Decentralized decision making with spatially distributed data

XuanLong Nguyen
Department of Statistics
University of Michigan

Acknowledgement:
Michael Jordan, Martin Wainwright, Ram Rajagopal, Pravin Varaiya

Decentralized systems and spatial data

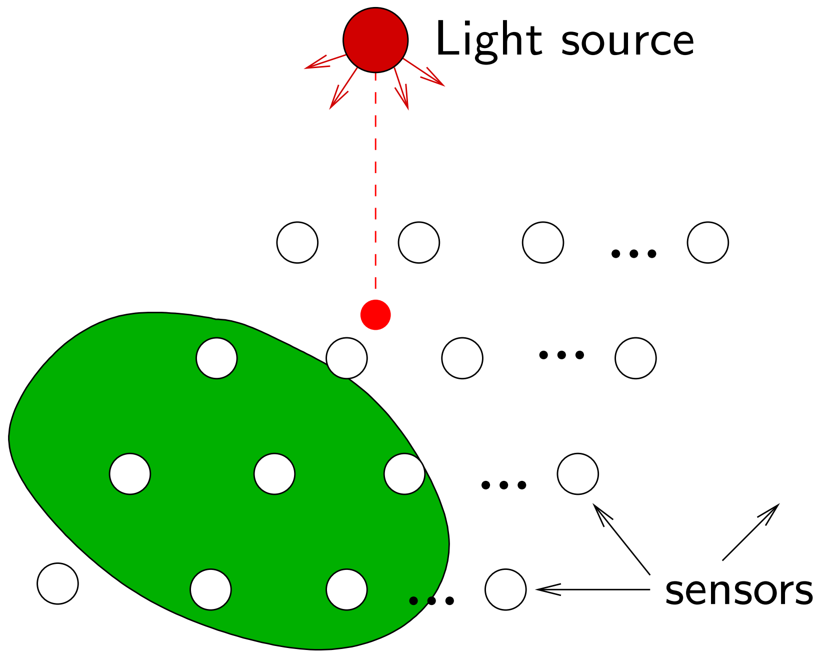
- Many applications and systems involve collecting and transmitting large volume of data through distributed network (sensor signals, image streams, network system logs, etc)

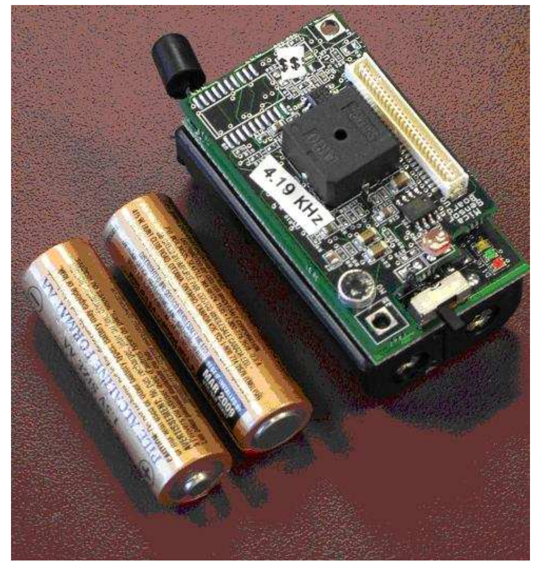
- Two interacting and conflicting forces
  - statistical inference and learning arise from spatial dependence
  - decentralized communication and computations

Decentralized systems and spatial data

- Many applications and systems involve collecting and transmitting large volume of data through distributed network (sensor signals, image streams, network system logs, etc)
- Two interacting and conflicting forces
  - statistical inference and learning arise from spatial dependence
  - decentralized communication and computations
- Extensive literature dealing with each of these two aspects separately
- We are interested in decentralized learning and decision-making methods for spatially distributed data
  - computation/communication efficiency vs. statistical efficiency

Example 1 – sensor network for detection



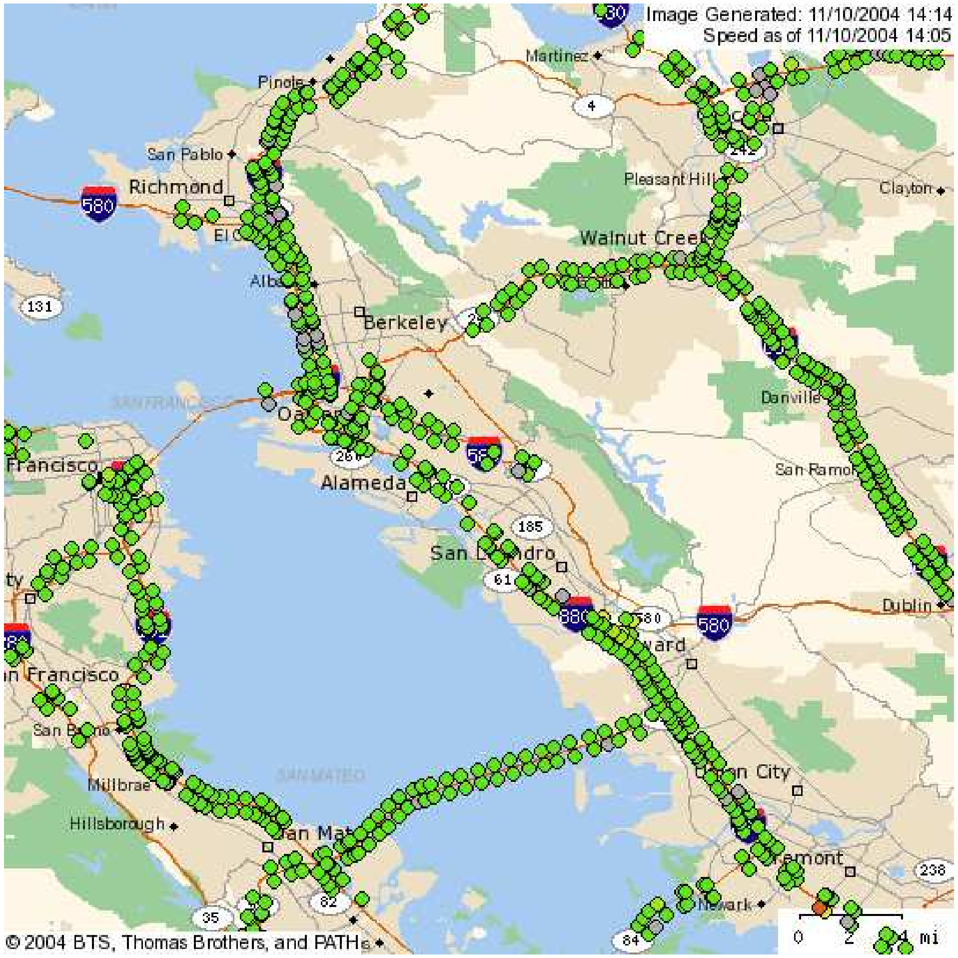


Set-up:

- Wireless network of tiny sensor motes, each equiped with light/ humidity/ temperature sensing capabilities
- Measurement of signal strength ([0–1024] in magnitude, or 10 bits)

Common goal: Is the light source inside the green region or not?

Example 2 – sensor network for traffic monitoring



Multiple goals: Different sensors measuring different locations

Two types of set-ups

- aggregation of data to make a good decision toward a common goal
  - all sensors collect measurements of the same phenomenon and report their messages to a fusion center

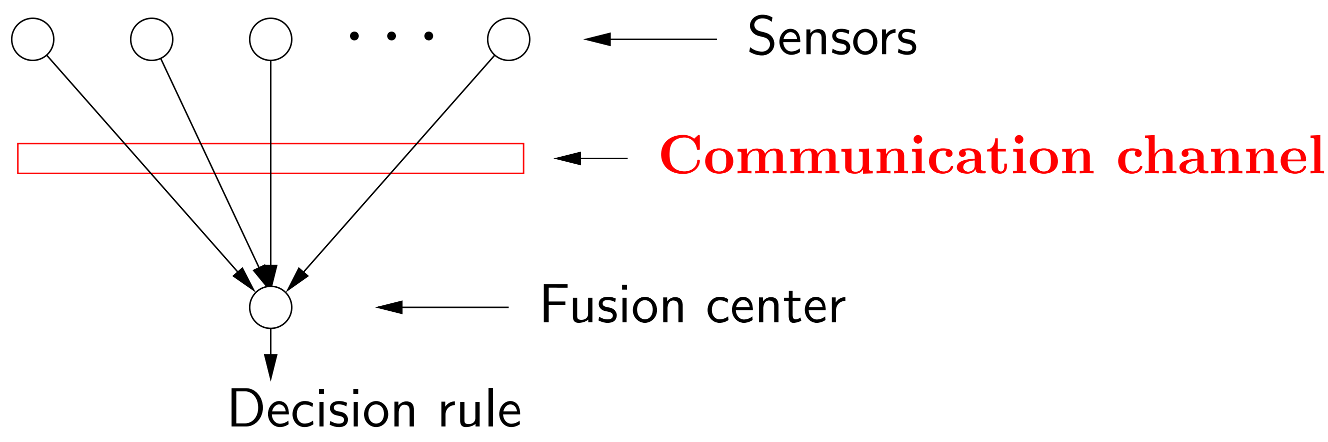
- completely distributed network of sensors – each making separate decisions for own goal
  - different sensors have statistically dependent measurements about one or more phenomena of interest

Talk outline

- Set-up 1: decentralized detection (classification) problem
  - algorithmic and modeling ideas (marginalized kernels, convex optimization)
  - statistical properties (use of surrogate loss and f-divergence)

- Set-up 2: completely distributed decision-making for multiple sensors
  - algorithmic ideas (message-passing in graphical models)
  - statistical tools (from sequential analysis)

A decentralized detection system

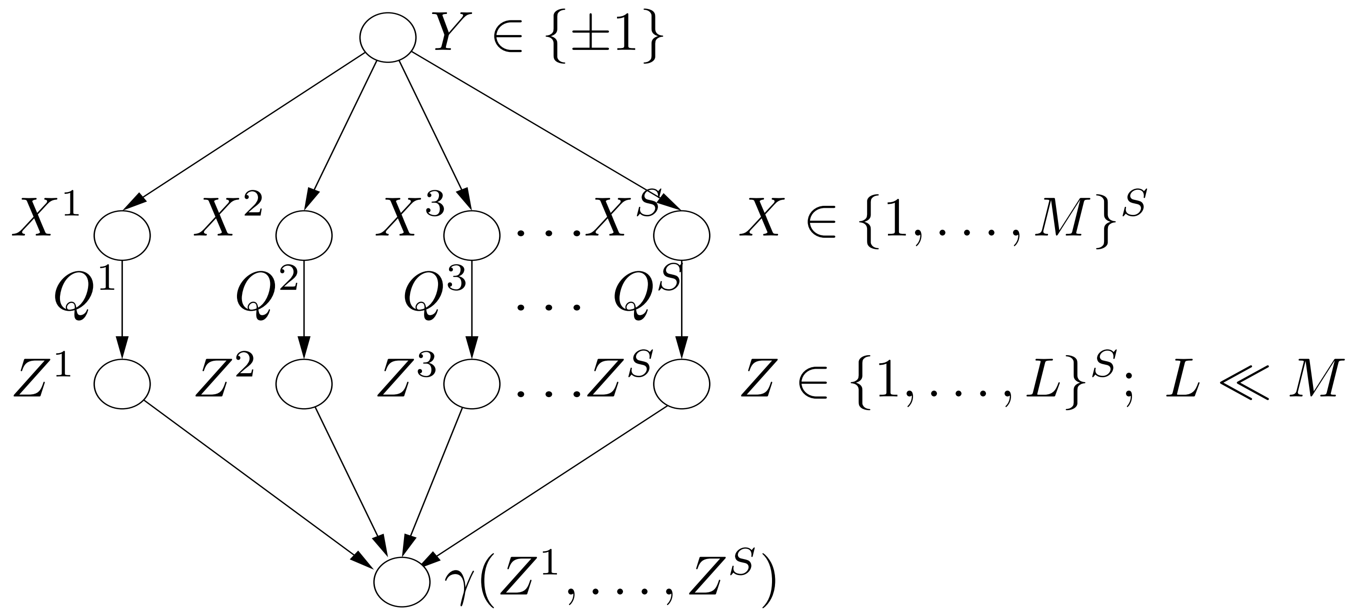


- **Decentralized setting:** Communication constraints between sensors and fusion center (e.g., bit constraints)

- **Goal:** Design decision rules for sensors and fusion center

- **Criterion:** Minimize *probability of incorrect detection*

Problem set-up



Problem: Given training data $(x_i, y_i)_{i=1}^n$, find the decision rules $(Q^1, \ldots, Q^s; \gamma)$ so as to minimize the detection error probability:

$$P(Y \neq \gamma(Z^1, \ldots, Z^s)).$$

Decentralized detection

- General set-up:
  - data are (X, Y) pairs, assumed iid for simplicity, where $Y \in \{0, 1\}$
  - given X, let $Z = Q(X)$ denote the covariate vector, where $Q \in \mathcal{Q}$
  - $\mathcal{Q}$ is some set of random mappings, namely, quantizers
  - a family of $\{\gamma(\cdot)\}$, where γ is a discriminant decision function lying in some (nonparametric) family Γ

- Problem: Find decision (Q, γ) that minimizes the probability of error $P(Y \neq \gamma(Z))$

- Many problems have similar formulation:
  - decentralized compression and detection
  - feature selection, dimensionality reduction
  - problem of sensor placement

Perspectives

- *Signal processing literature*
  - everything is assumed known except for Q – the problem is to find Q subject to network system constraints
  - maximization of an "f-divergence" (e.g., Hellinger distance, Chernoff distance)
  - basically a heuristic literature from a statistical perspective (plug-in estimation)
  - supporting arguments from asymptotics

- *Statistical learning literature*
  - decision-theoretic flavor
  - Q is assumed known and the problem is to find γ
  - this is done via minimization of a "surrogate convex loss" (e.g., boosting, logistic regression, support vector machine)

