

Learning marginalized kernels and decentralized detection problems

XuanLong Nguyen
U.C. Berkeley

Joint work with Martin J. Wainwright and Michael I. Jordan
Division of Computer Science and Department of Statistics
U.C. Berkeley

Motivation

- **Marginalized kernels** have been used to derive similarity measures for classification of structured data (Jaakkola et al 1998, Tsuda et al, 2002)
 - exploiting prior knowledge provided by probabilistic graphical models
- Suppose $X \sim P(X, H|\theta)$, where H is a latent random vector
- Knowing both X and H provides a natural *base kernel* function $K_b((x, h), (x', h'))$
- Marginalized kernel is defined by **marginalizing** over hidden H, H'

$$\begin{aligned} K(x, x') &= \mathbb{E}[K_b((x, H), (x', H'))|x, x'] \\ &= \sum_{h, h'} K_b((x, H), (x', H'))P(h|x, \theta)P(h'|x', \theta) \end{aligned}$$

Motivation

How is $P(X, H|\theta)$ estimated in the first place?

There are two options

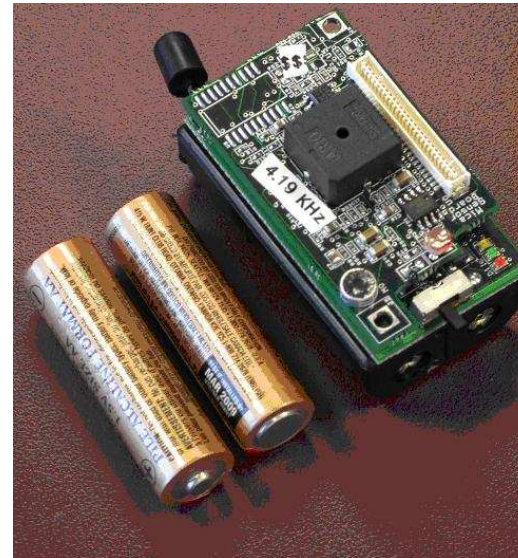
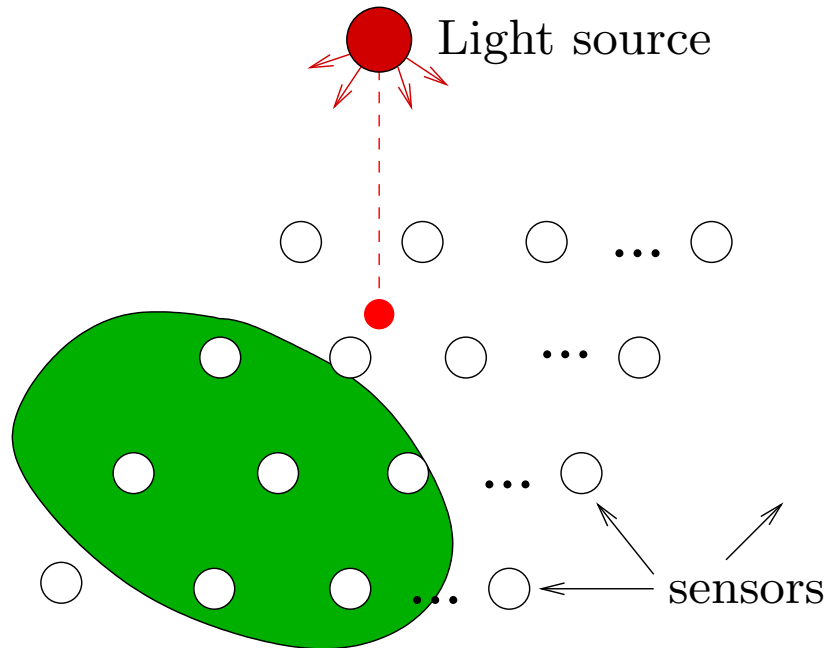
- learning θ from (unlabeled) data sample X by MLE
- **kernel learning approach**
 - only $P(H|X)$ needs to be learned
 - learning by directly minimizing directly the classification error

Decentralized detection problem can be reduced to a problem of *learning marginalized kernels*

Talk outline

- Problem of decentralized detection with sensor networks
 - classification under communication constraints
- Decentralized detection as (parametrized) marginalized kernel learning
 - optimization issues
 - generalization error bound issues

