges in sensor networks
- Why graphical models are used?

Graphical models

- Definitions, Markov property, Factorization
- Sum Product Algorithm for calculating marginal distributions

Mapping a Network Application to a Graphical Model

- Self localization in Sensor Networks

Summary (2)

Message censoring & Bit trading

- Optimisation for communication
- First step to get some logic in one particular node
- Algorithm with more local word load

Effects of message approximation

- the message at each iteration is a noisy (erroneous) version of some true BP fixed point
- For two different error measures
- Mathematical model for error propagation

Final Slide

Thank you for your attention!

Beliefs are more powerful than facts.
Brian Herbert and Kevin Anderson, Dune: House of Atreides

References

[Çetin et al. 06] M. Çetin, L. Chen, J.W. Fisher III, A.T. Ihler, R.L. Moses, M. J. Wainwright, and A.S. Willsky: “Distributed Fusion in Sensor Networks”, in IEEE Signal Processing Magazine, July 2006

[Ihler et al. 05] A.T. Ihler, J.W. Fisher III, and A.S. Willsky: “Loopy Belief Propagation: Convergence and Effects of Message Errors”, in Journal of Machine Learning Research 6 (2005) 905–936

[Roweis 06] S. Roweis: ”Probabilistic Graphical Models”, Lecture notes at Machine Learning Summer School, Taiwan, July, 2006

[Shlens 07] J. Shlens: “Notes on Kullback-Leibler Divergence and Likelihood Teory”, System Neurobiology Laboratory, Salk Institute for Biological Studies, California, August 2007

[Ihler et al.] A.T. Ihler, J.W. Fisher III, R. L. Moses, and A.S. Willsky: “Nonparametric Belief Propagation for Self-Localization of Sensor Networks”, submitted to IEEE JOURNAL ON SEL. AREAS IN COMM., SPECIAL ISSUE ON COLLABORATIVE SENSOR NETWORKS