Overview of our approach

- Treat as a nonparametric joint learning problem
  - estimate both Q and γ
  - subject to constraints from a distributed system
- Use kernel methods and convex surrogate loss functions
  - tools from convex optimization to derive an efficient algorithm
- Exploit a correspondence between surrogate losses and divergence functionals
  - obtains consistency of learning procedure

Kernel methods for classification

- Classification: Learn $\gamma(z)$ that predicts label y
- $K(z, z')$ is a *symmetric positive semidefinite* kernel function
  - natural choice of basis function for spatially distributed data

- *feature space* $\mathcal{H}$ in which K acts as an inner product, i.e., $K(z, z') = \langle \Psi(z), \Psi(z') \rangle$

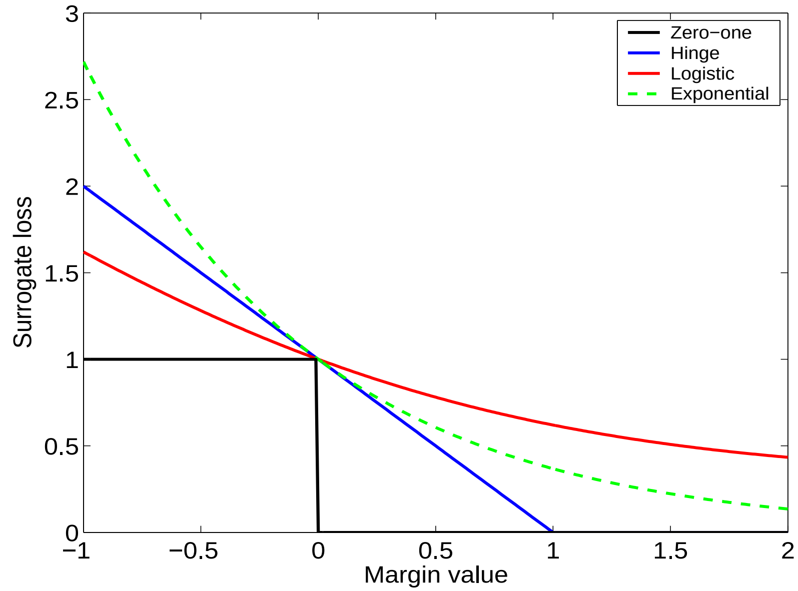
- Kernel-based algorithm finds linear function in $\mathcal{H}$, i.e.

$$\gamma(z) = \langle \mathrm{w}, \Psi(z) \rangle$$

- Advantages:
  - optimizing over kernel function classes is computionally efficient
  - solution γ is represented in terms of kernels only:

$$\gamma(z) = \sum_{i=1}^{n} \alpha_i K(z_i, z)$$

Convex surrogate loss functions $\phi(\alpha)$

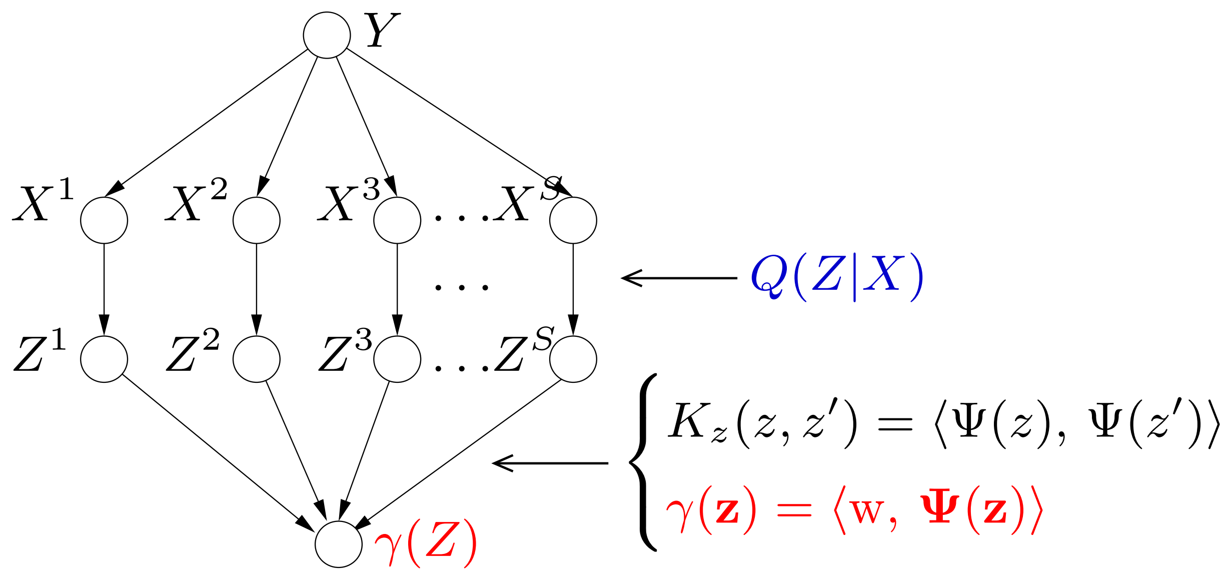


- minimizing (regularized) empirical ϕ-risk $\hat{E}\phi(Y\gamma(Z))$:

$$\min_{\gamma \in \mathcal{H}} \sum_{i=1}^{n} \phi(y_i \gamma(z_i)) + \frac{\lambda}{2} \|\gamma\|^2,$$

- $(z_i, y_i)_{i=1}^n$ are training data in $\mathcal{Z} \times \{\pm 1\}$
- ϕ is a convex *loss function* (upper bound of 0-1 loss)

Stochastic decision rules at each sensor



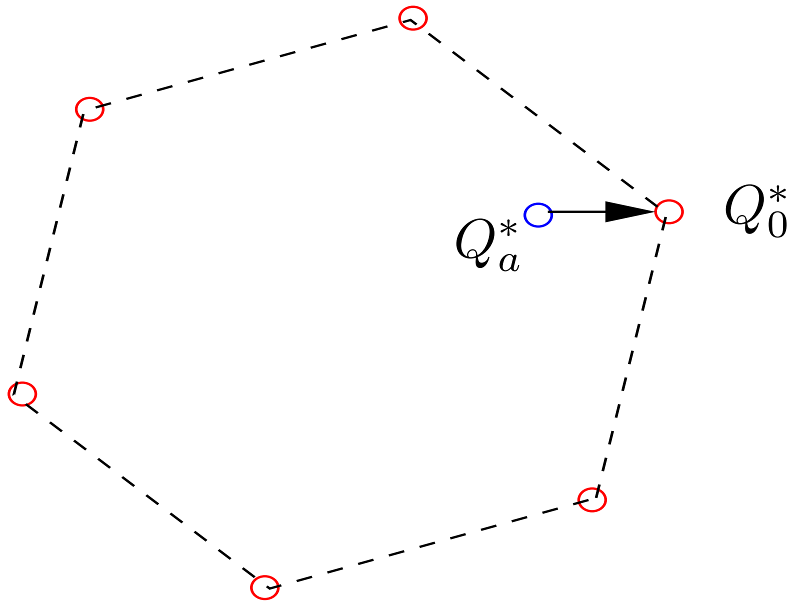
- Approximate deterministic sensor decisions by stochastic rules $Q(Z|X)$
- Sensors do not communicate directly $\Longrightarrow$ factorization: $Q(Z|X) = \prod_{t=1}^{S} Q^t(Z^t|X^t)$
- The overall decision rule is represented by $\begin{cases} \mathbf{Q} = \prod \mathbf{Q^t}, \\ \gamma(\mathbf{z}) = \langle \mathrm{w},\ \mathbf{\Psi}(\mathbf{z}) \rangle \end{cases}$

High-level strategy:
Joint optimization

- Minimize over (Q, γ) an empirical version of $\mathbb{E}\phi(Y\gamma(Z))$
- Joint minimization:
  - fix Q, optimize over γ: A simple convex problem
  - fix γ, perform a gradient update for Q, sensor by sensor

High-level strategy:

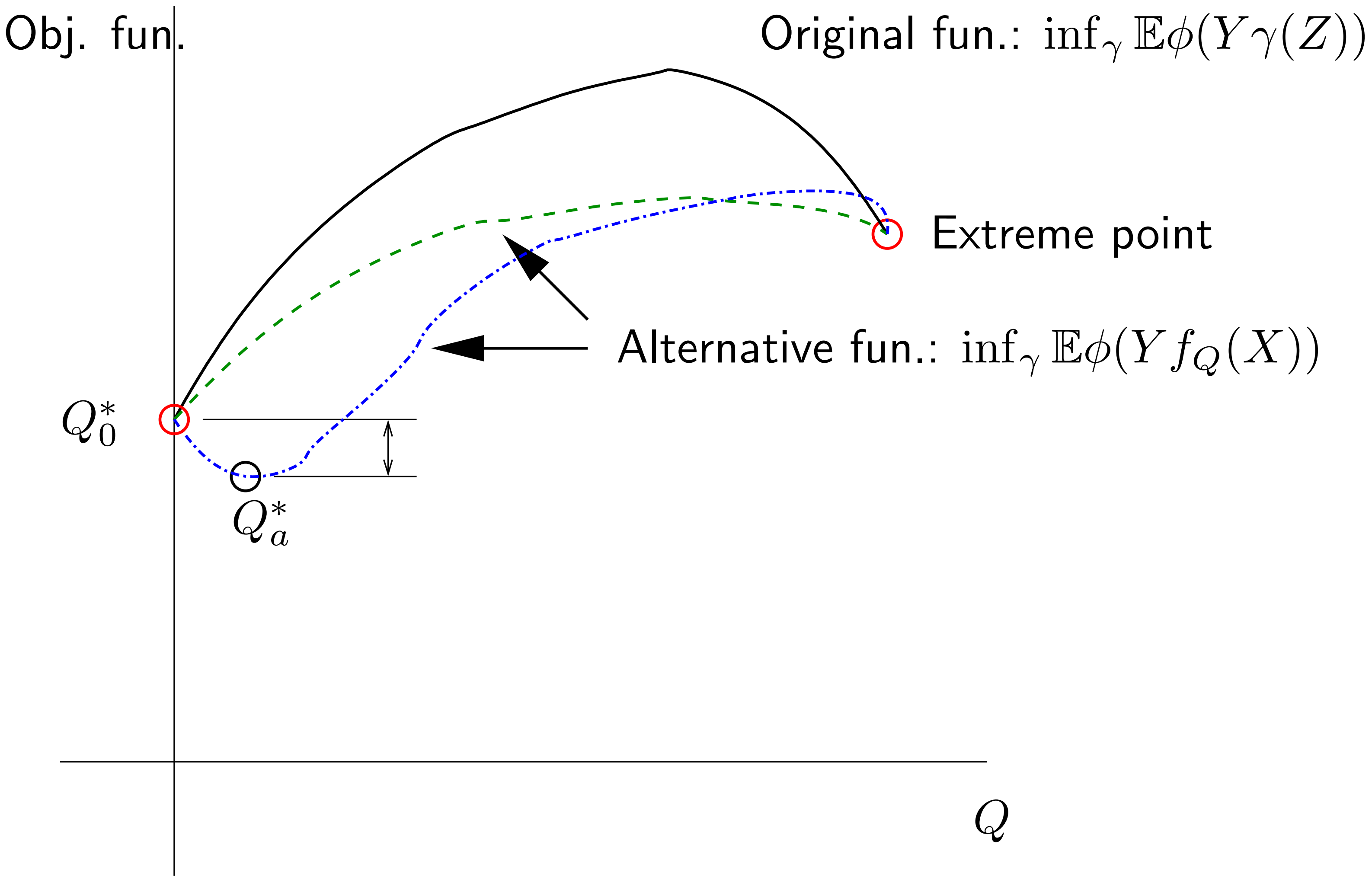
Space of stochastic quantization rule Q



- is convex hull of the set of deterministic Q
- optimal decision rule Q_0^* is deterministic
- optimizing over deterministic rules is NP-hard

High-level strategy:

Alternative objective function



Approximating empirical ϕ-risk

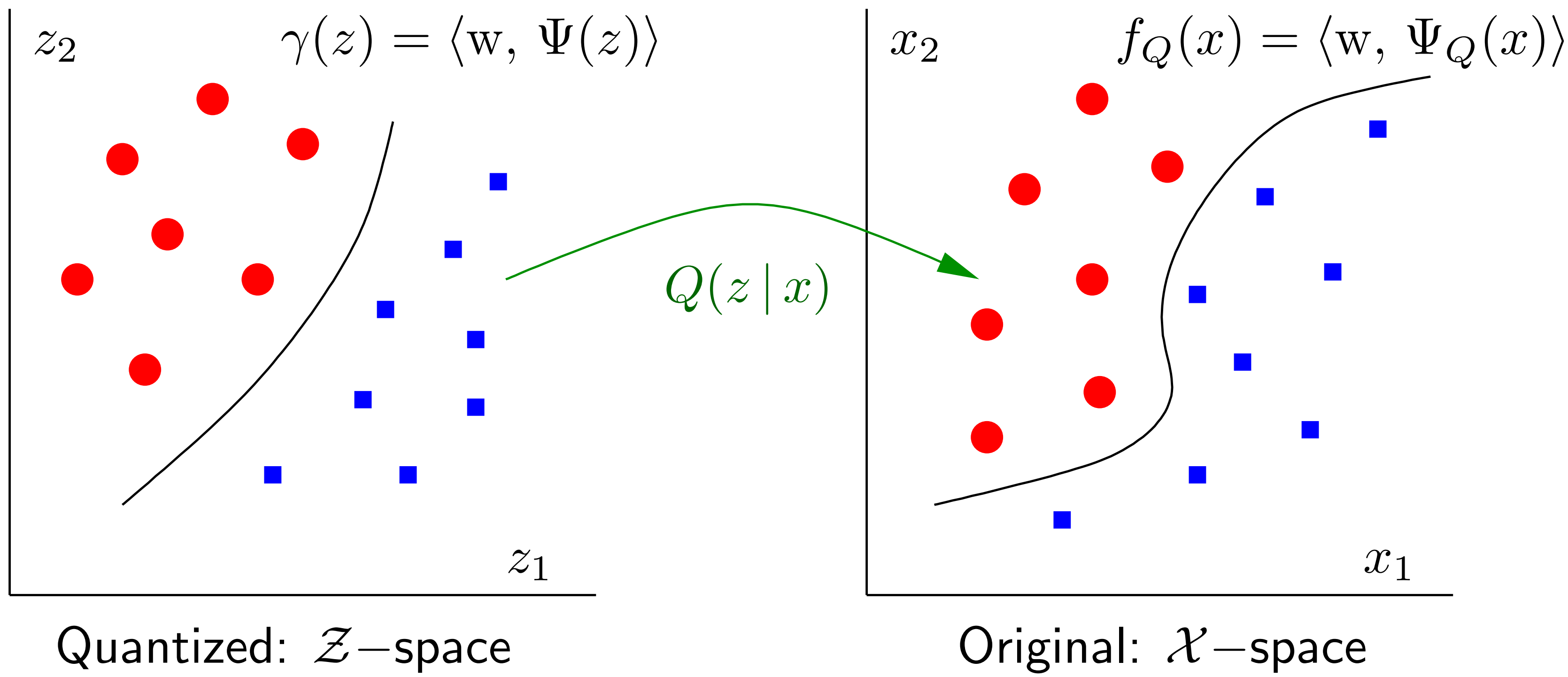
- The regularized empirical ϕ-risk $\hat{\mathbb{E}}\phi(Y\gamma(Z))$ has the form:

$$G_0 = \sum_z \sum_{i=1}^{n} \phi(y_i \gamma(z)) Q(z|x_i) + \frac{\lambda}{2}||\mathrm{w}||^2$$