Detection (classification) with tiny sensors



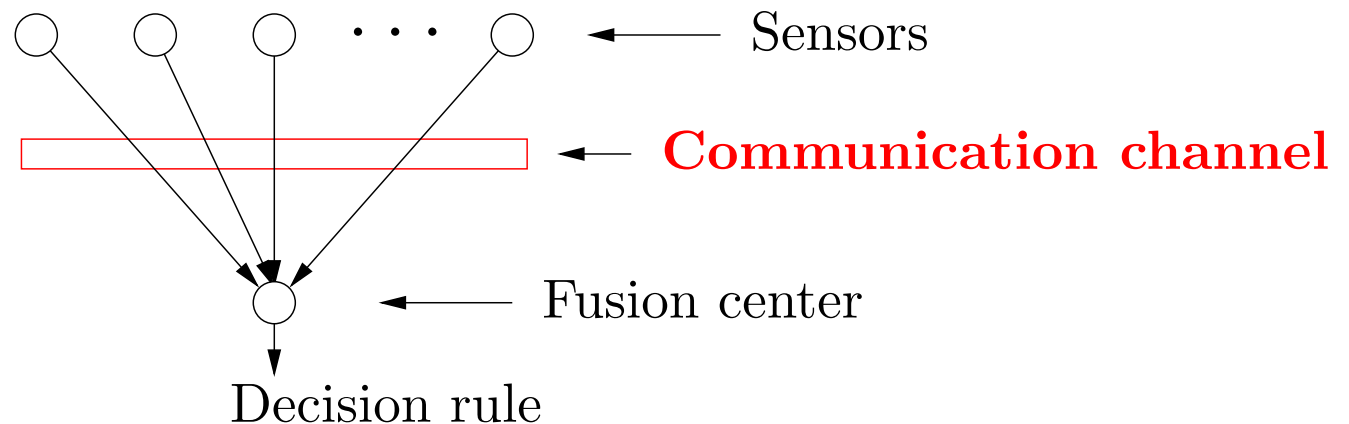
Set-up:

- Wireless network of tiny sensor motes, each equipped with a light receiver
- Measurement of light strength ($[0-1024]$ in magnitude, or 10 bits)

Goal: Determine position of light source relative to the green region.

I.e., is light source *inside* or *outside*?

Decentralized classification



- **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 classification*

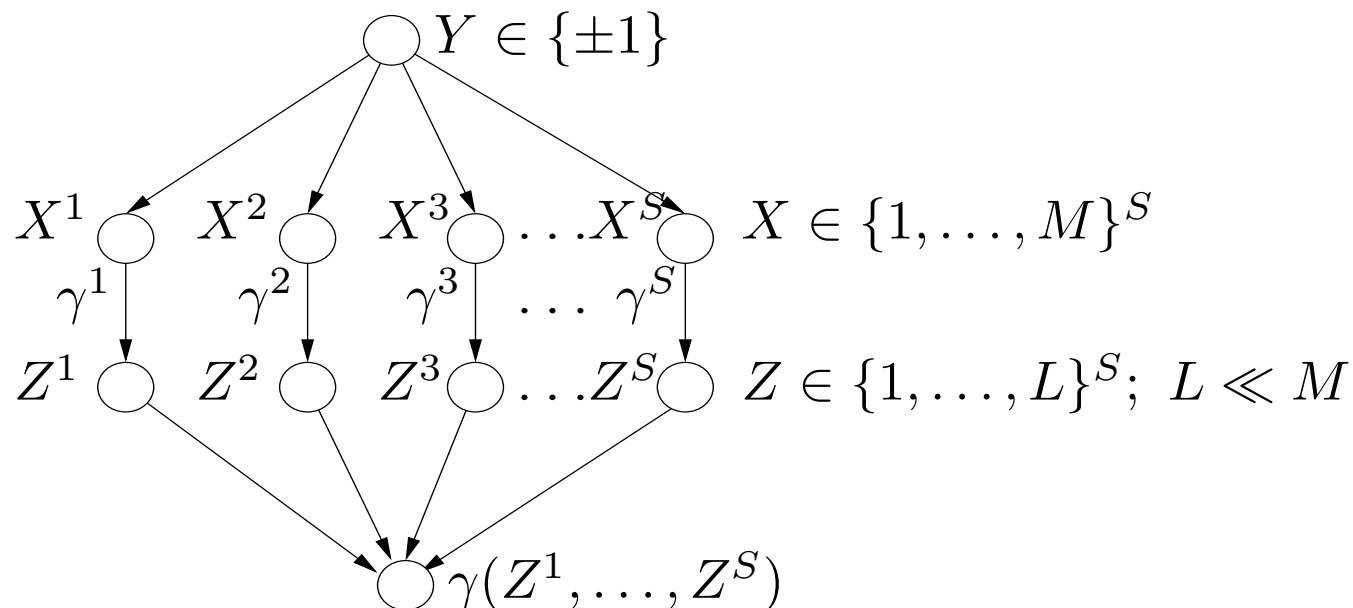
Related work

- Classical work on classification:
 - completely centralized
 - no consideration of communication-theoretic infrastructure
- Decentralized detection in signal processing (e.g., Tsitsiklis, 1993)
 - joint distribution assumed to be known
 - locally-optimal rules under conditional independence assumptions (i.e., Naive Bayes)

Overview of our approach

- Treat as a learning problem
 - under constraints from a distributed system
- Use **kernel methods**
 - tools from convex optimization to derive an efficient algorithm
- Exploit the notion of a **marginalized kernel**
 - arises naturally from the marginalization of the sensor message conditionally on the sensor signal

Problem set-up



Problem: Given training data $(x_i, y_i)_{i=1}^n$, find the decision rules $(\gamma^1, \dots, \gamma^S; \gamma)$ so as to minimize the **misclassification probability**:

$$P(Y \neq \gamma(Z^1, \dots, Z^S)).$$

Kernel methods for classification

- $K(x, x')$ is a *symmetric positive semidefinite* kernel function
- *feature space* \mathcal{H} in which K acts as an inner product, i.e., $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$
- Kernel-based algorithm finds **linear function** in \mathcal{H} , i.e.