- **Challenge:** Even evaluating G_0 at a single point is intractable
  Requires summing over L^S possible values for z

- **Idea**:
  - Approximate G_0 by another objective function G
  - $G_0 \equiv G$ for deterministic Q

"Marginalizing" over feature space



Stochastic decision rule $Q(z \,|\, x)$:

- maps between $\mathcal{X}$ and $\mathcal{Z}$
- induces marginalized feature map Ψ_Q from base map Ψ (or marginalized kernel K_Q from base kernel K)

Marginalized feature space $\{\Psi_Q(x)\}$

Marginalized feature space $\{\Psi_Q(x)\}$

- Define a new feature space $\Psi_Q(x)$ and a linear function over $\Psi_Q(x)$:

$$\begin{cases} \Psi_Q(x) = \sum_z Q(z|x)\Psi(z) & \Longleftarrow \text{ Marginalization over } z \\ f_Q(x) = \langle w, \Psi_Q(x) \rangle \end{cases}$$

Marginalized feature space $\{\Psi_Q(x)\}$

- Define a new feature space $\Psi_Q(x)$ and a linear function over $\Psi_Q(x)$:

$$\begin{cases} \Psi_Q(x) = \sum_z Q(z|x)\Psi(z) & \Longleftarrow \text{Marginalization over } z \\ f_Q(x) = \langle w, \Psi_Q(x) \rangle \end{cases}$$

- The alternative objective function G is the ϕ-risk for f_Q:

$$G = \sum_{i=1}^{n} \phi(y_i f_Q(x_i)) + \frac{\lambda}{2} \|\mathrm{w}\|^2$$

Marginalized feature space $\{\Psi_Q(x)\}$

- Define a new feature space $\Psi_Q(x)$ and a linear function over $\Psi_Q(x)$:

$$\begin{cases} \Psi_Q(x) \;=\; \sum_z Q(z|x)\Psi(z) & \Longleftarrow \text{ Marginalization over } z \\ f_Q(x) \;=\; \langle w,\, \Psi_Q(x)\rangle \end{cases}$$

- The alternative objective function G is the ϕ-risk for f_Q:

$$G = \sum_{i=1}^{n} \phi(y_i f_Q(x_i)) + \frac{\lambda}{2}\|\mathrm{w}\|^2$$

- $\Psi_Q(x)$ induces a marginalized kernel over $\mathcal{X}$:

$$K_Q(x,x') := \langle \Psi_Q(x),\, \Psi_Q(x')\rangle = \sum_{z,z'} Q(z|x)Q(z'|x')\, K_z(z,z')$$

$\Rightarrow$ Marginalization taken over message z conditioned on sensor signal x

Marginalized kernels

- Have been used to derive kernel functions from generative models (e.g. Tsuda, 2002)

- Marginalized kernel $K_Q(x, x')$ is defined as:

$$K_Q(x, x') := \sum_{z,z'} \underbrace{Q(z|x)Q(z'|x')}_{\text{Factorized distributions}} \underbrace{K_z(z, z')}_{\text{Base kernel}},$$

- If $K_z(z, z')$ is decomposed into smaller components of z and z', then $K_Q(x, x')$ can be computed efficiently (in polynomial-time)

Centralized and decentralized function

- Centralized decision function obtained by minimizing ϕ-risk:

$$f_Q(x) = \langle \mathrm{w}, \, \Psi_Q(x) \rangle$$

  - f_Q has direct access to sensor signal x

Centralized and decentralized function

- Centralized decision function obtained by minimizing ϕ-risk:

$$f_Q(x) = \langle \mathrm{w}, \Psi_Q(x) \rangle$$

  - f_Q has direct access to sensor signal x

- Optimal w also define decentralized decision function:

$$\gamma(z) = \langle \mathrm{w}, \Psi(z) \rangle$$

  - γ has access only to quantized version z

Centralized and decentralized function

- Centralized decision function obtained by minimizing ϕ-risk:

$$f_Q(x) = \langle \mathrm{w}, \, \Psi_Q(x) \rangle$$

  - f_Q has direct access to sensor signal x

- Optimal w also define decentralized decision function:

$$\gamma(z) = \langle \mathrm{w}, \, \Psi(z) \rangle$$

  - γ has access only to quantized version z

- Decentralized γ behaves *on average* like the centralized f_Q:

$$f_Q(x) = \mathbb{E}[\gamma(Z)|x]$$

Optimization algorithm

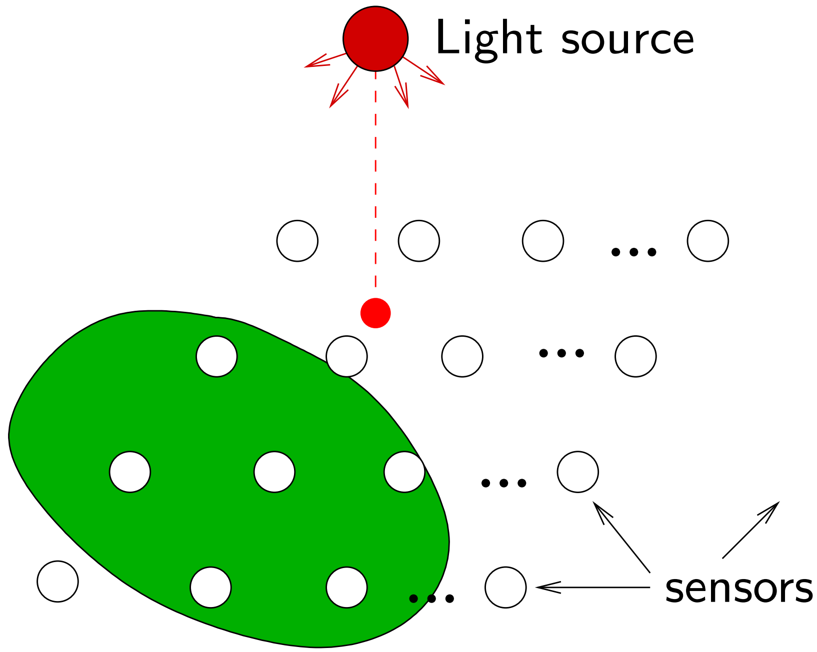
Goal: Solve the problem:

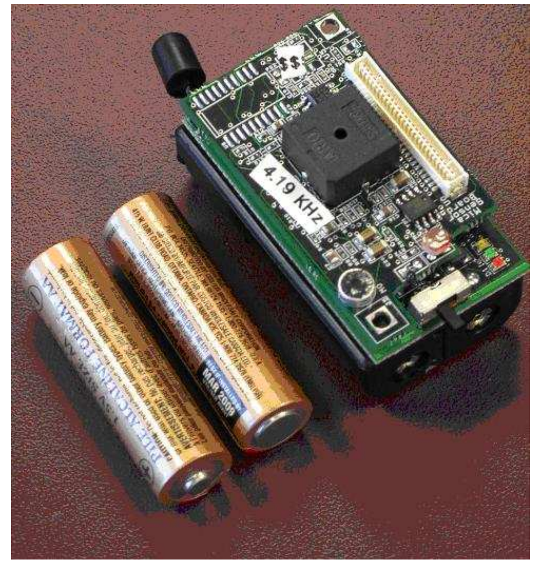
$$\inf_{\mathrm{w};Q} G(\mathrm{w};Q) := \sum_i \phi\Big(y_i \langle \mathrm{w}, \sum_z Q(z|x_i)\Psi(z)\rangle\Big) + \frac{\lambda}{2}||\mathrm{w}||^2$$

- Finding optimal weight vector:
  - G is convex in w with Q fixed
  - solve dual problem (quadratically-constrained convex program) to obtain optimal $\mathrm{w}(Q)$
- Finding optimal decision rules:
  - G is convex in Q^t with w and all other $\{Q^r, r \neq t\}$ fixed
  - efficient computation of *subgradient* for G at optimal $(\mathrm{w}(Q), Q)$

Overall: Efficient joint minimization by blockwise coordinate descent

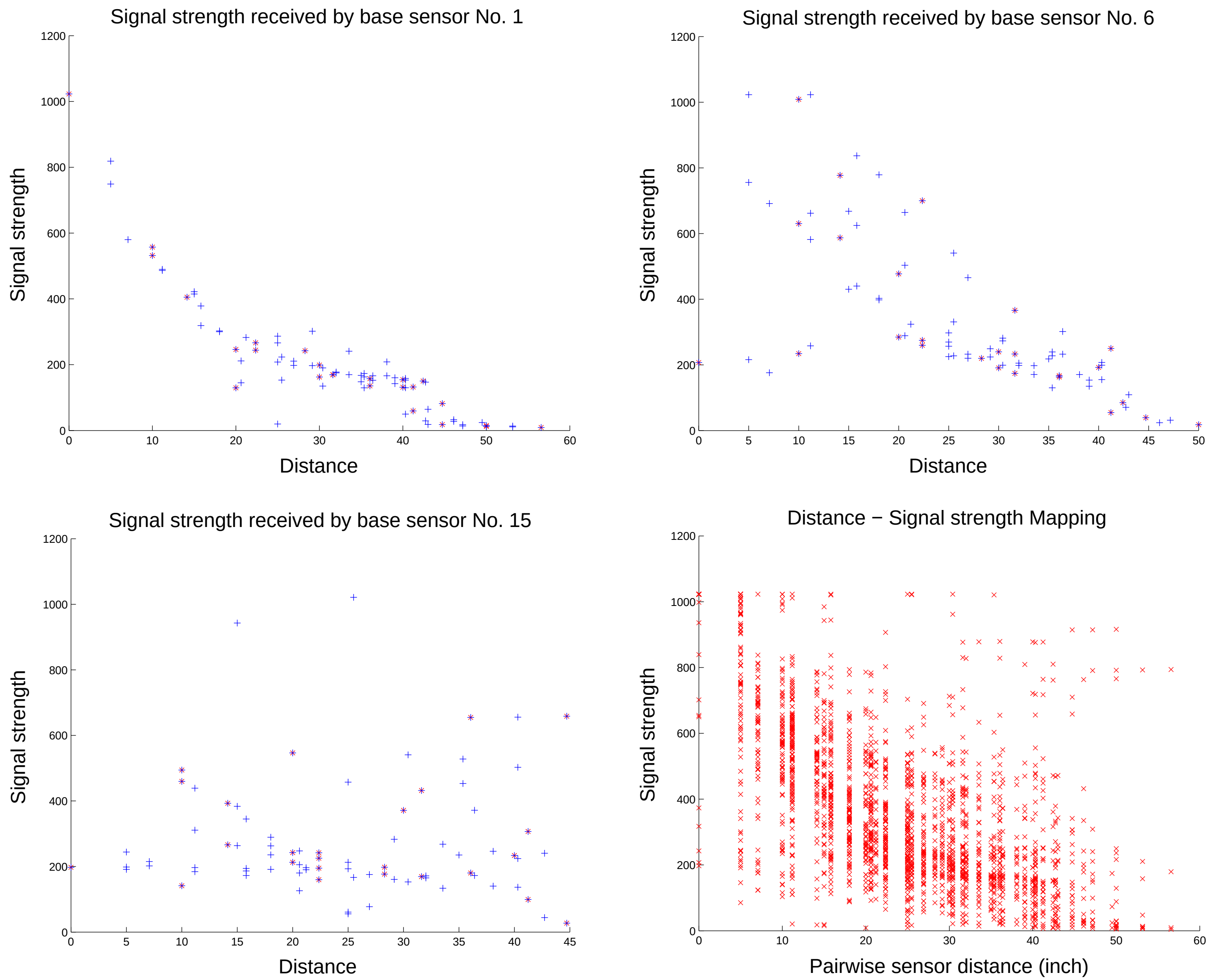
Wireless network with Mica motes



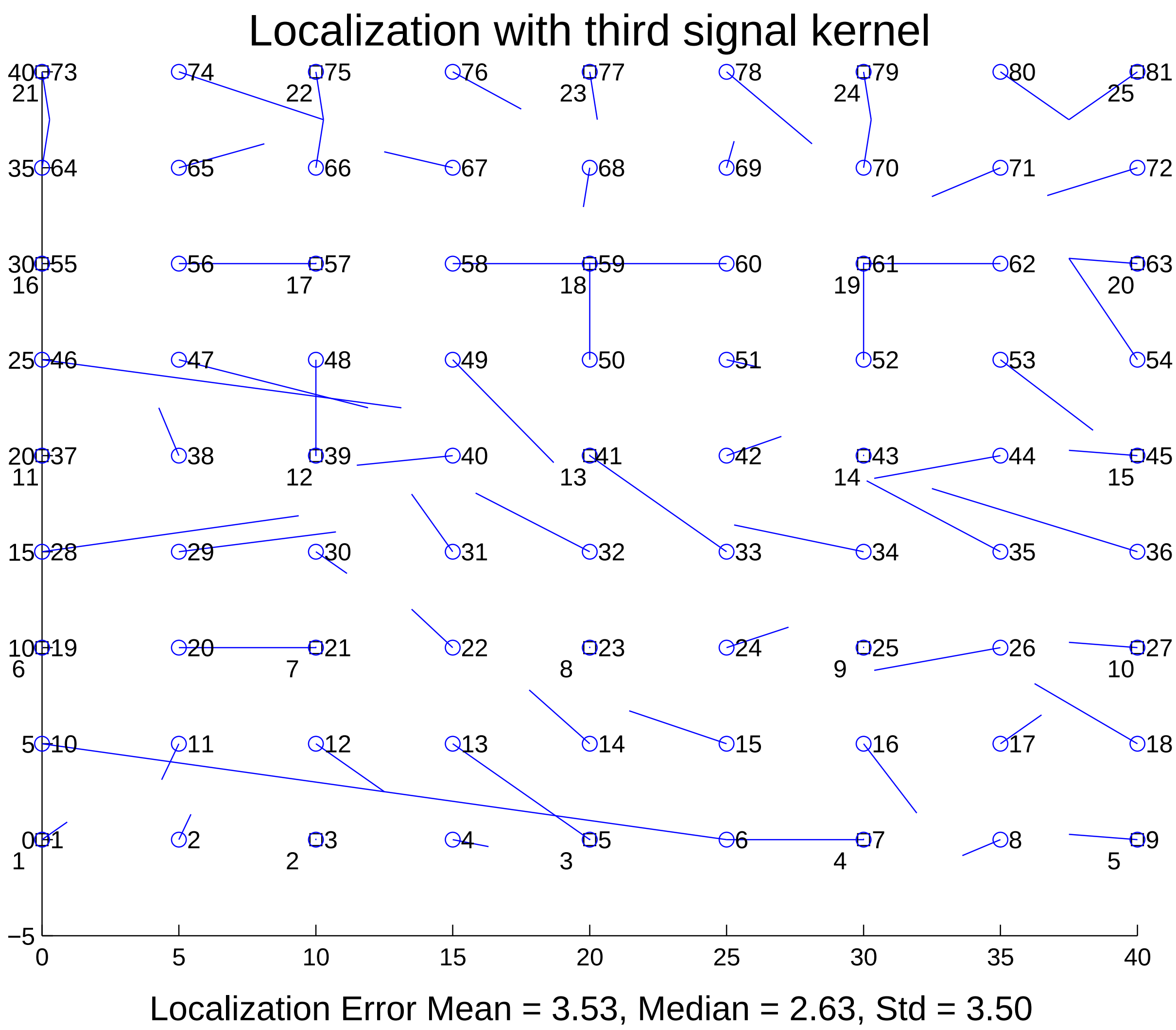


- $5 \times 5 = 25$ tiny sensor motes, each equipped with a light receiver
- Light signal strength requires **10-bit** ([0–1024] in magnitude)
- Perform classification with respect to different regions, subject to bit constraints
- Each problem has 25 training positions, 81 test positions

Wireless sensor network data (light signal)

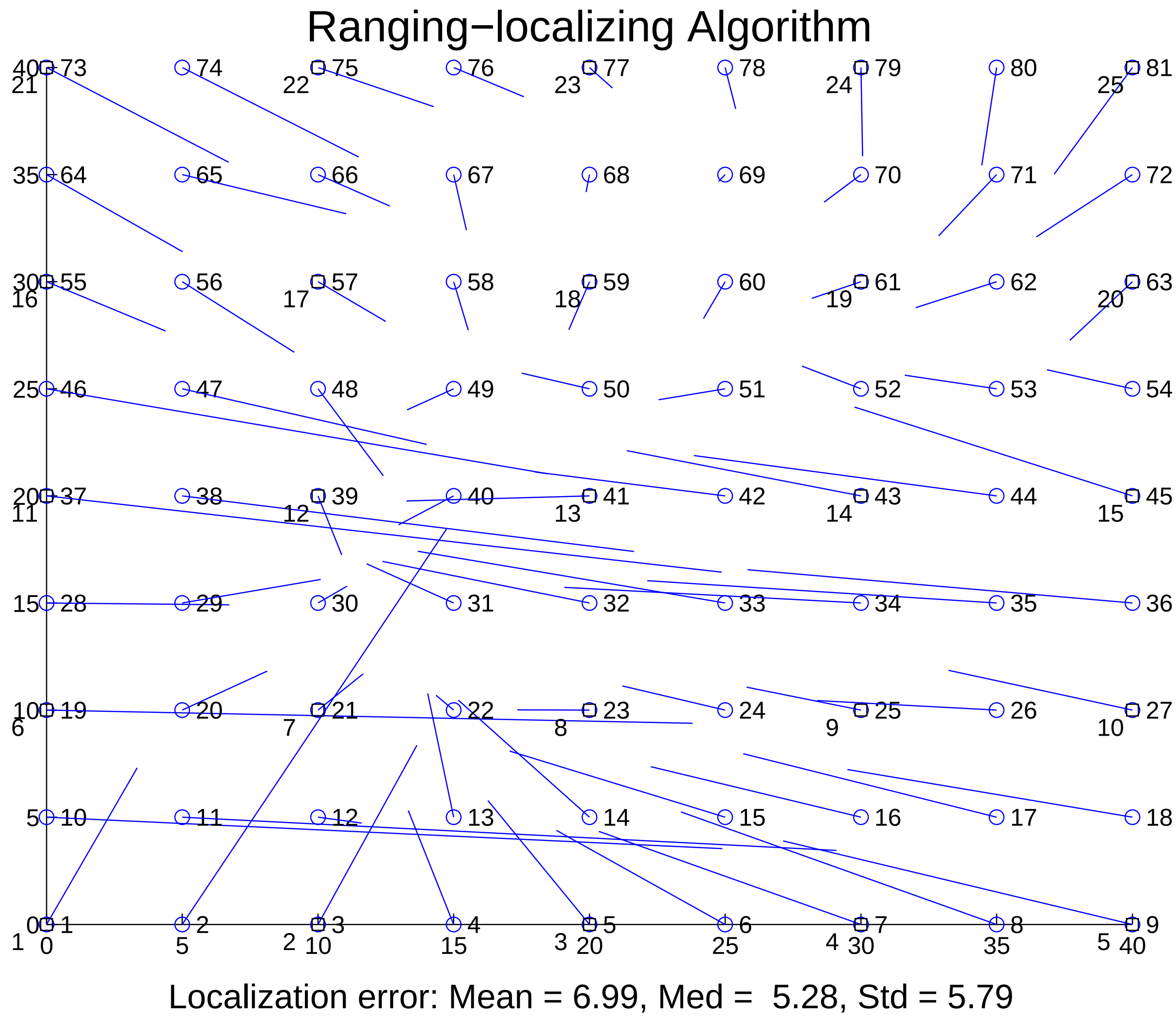


Location estimation result

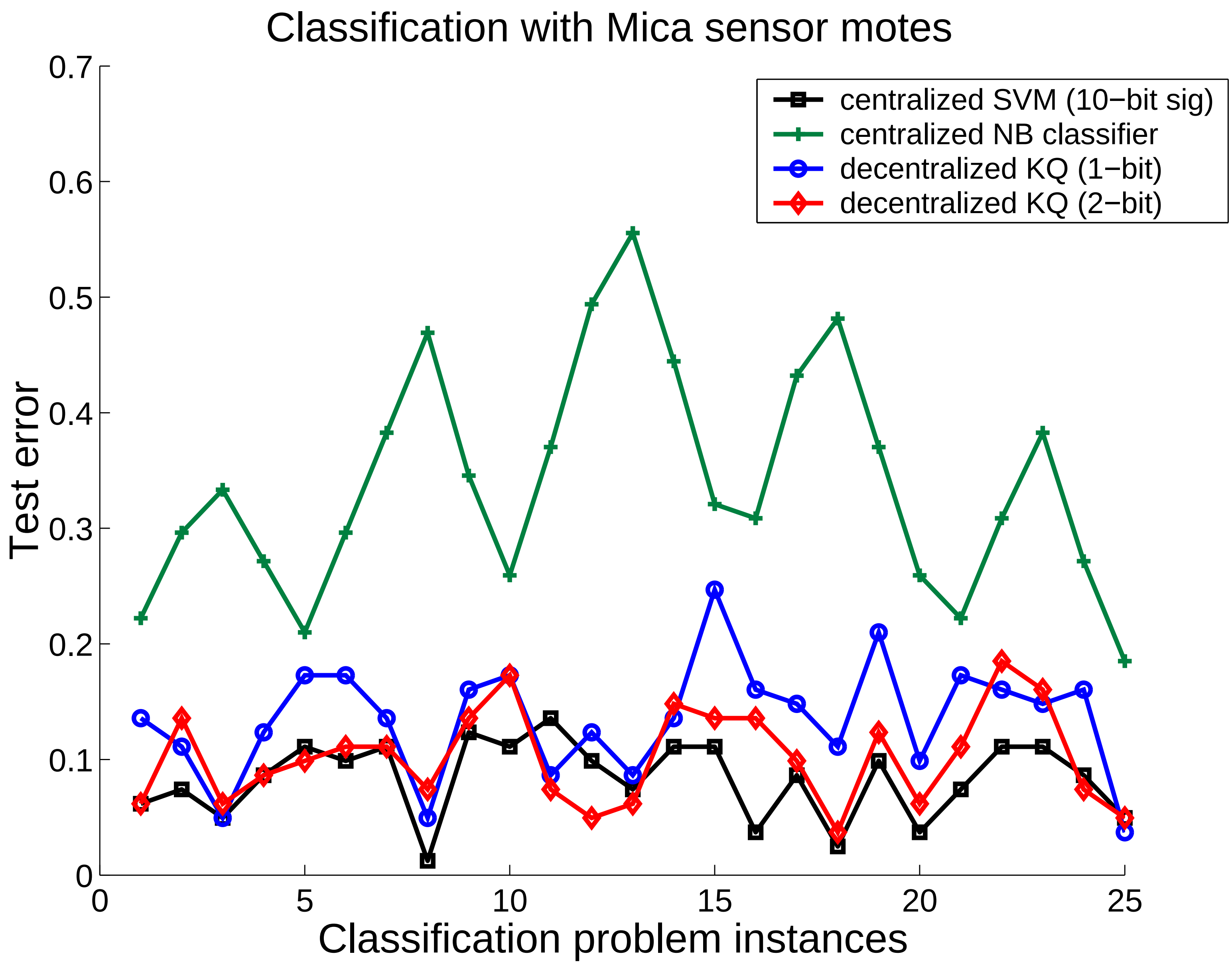


- compare to a well-known range-based method: (6.99, 5.28, 5.79)

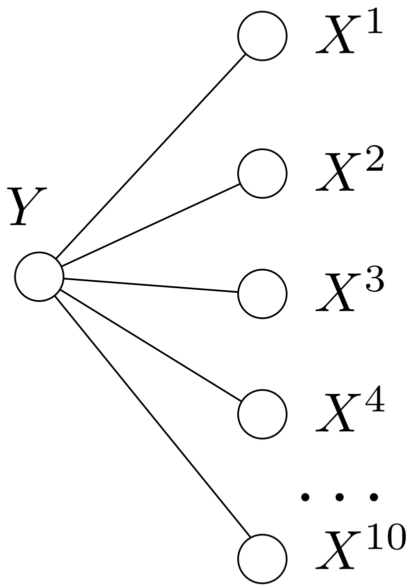
Location estimation result (existing method)



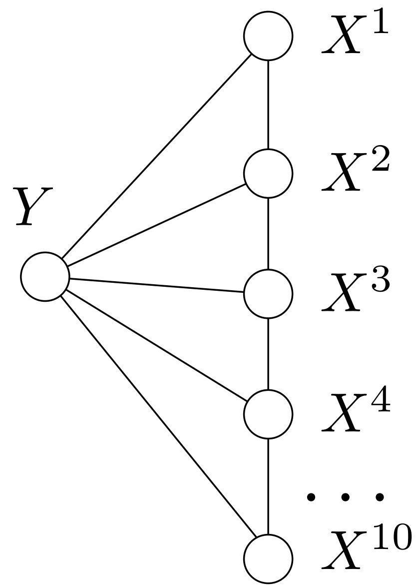
- compare to our kernel-based learning method: (3.53, 2.63, 3.50)


Classification with Mica sensor motes
Test error
Classification problem instances
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
5
10
15
20
25
centralized SVM (10−bit sig)
centralized NB classifier
decentralized KQ (1−bit)
decentralized KQ (2−bit)