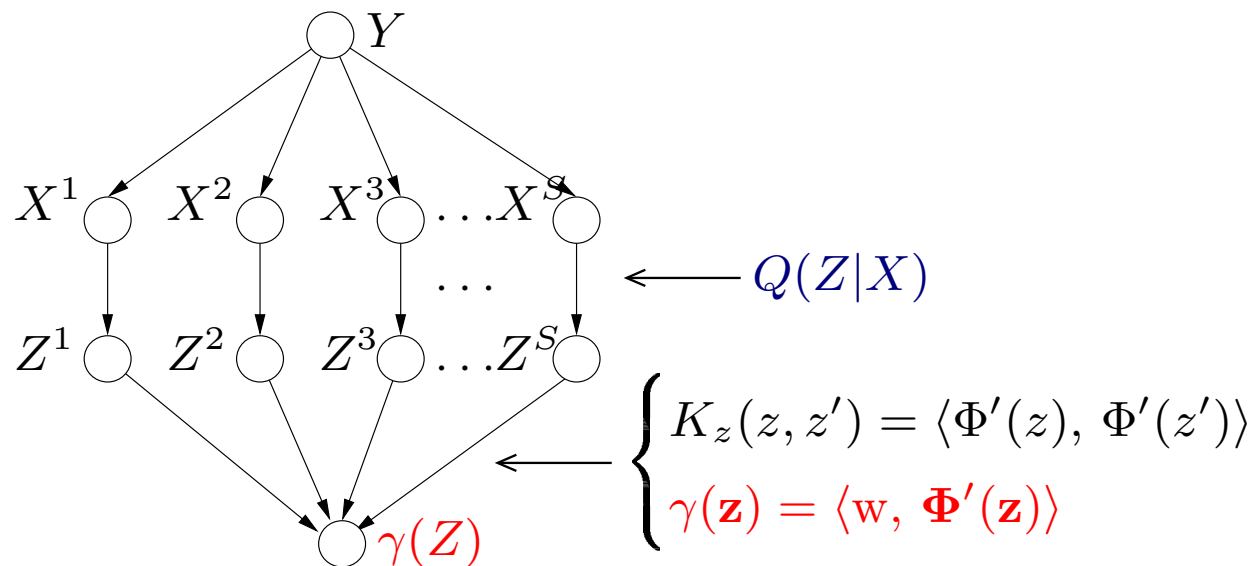
$$f(x) = \langle \mathbf{w}, \Phi(x) \rangle,$$

by minimizing empirical ϕ -risk:

$$\min_{\mathbf{w}} \sum_{i=1}^n \phi(y_i f(x_i)) + \frac{\lambda}{2} \|\mathbf{w}\|^2,$$

- $(x_i, y_i)_{i=1}^n$ are training data in $\mathcal{X} \times \{\pm 1\}$
- 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 \implies factorization:

$$Q(Z|X) = \prod_{t=1}^S Q^t(Z^t|X^t)$$

- The overall decision rule is represented by $\left\{ \begin{array}{l} \mathbf{Q} = \prod \mathbf{Q}^t, \\ \gamma(\mathbf{z}) = \langle \mathbf{w}, \Phi'(\mathbf{z}) \rangle \end{array} \right.$

The trick: Reduce to kernel learning

- Define a marginalized kernel $K_Q(x, x')$ over \mathcal{X} :

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

- 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)
- Given training sample $(x_i, y_i)_{i=1}^n$, learn an optimal discriminant function $f_Q(x) = \langle w, \Phi_Q(x) \rangle$ in K_Q -induced feature space
 - **optimization is done over both w and Q**

Marginalized kernels: Examples

- For first-order count kernel $K_z(z, z') := \sum_{t=1}^S \mathbb{I}[z^t = z'^t]$, the corresponding marginalized kernel takes the form:

$$\begin{aligned} K_Q(x, x') &= \sum_{z, z'} Q(z|x)Q(z'|x') \sum_{t=1}^S \mathbb{I}[z^t = z'^t] \\ &= \sum_{z, z'} \prod_{t=1}^S Q^t(z^t|x^t)Q(z'^t|x'^t) \sum_{t=1}^S \mathbb{I}[z^t = z'^t] \\ &= \sum_{t=1}^S P(z^t = z'^t|x^t, x'^t) \end{aligned}$$

- Similarly, we can define higher-order marginalized kernels

Kernel-based decision rule representation

- Optimal weight vector w and centralized function obtained by minimizing ϕ -risk:

$$f_Q(x) = \langle w, \Phi_Q(x) \rangle$$

- Optimal w also defines decision function for fusion center:

$$\gamma(z) = \langle w, \Phi'(z) \rangle$$

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

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

- **Centralized** rule f_Q : direct access to sensor signal x
- **Decentralized** rule at fusion center: only sees quantized version z

What was going on: Minimizing empirical ϕ -risk

- The regularized empirical ϕ -risk has the form:

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

- **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

Approximating empirical ϕ -risk

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

$$\begin{cases} \Phi_Q(x) = \sum_z Q(z|x)\Phi'(z) & \Leftarrow \text{Marginalization over } z \\ f_Q(x) = \langle w, \Phi_Q(x) \rangle \end{cases}$$

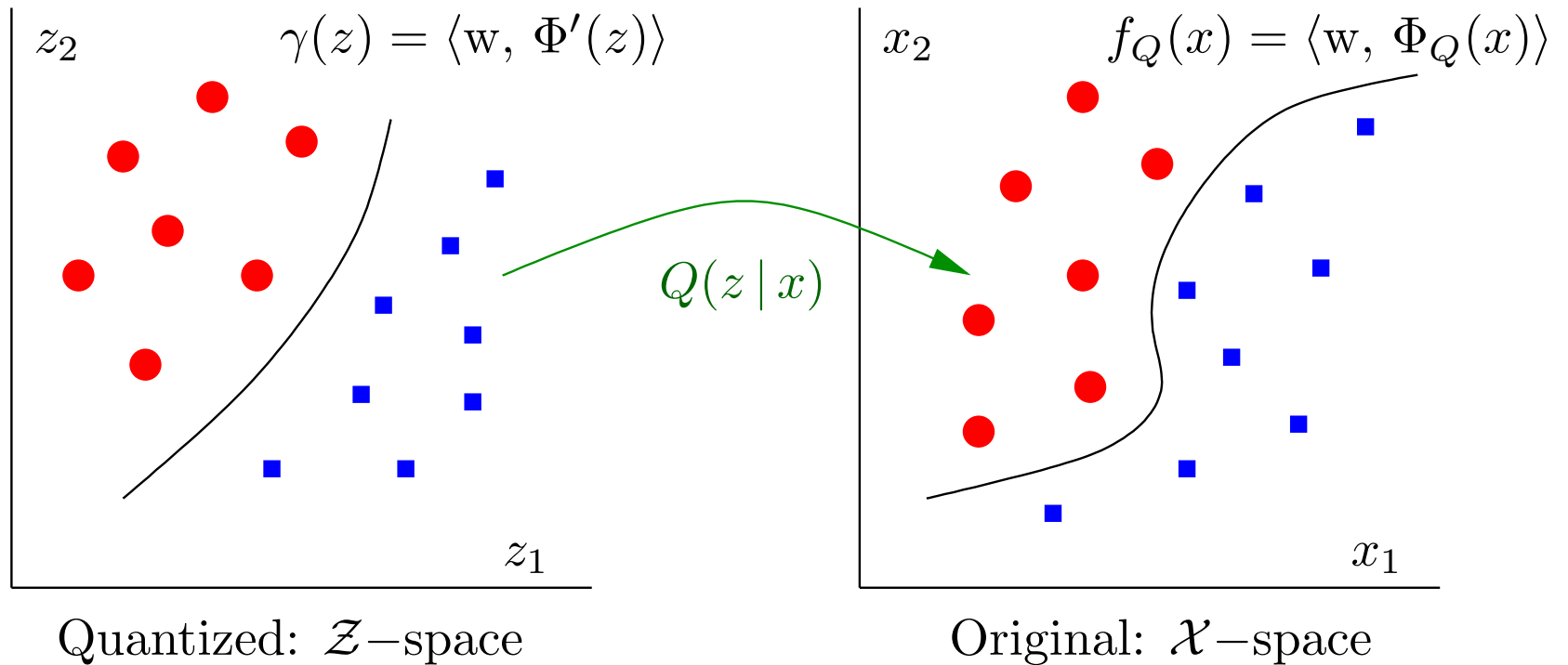
- The alternative objective function G is the ϕ -risk for f_Q :

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

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

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

Geometric illustration



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

- maps between \mathcal{X} and \mathcal{Z}
- induces marginalized feature map Φ_Q from base map Φ' (or marginalized kernel K_Q from base kernel K)

Approximating empirical ϕ -risk

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

$$\begin{cases} \Phi_Q(x) = \sum_z Q(z|x)\Phi'(z) & \Leftarrow \text{Marginalization over } z \\ f_Q(x) = \langle w, \Phi_Q(x) \rangle \end{cases}$$

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

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

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

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

Optimization algorithm

Goal: Solve the problem:

$$\inf_{\mathbf{w}; Q} G(\mathbf{w}; Q) := \frac{1}{\lambda} \sum_i \phi \left(y_i \langle \mathbf{w}, \sum_z Q(z|x_i) \Phi'(z) \rangle \right) + \frac{1}{2} \|\mathbf{w}\|^2$$

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

Overall: Efficient joint minimization by blockwise coordinate descent

Kernel Quantization (KQ) algorithm

- (a) With Q fixed, compute the optimizing $w(Q)$ by solving the dual problem for the primal problem $\inf_w G$.
- (b) For some index t , fix $w(Q)$ and $\{Q^r, r \neq t\}$ and take a gradient step in Q^t .

Upon convergence, a deterministic decision rule for each sensor t is defined via:

$$\gamma^t(x^t) := \operatorname{argmax}_{z^t \in \mathcal{Z}} Q(z^t | x^t).$$

Optimization of w given fixed Q

- For each fixed $Q \in \mathcal{Q}$, the value of the primal problem $\inf_w G(w; Q)$ is attained and equal to its dual form:

$$\max_{\alpha \in \mathbb{R}^n} \left\{ -\frac{1}{\lambda} \sum_{i=1}^n \phi^*(-\lambda \alpha_i) - \frac{1}{2} \sum_{1 \leq i, j \leq n} \alpha_i \alpha_j y_i y_j K_Q(x_i, x_j) \right\}, \quad (1)$$

Furthermore, any optimal solution α to problem (1) defines the optimal primal solution $w(Q)$ to $\min_w G(w; Q)$ via $w(Q) = \sum_{i=1}^n \alpha_i y_i \Phi_Q(x_i)$.

Notations: $\phi^*(u) := \sup_{v \in \mathbb{R}} \{u \cdot v - \phi(v)\}$ is the conjugate dual of ϕ .

Optimization of $Q^t(z^t|x^t)$

- Our approach is to compute the gradient for G w.r.t. Q^t , and then apply a gradient-based method.
- One challenge is that G may not be differentiable (i.e. hinge-loss).
- Moreover, G is defined in terms of feature vectors $\Phi'(z)$, which are typically large or infinite-dimensional quantities.
- Using tools from convex duality, we show that at least one subgradient for G evaluated at the optimal argument $(w(Q), Q)$ of the dual problem, can be computed efficiently.

Optimization of $Q^t(z^t|x^t)$

- Let $w(Q)$ be the optimizing argument of $\min_w G(w; Q)$, and let α be an optimal solution to the dual problem (1). Then the following element

$$-\lambda \sum_{(i,j)(z,z')} \alpha_i \alpha_j Q(z'|x_j) \frac{Q(z|x_i)}{Q^t(z^t|x_i^t)} K_z(z, z') \mathbb{I}[x_i^t = \bar{x}^t] \mathbb{I}[z^t = \bar{z}^t]$$

is an element of the subdifferential $\partial_{Q^t(\bar{z}^t|\bar{x}^t)} G$ evaluated at $(w(Q), Q)$.

- This subgradient can be computed efficiently in the same way a marginalized kernel is computed.

Estimation error bounds

- Our algorithm finds a function f_Q that minimizes *empirical* risk:

$$\hat{\mathbb{E}}\phi(Y f_Q(x)) = \sum_{i=1}^n \phi(y_i f_Q(x_i))$$

- But, we want to find decision rule (Q, γ) so that the risk for γ is minimized:

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

- **General question:** Relate $\mathbb{E}\phi(Y \gamma(Z))$ to $\hat{\mathbb{E}}\phi(Y f_Q(X))$ in terms of number samples n , and sensor network parameters
 - Differs from traditional issue of ϕ risk versus empirical ϕ risk for the *same* function

Theorem: Estimation error bounds

- Define

$$\begin{cases} \mathcal{F} := \{f_Q : x \mapsto \langle w, \Phi_Q(x) \rangle \text{ for all } Q\} \\ \mathcal{F}_0 := \mathcal{F} \text{ restricted to deterministic } Q \end{cases}$$

- Let $R_n(\mathcal{F})$ and $R_n(\mathcal{F}_0)$ be Rademacher complexity for \mathcal{F} and \mathcal{F}_0 , respectively. Then,
- Given n i.i.d. labeled data points $(x_i, y_i)_{i=1}^n$, with probability at least $1 - 2\delta$,

$$\begin{aligned} \inf_{f_Q \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \phi(y_i f_Q(x_i)) - 2\ell R_n(\mathcal{F}) - \sqrt{\frac{\ln(2/\delta)}{2n}} \\ \leq \inf_{f \in \mathcal{F}} \mathbb{E} \phi(Y \gamma(Z)) \leq \\ \inf_{f_Q \in \mathcal{F}_0} \frac{1}{n} \sum_{i=1}^n \phi(y_i f_Q(x_i)) + 2\ell R_n(\mathcal{F}_0) + \sqrt{\frac{\ln(2/\delta)}{2n}} \end{aligned}$$

Bounds on Rademacher complexity

Sensor network parameters:

- n \equiv number of observation samples
- L \equiv number of quantization levels
- M \equiv number of signal levels
- S \equiv number of sensors

Bounds:

- $R_n(\mathcal{F})$ decreases at rate $O(1/\sqrt{n})$,
- $R_n(\mathcal{F})$ increases at rate $O([MS^2 L \log L]^{\frac{1}{2}})$

Bounding Rademacher complexity for \mathcal{F}_0

- For \mathcal{F}_0 , we have:

$$R_n(\mathcal{F}_0) \leq \frac{2B}{n} \left[\mathbb{E} \sup_{Q \in \mathcal{Q}_0} \sum_{i=1}^n K_Q(X_i, X_i) + 2(n-1) \sqrt{n/2} \sup_{z, z'} K_z(z, z') \sqrt{2MS \log L} \right]^{1/2}$$