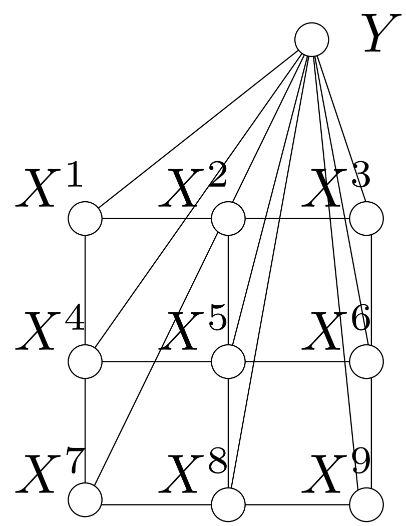
Simulated sensor networks



Naive Bayes net

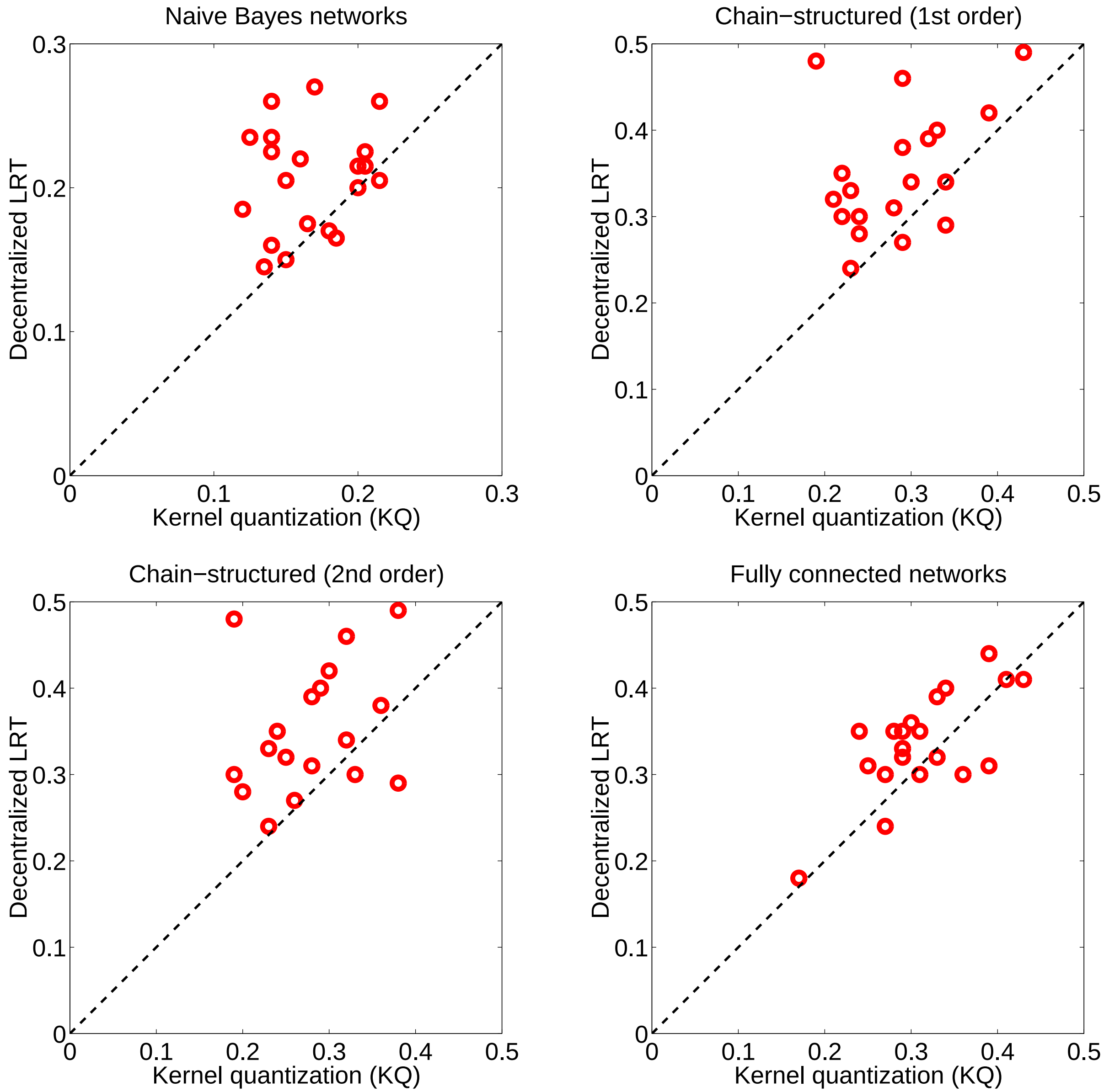


Chain-structured network



Spatially-dependent network

Joint estimation method vs. decentralized LRT



Talk outline

- decentralized detection (classification) problem
  - algorithmic and modeling ideas (marginalized kernels, convex optimization)
  - statistical properties (use of surrogate loss and f-divergence)

- completely distributed decision making for multiple sensors
  - algorithmic ideas (message-passing in graphical models)
  - statistical tools (from sequential analysis)

Statistical properties of surrogate losses

- recall that our algorithm essentially solves

$$\min_{\gamma,Q} \mathbb{E}\phi(Y, \gamma(Z))$$

- does this also implies optimality in the sense of 0-1 loss?

- the answer lies in the *correspondence between loss functions and divergence functionals*

Intuitions about loss functions and divergences

- loss functions quantify our decision rules
  - the sensor messages, and the classifier at the fusion center

- divergences quantify the distance (separation) between two probability distributions (populations of data)

- the best sensor messages and classifier is the one that best separate the two populations of data (corresponding to two class label $Y = \{\pm 1\}$)

- thus, loss functions and divergences are *dual* of one another:
  - minimize a loss function is equivalent to maximizing an associated divergence

f-divergence (Ali-Silvey Distance)

The f-divergence between two densities μ and π is given by

$$I_f(\mu, \pi) := \int_z \pi(z) f\left(\frac{\mu(z)}{\pi(z)}\right) d\nu.$$

where $f : [0, +\infty) \to \mathbb{R} \cup \{+\infty\}$ is a continuous convex function

f-divergence (Ali-Silvey Distance)

The f-divergence between two densities μ and π is given by

$$I_f(\mu, \pi) := \int_z \pi(z) f\left(\frac{\mu(z)}{\pi(z)}\right) d\nu.$$

where $f : [0, +\infty) \to \mathbb{R} \cup \{+\infty\}$ is a continuous convex function

- Kullback-Leibler divergence: $f(u) = u \log u$.

$$I_f(\mu, \pi) = \int_z \mu(z) \log \frac{\mu(z)}{\pi(z)}.$$

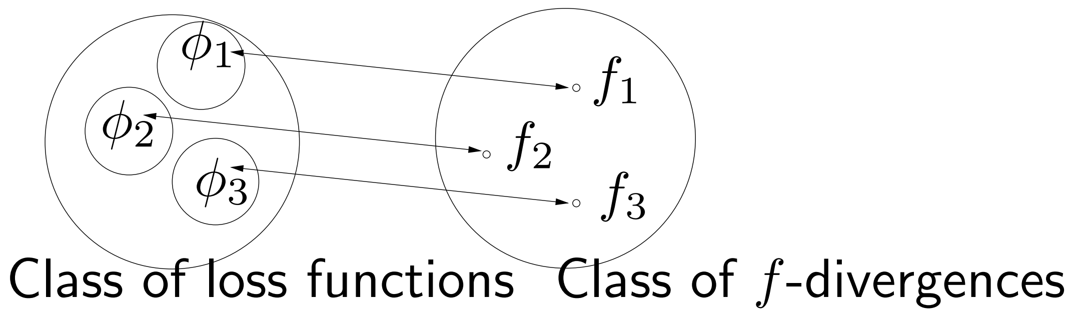
- variational distance: $f(u) = |u - 1|$.

$$I_f(\mu, \pi) := \int_z |\mu(z) - \pi(z)|.$$

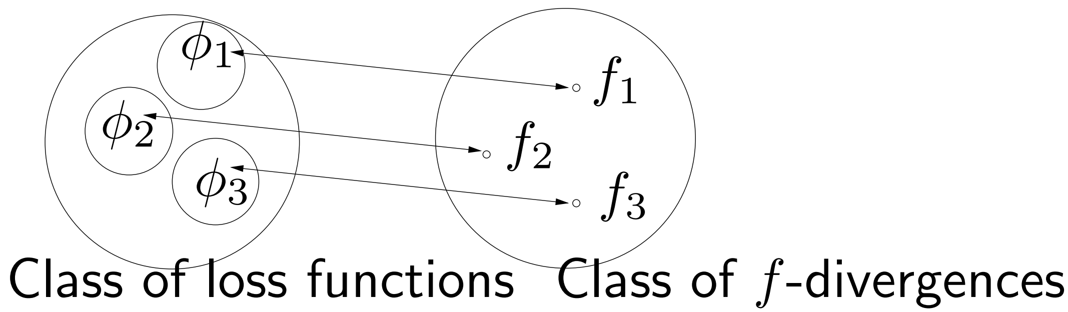
- Hellinger distance: $f(u) = \frac{1}{2}(\sqrt{u} - 1)^2$.

$$I_f(\mu, \pi) := \int_{z \in \mathcal{Z}} (\sqrt{\mu(z)} - \sqrt{\pi(z)})^2.$$

Surrogate loss and f-divergence



Surrogate loss and f-divergence



- Measures on Z associated with $Y = 1$ and $Y = -1$:

$$\begin{aligned} \mu(z) &:= P(Y = 1, z) \\ \pi(z) &:= P(Y = -1, z) \end{aligned}$$

- Fixing Q, define the optimal risk for each ϕ loss by optimizing over discriminant decision function γ:

$$R_\phi(Q) := \min_\gamma \mathbb{E}\phi(Y, \gamma(Z))$$

Link between ϕ-losses and f-divergences

Theorem: (Nguyen et al, 2009)

(a) For any surrogate loss ϕ, there is an f-divergence for some lower-semicontinuous convex f such that

$$R_\phi(Q) = -I_f(\mu, \pi).$$

- In addition, if ϕ is continuous and satisfies a (weak) regularity condition, f has to satisfy a number of conditions A.

(b) Conversely, if a convex f satisfies conditions A, there exists a convex surrogate loss ϕ that induces the corresponding f-divergence.

Link between ϕ-losses and f-divergences

Theorem: (Nguyen et al, 2009)

(a) For any surrogate loss ϕ, there is an f-divergence for some lower-semicontinuous convex f such that

$$R_\phi(Q) = -I_f(\mu, \pi).$$

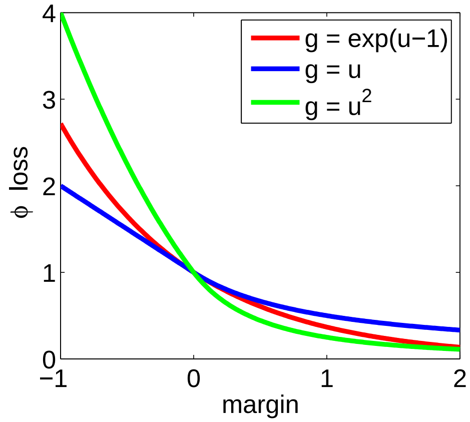
- In addition, if ϕ is continuous and satisfies a (weak) regularity condition, f has to satisfy a number of conditions A.

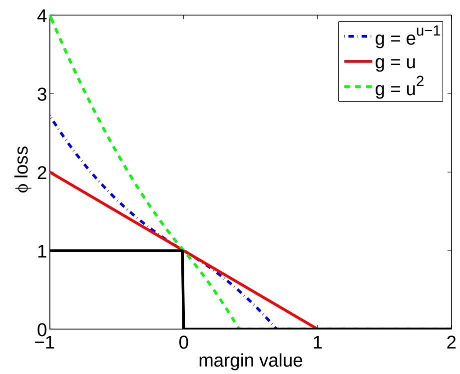
(b) Conversely, if a convex f satisfies conditions A, there exists a convex surrogate loss ϕ that induces the corresponding f-divergence.

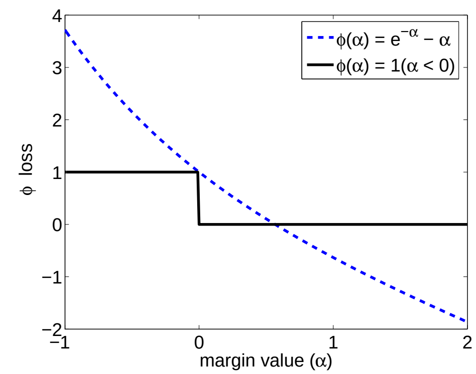
- the correspondence stems from a convex duality relationship
- we can construct *all* surrogate loss functions ϕ that induce the f-divergence
- ϕ is "parametrized" using the conjugate dual of f

Examples of surrogate losses for a given f-divergence

- Left: corresponding to Hellinger distance, including $\phi(\alpha) = \exp(-\alpha)$ (in boosting algorithm)







- Middle: corresponding to variational distance, including $\phi(\alpha) = (1-\alpha)_+$ (in support vector machine) and the 0-1 loss

- Right: corresponding to symmetric KL divergence, including $\phi(\alpha) = e^{-\alpha} - \alpha - 1$

A theory of equivalent surrogate loss functions

- two loss functions ϕ_1 and ϕ_2, corresponding to f-divergences induced by f_1 and f_2
- ϕ_1 and ϕ_2 are universally equivalent if for any $P(X,Y)$ and mapping rules Q_A, Q_B, there holds:

$$R_{\phi_1}(Q_A) \leq R_{\phi_1}(Q_B) \Leftrightarrow R_{\phi_2}(Q_A) \leq R_{\phi_2}(Q_B).$$

A theory of equivalent surrogate loss functions

- two loss functions ϕ_1 and ϕ_2, corresponding to f-divergences induced by f_1 and f_2
- ϕ_1 and ϕ_2 are universally equivalent if for any $P(X,Y)$ and mapping rules Q_A, Q_B, there holds:

$$R_{\phi_1}(Q_A) \leq R_{\phi_1}(Q_B) \Leftrightarrow R_{\phi_2}(Q_A) \leq R_{\phi_2}(Q_B).$$

- **Theorem 3:**

  ϕ_1 and ϕ_2 are universally equivalent if and only if

$$f_1(u) = cf_2(u) + au + b$$

  for constants $a, b \in \mathbb{R}$ and $c > 0$

- this result extends a theorem of Blackwell's, which is concerned only with f-divergences and the 0-1 loss, *not* the surrogate loss functions

Empirical risk minimization procedure

- let ϕ be a convex surrogate equivalent to $0-1$ loss
- $(\mathcal{C}_n, \mathcal{D}_n)$ is a sequence of increasing function classes for (γ, Q)
- given i.i.d. data pairs $(X_i, Y_i)_{i=1}^n$
- our procedure learns:

$$(\gamma_n^*, Q_n^*) := \operatorname{argmin}_{(\gamma,Q)\in(\mathcal{C}_n,\mathcal{D}_n)} \hat{\mathbb{E}}\phi(Y\gamma(Z))$$