Bounding Rademacher complexity for \mathcal{F}

- Step 1: Bounding $R_n(\mathcal{F})$ in terms of the covering number:

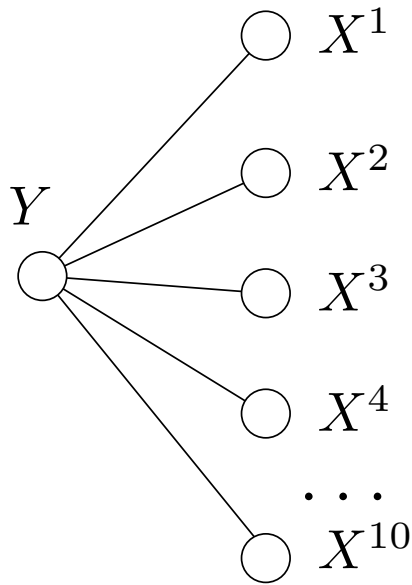
$$\widehat{R}_n(\mathcal{F}) \leq C \int_0^\infty \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon$$

- Step 2: Relating the covering number for \mathcal{F} to S, M, L : $\log N(\epsilon, \mathcal{F}, L_2(P_n))$ of \mathcal{F} is bounded above by

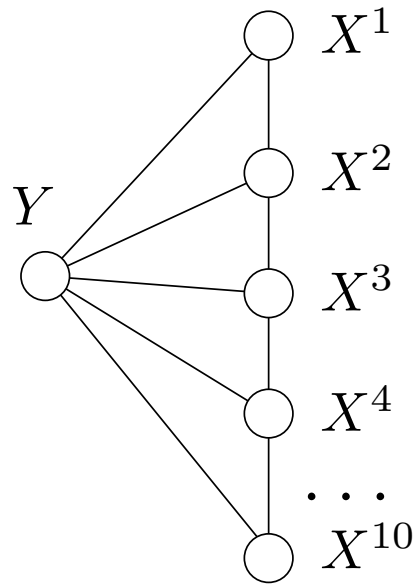
$$\sup_{Q \in \mathcal{Q}} \log N(\epsilon/2, \mathcal{F}_Q, L_2(P_n)) + (L-1)MS \log \frac{2L^S \sup \|\alpha\|_1 \sup_{z, z'} K_z(z, z')}{\epsilon}$$

- **In summary**, $R_n(\mathcal{F})$ increases with rate $O([MS^2L \log L]^{\frac{1}{2}})$ and decreases with rate $O(1/\sqrt{n})$