- let $R_{bayes}^* := \inf_{(\gamma,Q)\in(\Gamma,\mathcal{Q})} P(Y \neq \gamma(Z))$ $\Leftarrow$ optimal Bayes error
- our procedure is Bayes-consistent if

$$R_{bayes}(\gamma_n^*, Q_n^*) - R_{bayes}^* \to 0$$

Bayes consistency

Theorem: If

- $\cup_{n=1}^{\infty}(\mathcal{C}_n, \mathcal{D}_n)$ is dense in the space of pairs of decision rules (γ, Q)

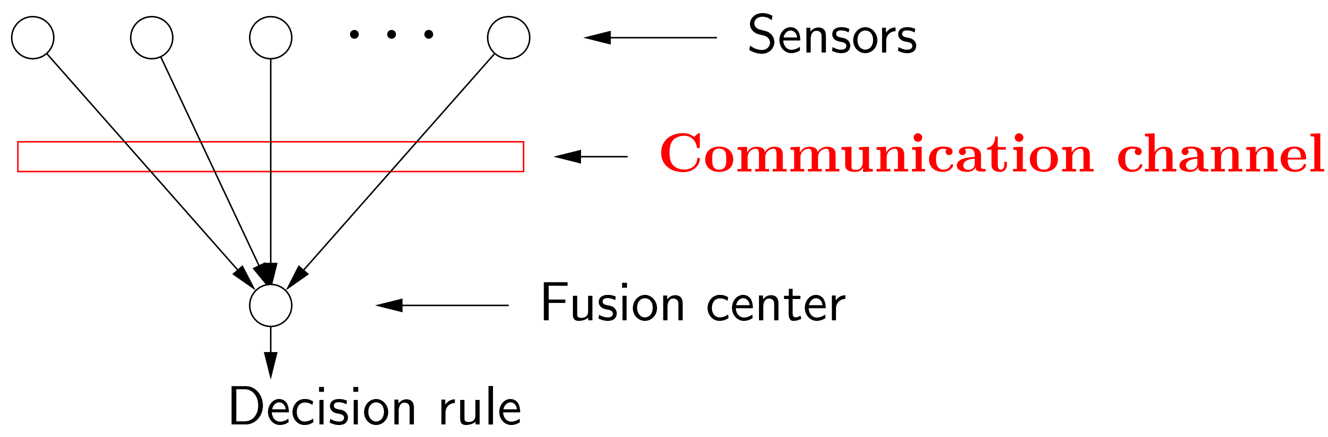
- sequence $(\mathcal{C}_n, \mathcal{D}_n)$ increases in size sufficiently slowly

then our procedure is consistent, i.e.,

$$\lim_{n \to \infty} R_{bayes}(\gamma_n^*, Q_n^*) - R_{bayes}^* = 0 \quad \text{in probability.}$$

- proof exploits the developed equivalence of ϕ loss and $0-1$ loss
- decomposition of ϕ risk into approximation error and estimation error

Brief summary



- **Joint estimation:** over the space of sensor messages, and over the space of classifier at the fusion center
  - subject to communication constraints

- **Challenges:**
  - the space of sensor messages is large, requiring better understanding of optimal messages
  - evaluation of risk function is hard, requiring approximation methods
  - underlying problem is non-convex, requiring clever "convexification"

Other formulations of aggregation in decentralized systems

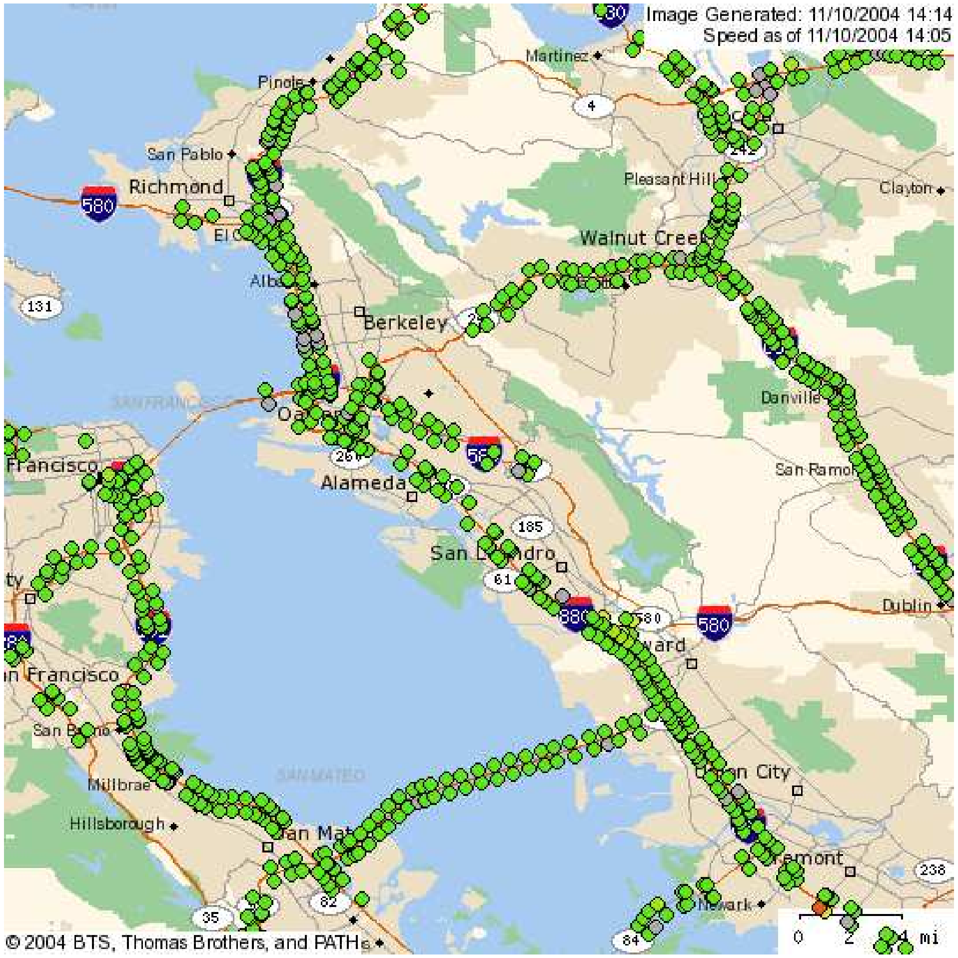
- moving from binary decision to multi-category decision (on-going work)
- accounting for sequential aspect of data (Nguyen et al, 2008)

Talk outline

- Set-up 1: decentralized detection (classification) problem
  - algorithmic and modeling ideas (marginalized kernels, convex optimization)
  - statistical properties (use of surrogate loss and f-divergence)

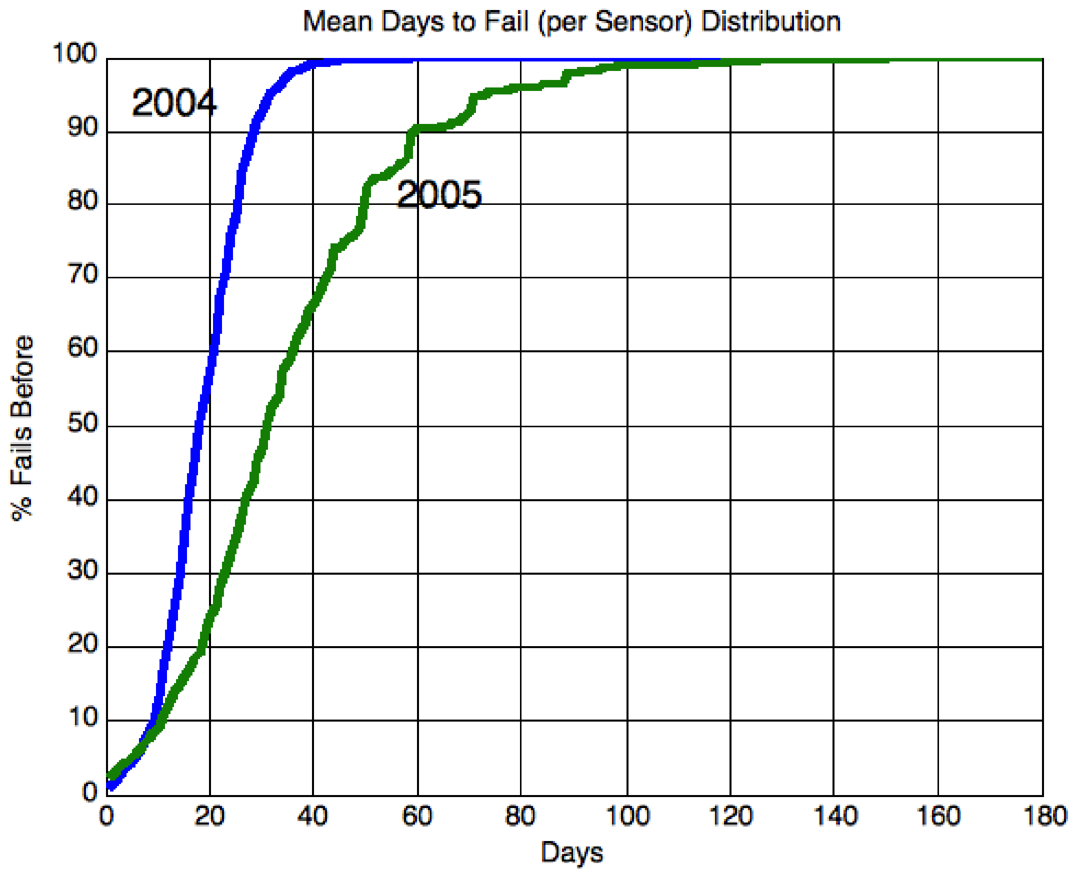
- Set-up 2: completely distributed decision-making for multiple sensors
  - algorithmic ideas (message-passing in graphical models)
  - statistical tools (from sequential analysis)

Failure detection for multiple sensors



traffic-measuring sensors placed along freeway network (Northern California)

Mean days to failure



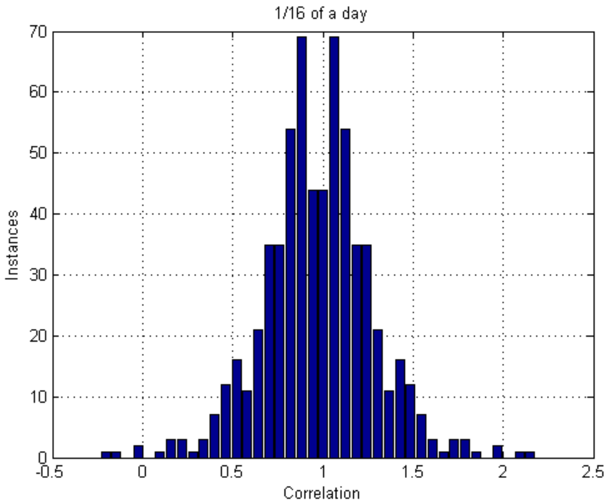
- as many as 40% sensors fail a given day
- separating sensor failure from events of interest is difficult
- “multiple change point detection” problem

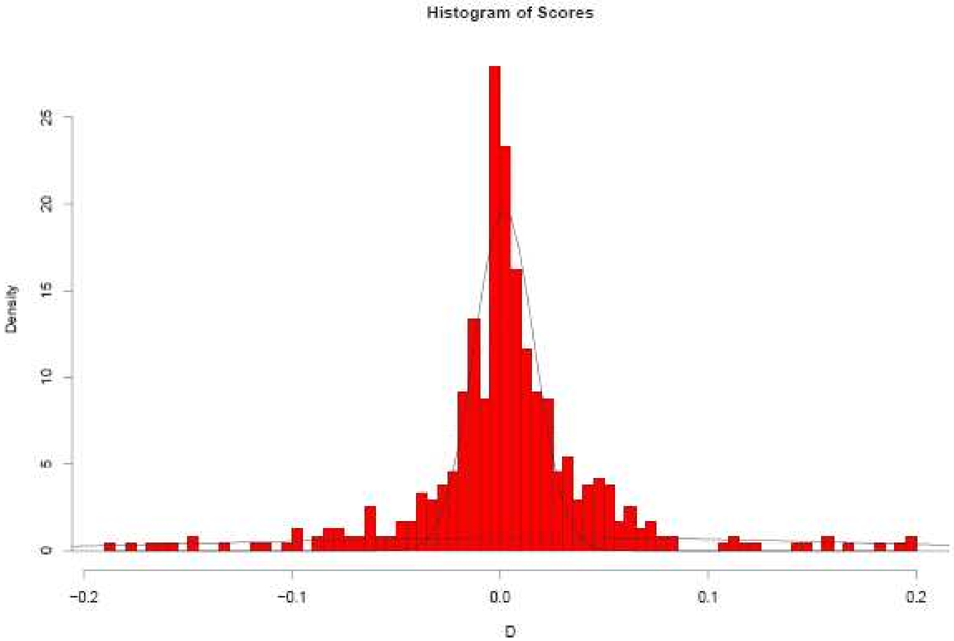
Set-up and underlying assumptions

- m sensors labeled by $U = \{u_1, \ldots, u_m\}$
- each sensor u receives sequence of data $X_t(u)$ for $t = 1, 2, \ldots$
- neighboring and functioning sensors have coorelated measurements
  - a failed sensor's measurement is not with its neighbors
- each sensor u fails at time $\lambda_u \sim \pi_u$
  - λ_u *a priori* are independent random variables
- correlation statistics $S_n(u, v)$ satisfies:

$$\begin{aligned} S_n(u,v) &\sim f_0(\cdot|u,v), iid \;\; n < \min(\lambda_u, \lambda_v) \\ &\sim f_1(\cdot|u,v), iid \;\; \text{otherwise} \end{aligned}$$