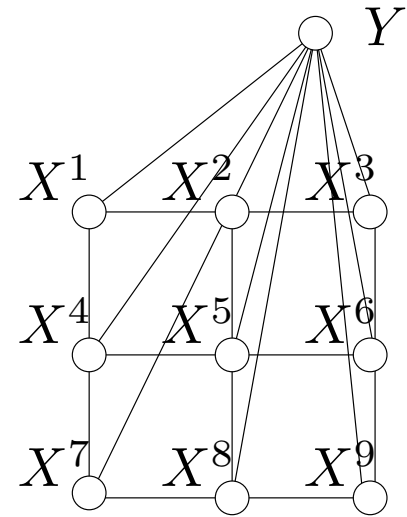
Simulated sensor networks



Naive Bayes net



Chain-structured network

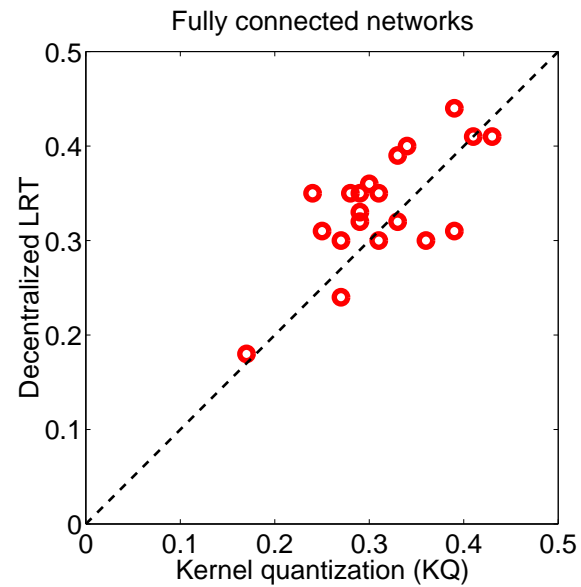
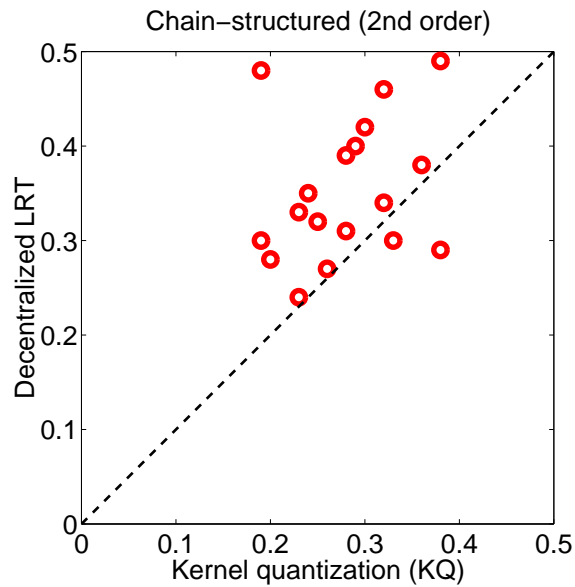
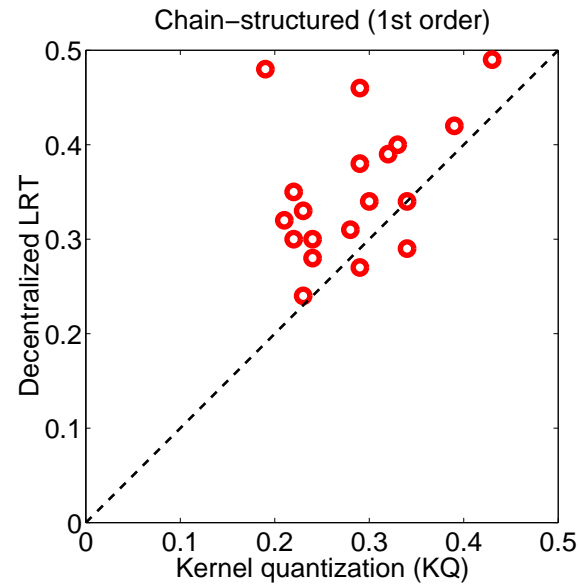
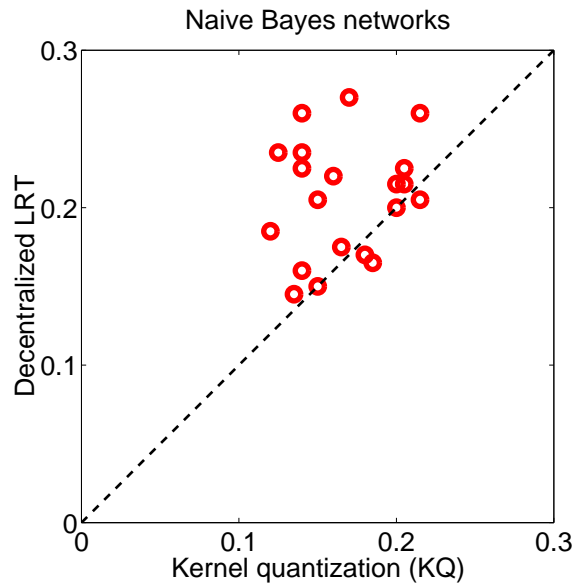


Spatially-dependent network

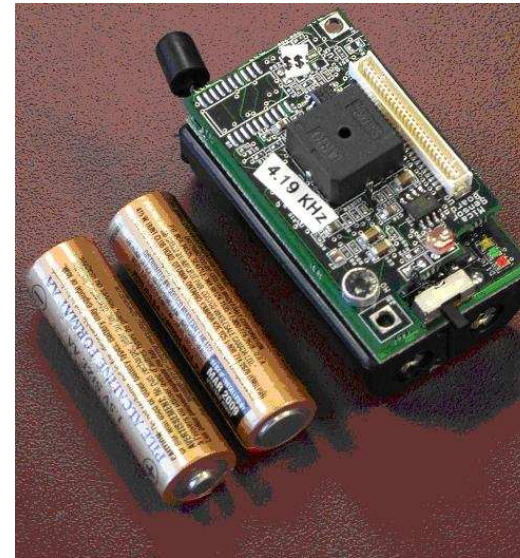
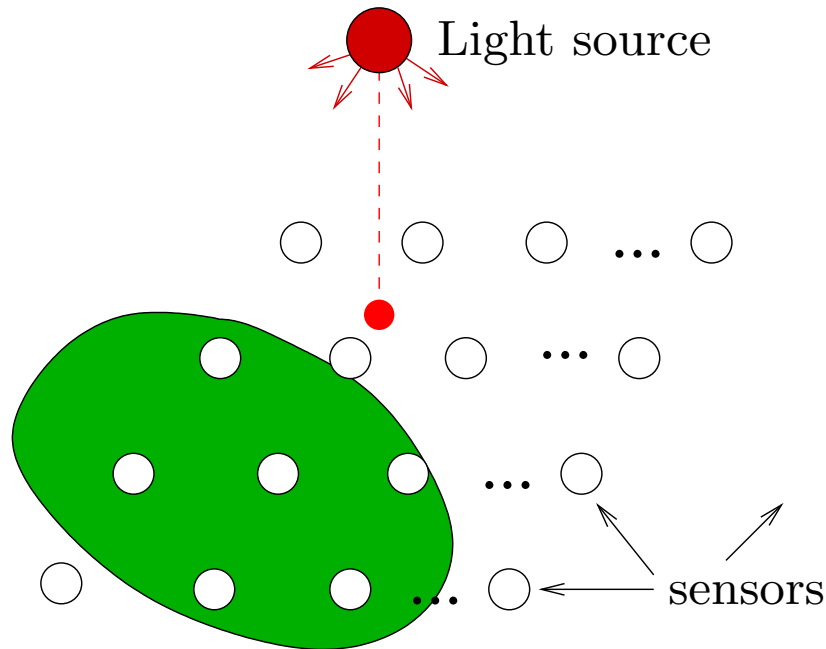
Experimental procedure

- For each network type, generate 20 random instances
- For each random instance:
 - Generate 400 observations
 - Each observation takes 8 discrete values
 - Goal is to find a L -level quantization scheme, for $L = 2, 4, 6$
- Compare to a *decentralized* likelihood ratio test (LRT)

Kernel Quantization vs. Decentralized LRT

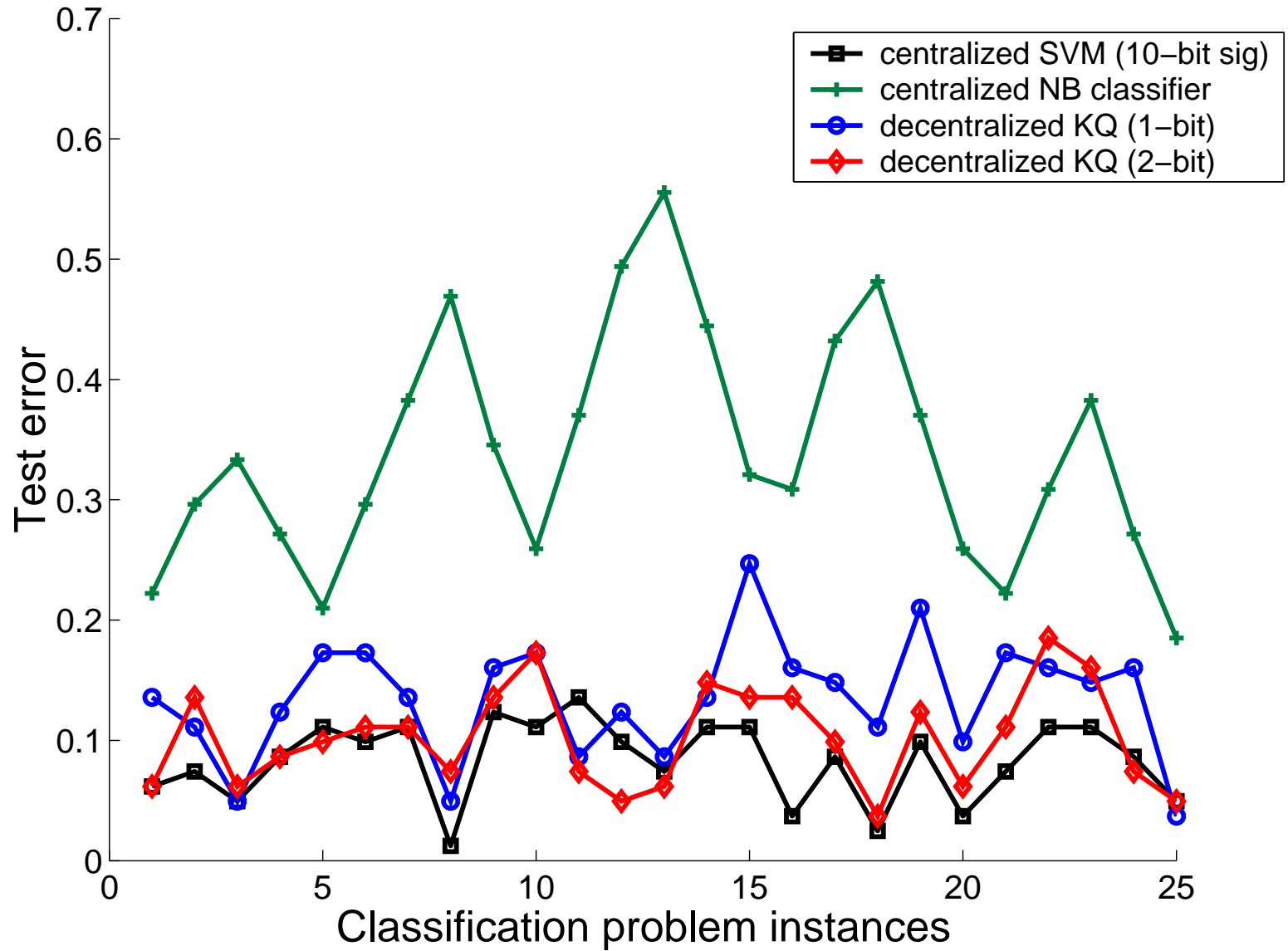


Wireless network with tiny Berkeley motes



- $5 \times 5 = 25$ 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
- Each problem has 25 training positions, 81 test positions
(Data collection courtesy Bruno Sinopoli)

Classification with Mica sensor motes



Conclusions

- Decentralized decision-making under communication constraints can be posed as parameterized kernel learning problems
- Extensions include
 - More complex decision making scheme \equiv more structured $Q(Z|X)$
 - Continuous data \equiv parameterizing $Q(Z|X)$