Distribution of correlation with neighbors

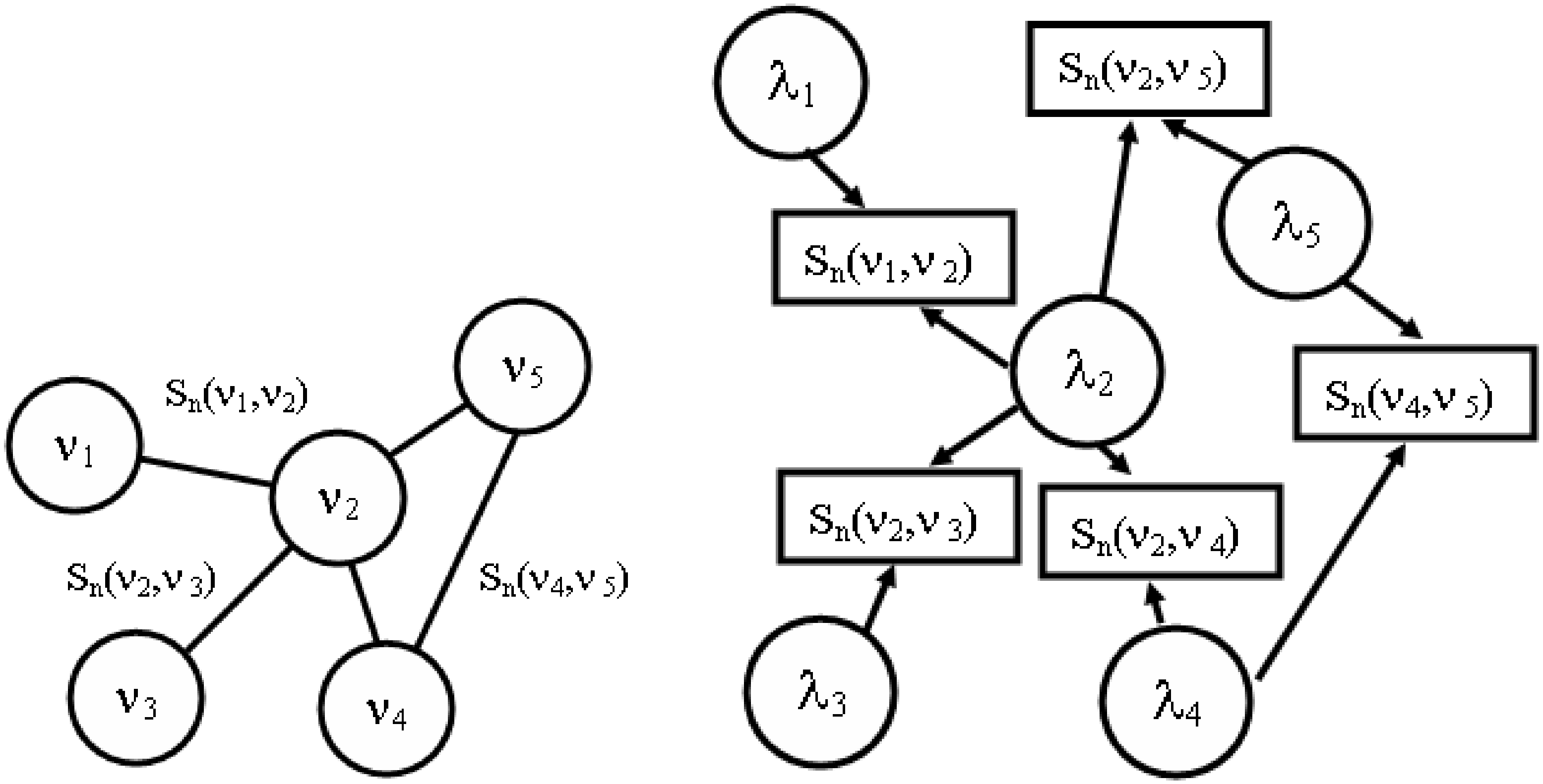




Left: A working sensor

Right: When failed

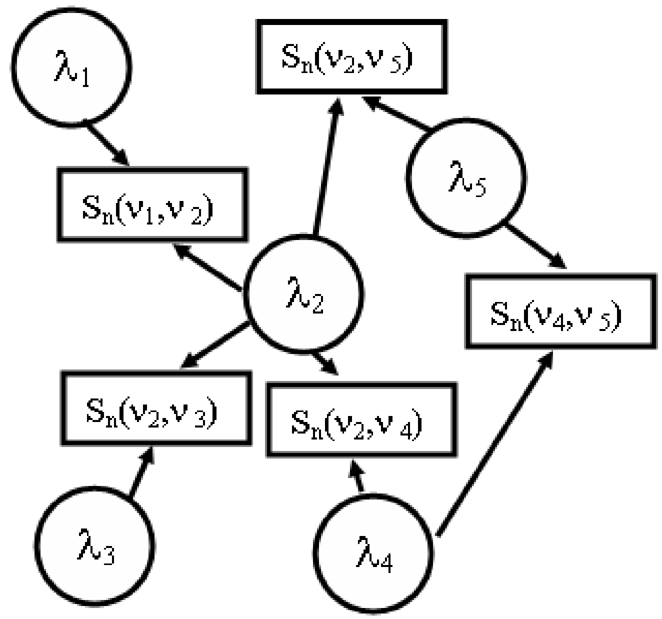
Graphical model of change points



Left: Dependency graph of sensors

Right: Graphical model of random variables

Detection rules are localized stopping rules



- detection rule for u, denoted by ν_u, is a *stopping time*, and depends on measurements of u and its neighbors
  - ν_u is a prediction of the "true" λ_u
- more precisely, for any $t > 0$:

$$\{\nu_u \leq t\} \in \sigma(\{S_n(u, u'), u' \in N(u), n \leq t\})$$

Performance metrics

- false alarm rate

$$PFA(\nu_u) = \mathbb{P}(\nu_u \leq \lambda_u).$$

- expected failure detection delay

$$D(\nu_u) = \mathbb{E}[\nu_u - \lambda_u | \nu_u \geq \lambda_u].$$

- problem formulation:

$$\min_{\nu_u} D(\nu_u) \text{ such that } PFA(\nu_u) \leq \alpha.$$

Single change point detection

(Shiryaev (1978), Tartakovski & Veeravalli(2005))

- optimal rule is to threshold the posterior of λ_u given data X

$$\nu_u(X) = \inf\{n : \Lambda_n > B_\alpha\},$$

where

$$\Lambda_n = \frac{\mathbb{P}(\lambda_u \leq n | X_1, \ldots, X_n)}{\mathbb{P}(\lambda_u > n | X_1, \ldots, X_n)}; \text{ and } B_\alpha = \frac{1-\alpha}{\alpha}.$$

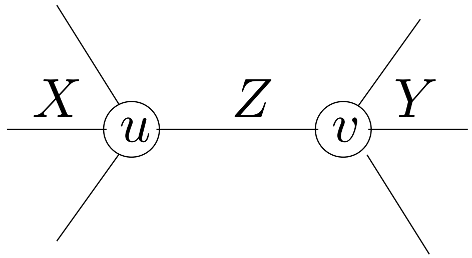
- this rule satisfies:

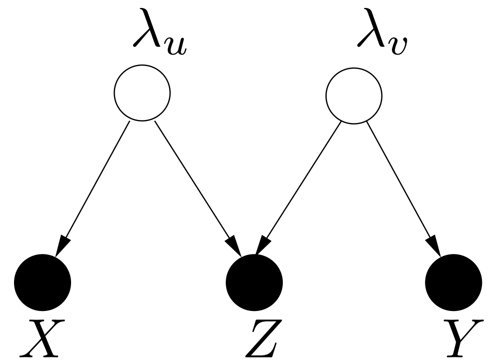
$$PFA(\nu_u(X)) \leq \alpha.$$

$$D(\nu_u(X)) \approx \frac{|\log \alpha|}{q_1(X) + d} \text{ as } \alpha \to 0.$$

where $q_1(X) = KL(f_1(X)||f_0(X))$, and d is the exponent of the a geometric prior on change point λ_u

Two sensors case: A naive extension





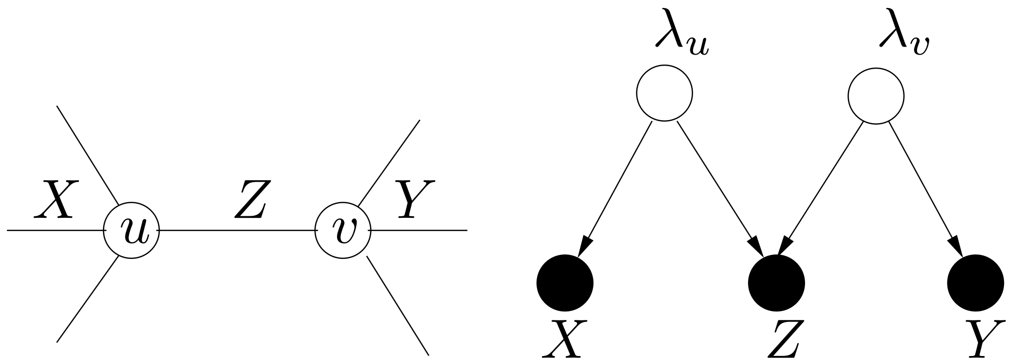
- Idea: conditioning on $X_1, \ldots, X_n$ and $Z_1, \ldots, Z_n$ to compute decision rule for u:

$$\nu_u(X, Z) \in \sigma(\{X, Z\}_1^n).$$

- Theorem: This approach does not help, i.e., no improvement in asymptotic delay time over the single change point approach:

$$\lim_{\alpha \to 0} D(\nu_u(X, Z)) = \lim_{\alpha \to 0} D(\nu_u(X)).$$

Two sensors case: A naive extension



- Idea: conditioning on $X_1, \ldots, X_n$ and $Z_1, \ldots, Z_n$ to compute decision rule for u:

$$\nu_u(X, Z) \in \sigma(\{X, Z\}_1^n).$$

- Theorem: This approach does not help, i.e., no improvement in asymptotic delay time over the single change point approach:

$$\lim_{\alpha \to 0} D(\nu_u(X, Z)) = \lim_{\alpha \to 0} D(\nu_u(X)).$$

- $\Rightarrow$ to predict λ_u, need to also use information given by Y

Localized stopping time with message exchange

- Main idea:
  - u should use information given by shared link Z *only if* its neighbor v is also functioning
- By combining with information given by Z, delay time is reduced:

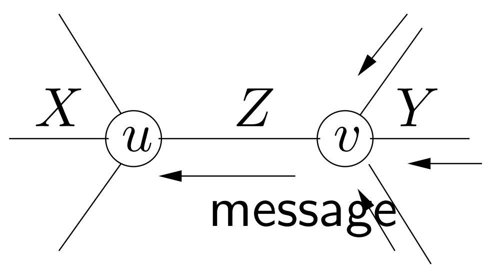
$$D(\nu_u(X)) \approx \frac{|\log \alpha|}{q_1(X) + d}$$

is strictly greater than

$$D(\nu_u(X, Z)) \approx \frac{|\log \alpha|}{q_1(X) + q_1(Z) + d}.$$

Localized stopping time with message exchange

- Main idea:
  - u should use information given by shared link Z *only if* its neighbor v is also functioning
  - but u never knows for sure if v works or fails, so...
  - u should use information given by shared link Z *only if* sensor u *thinks* neighbor v is also functioning
  - u thinks neighbor v is functioning if v thinks so, too, using information given by Z as well as Y



Continue...

- The protocol:
  - each sensor uses all links (variables) from sensors that are not yet declared to fail
  - if a sensor v raises a flag to declare that it fails, then v broadcasts this information to its neighbor(s), who promptly drop v from the list of their neighbors

Continue...

- The protocol:
  - each sensor uses all links (variables) from sensors that are not yet declared to fail
  - if a sensor v raises a flag to declare that it fails, then v broadcasts this information to its neighbor(s), who promptly drop v from the list of their neighbors
- Formally, for two sensors:
  - stopping rule for u, using only X: $\nu_u(X)$
  - stopping rule for u, using both X and Z: $\nu_u(X, Z)$
  - similarly, for sensor v: $\nu_v(Y)$ and $\nu_v(Y, Z)$
  - then, the overall rule for u is:

$$\bar{\nu}_u(X, Y, Z) = \nu_u(X, Z)\mathbb{I}(\nu_u(X, Z) \leq \nu_v(Y, Z)) + \max(\nu_u(X), \nu_v(Y, Z))\mathbb{I}(\nu_u(X, Z) > \nu_v(Y, Z)).$$

Performance bounds: theorem

(Rajagopal et al (2008))

- detection delay for u satisfies, for some constant $\delta_\alpha \in (0,1)$:

$$D(\bar{\nu}_u) \approx D(\nu_u(X,Z)(1-\delta_\alpha) + D(\nu_u(X))\delta_\alpha.$$

$\delta_\alpha =$ probability that u's neighbor declares "fail" before u

- for sufficiently small α there holds: $D(\bar{\nu}_u) < D(\nu_u(X))$

Performance bounds: theorem

(Rajagopal et al (2008))

- detection delay for u satisfies, for some constant $\delta_\alpha \in (0,1)$:

$$D(\bar{\nu}_u) \approx D(\nu_u(X,Z)(1-\delta_\alpha) + D(\nu_u(X))\delta_\alpha.$$

$\delta_\alpha =$ probability that u's neighbor declares "fail" before u

- for sufficiently small α there holds: $D(\bar{\nu}_u) < D(\nu_u(X))$
- false alarm rate for u satisfies:

$$PFA(\bar{\nu}_u) < 2\alpha + \xi(\bar{\nu}_u).$$

- $\xi(\bar{\nu}_u)$ is termed confusion probability: probability that u thinks v has not failed, while in fact, v already has:

$$\xi(\bar{\nu}_u) = \mathbb{P}(\bar{\nu}_u \leq \bar{\nu}_v, \lambda_v \leq \bar{\nu}_u \leq \lambda_u).$$

Performance bounds: theorem

(Rajagopal et al (2008))

- detection delay for u satisfies, for some constant $\delta_\alpha \in (0,1)$:

$$D(\bar{\nu}_u) \approx D(\nu_u(X,Z)(1-\delta_\alpha) + D(\nu_u(X))\delta_\alpha.$$

$\delta_\alpha =$ probability that u's neighbor declares "fail" before u

- for sufficiently small α there holds: $D(\bar{\nu}_u) < D(\nu_u(X))$
- false alarm rate for u satisfies:

$$PFA(\bar{\nu}_u) < 2\alpha + \xi(\bar{\nu}_u).$$

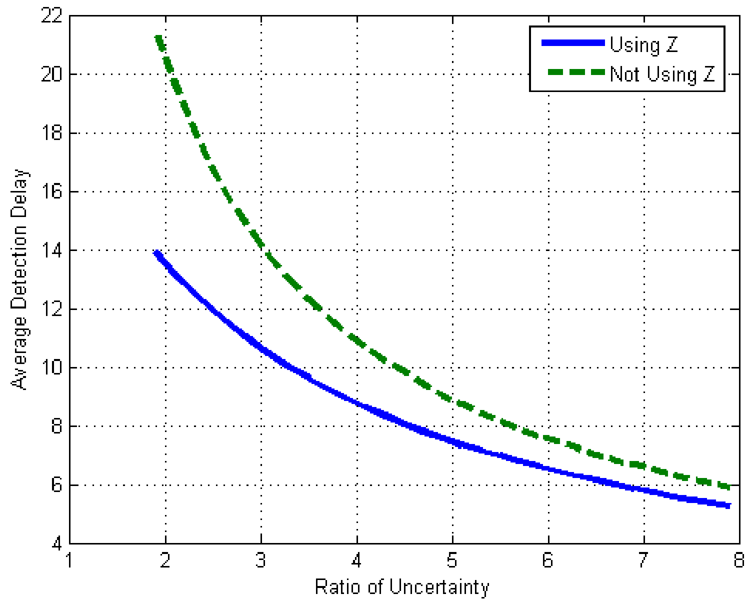
- $\xi(\bar{\nu}_u)$ is termed confusion probability: probability that u thinks v has not failed, while in fact, v already has:

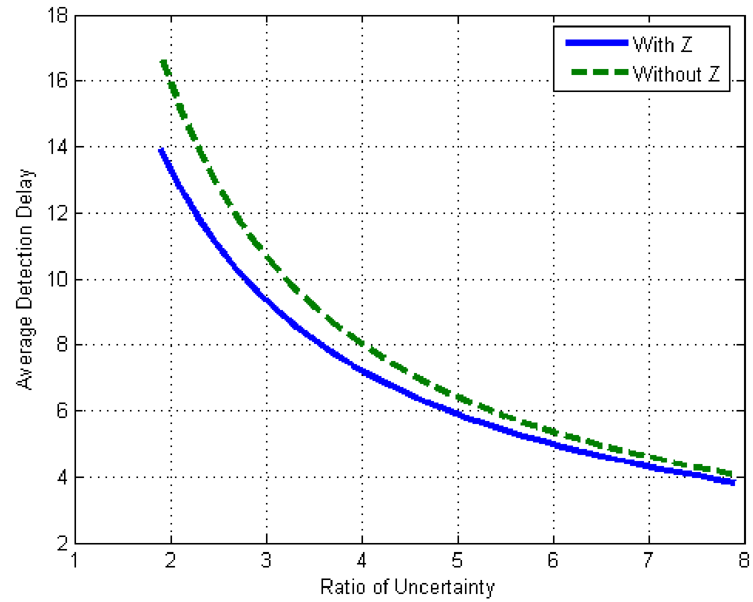
$$\xi(\bar{\nu}_u) = \mathbb{P}(\bar{\nu}_u \leq \bar{\nu}_v, \lambda_v \leq \bar{\nu}_u \leq \lambda_u).$$

- under certain conditions, $\xi(\bar{\nu}_u) = O(\alpha)$.

Effects of message passing

Two-sensor network:





X-axis: Ratio of informations $q_1(X)/q_1(Z)$

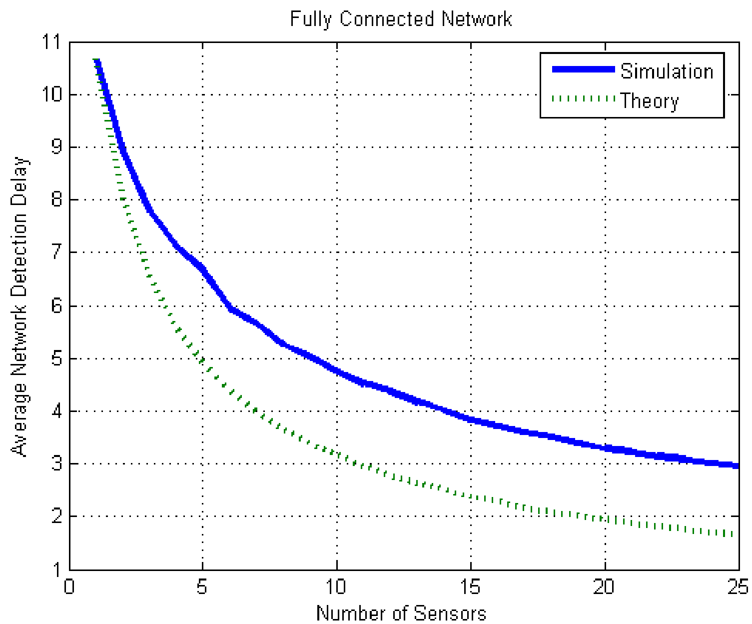
Y-axis: Detection delay time

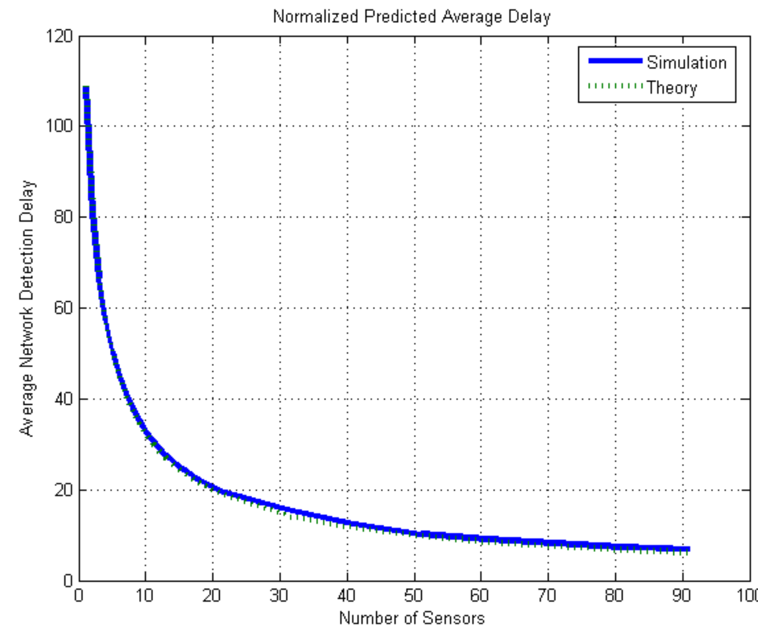
Left: evaluated by simulations

Right: predicted by our theory

Number of sensors vs Detection delay time

Fully connected network:



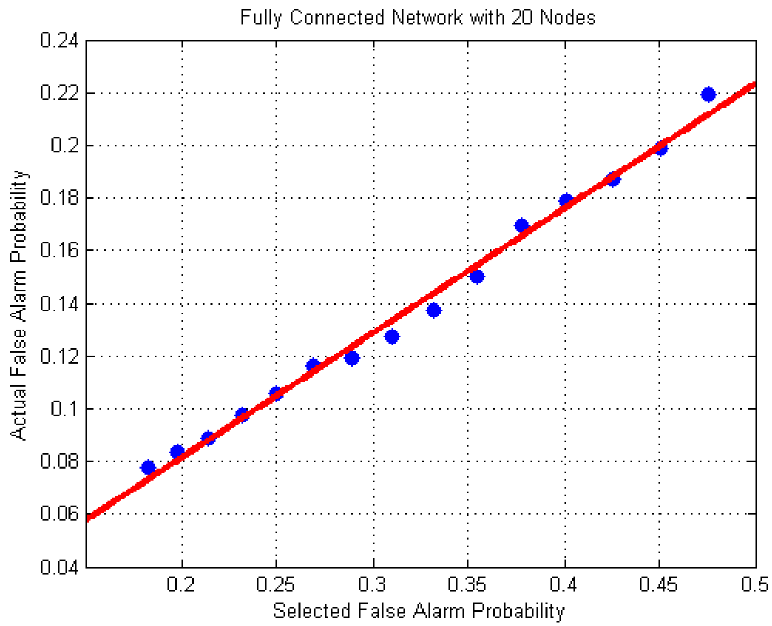


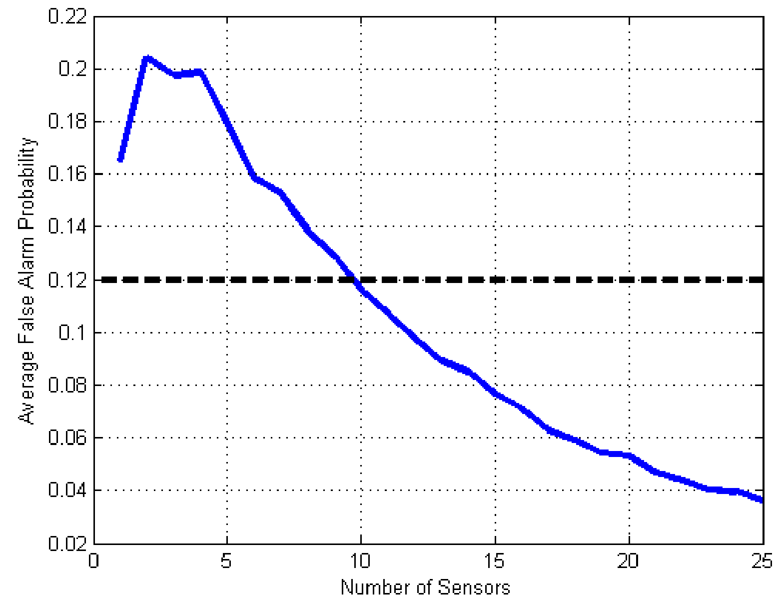
Left: $\alpha = .1$

Right: $\alpha = 10^{-4}$ (theory predicts well!)

False alarm rates

Fully connected network:



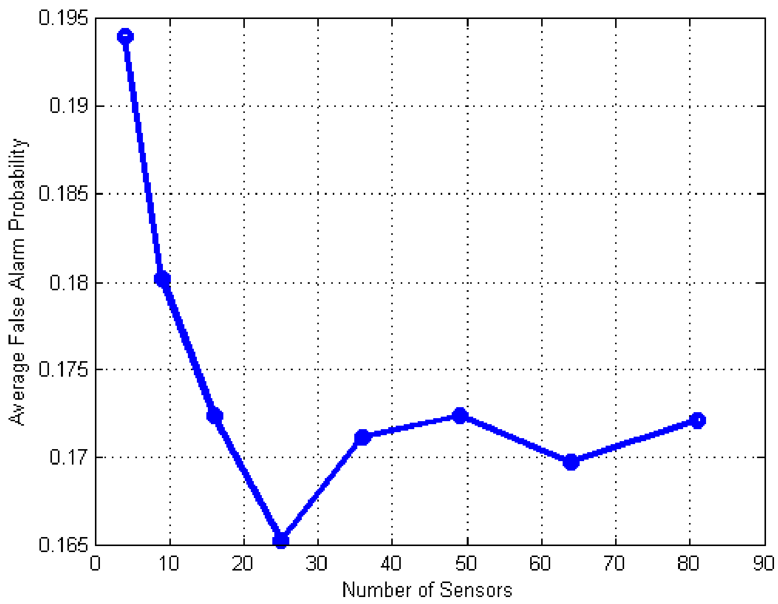


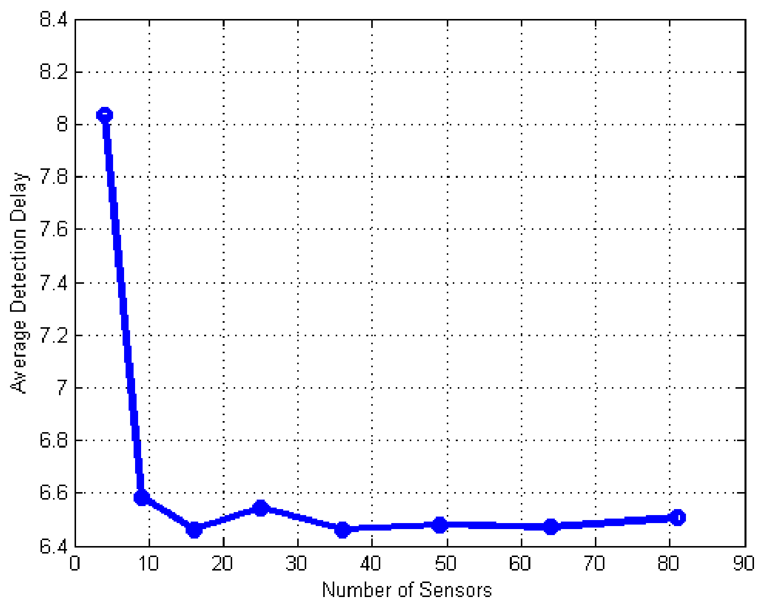
Left: Selected false alarm rate vs. actual rate

Right: Number of sensors vs. actual rate

Effects of network topology (and spatial dependency)

Grid network





Left: Number of sensors vs. actual false alarm rate

Right: Number of sensors vs. actual detection delay

Summary

- aggregation of data to make a good decision toward the same goal
  - how to learn jointly local messages and global detection decision
  - subject to the distributed constraints of system?

- decision-making with multiple and spatially dependent goals
  - how to devise efficient message passing schemes that utilize statistical dependency?

- tools from convex optimization, statistical modeling and asymptotic analysis