Applications of Exponential Families

Thanks to Yasemin Altun, Thomas Hofmann, Vishy Vishwanathan

Alexander J. Smola

Statistical Machine Learning Program
Canberra, ACT 0200 Australia
Alex.Smola@nicta.com.au

ICONIP 2006, Hong Kong, October 3

Outline

1. **Conditional Models**
   - Log-partition Function, Densities, and Expectations
   - Inner Products and Kernels
   - Examples of Kernels
2. Gaussian Process Classification
   - Feature map
   - Examples
3. Gaussian Process Regression
   - Homoscedastic Model
   - Heteroscedastic Model
4. Conditional Random Fields
   - Model Structure
   - Kernel Expansion
   - Connections to Hidden Markov Models

Conditional Models

Conditional Density

$$p(x|\theta) = \exp(\langle \phi(x), \theta \rangle - g(\theta))$$
$$p(y|x, \theta) = \exp(\langle \phi(x, y), \theta \rangle - g(\theta|x))$$

Log-partition function

$$g(\theta|x) = \log \int_y \exp(\langle \phi(x, y), \theta \rangle) dy$$

Sufficient Criterion

$p(x, y|\theta)$ is a member of the exponential family itself.

Key Idea

Avoid computing $\phi(x, y)$ directly, only evaluate inner products

$$k((x, y), (x', y')) := \langle \phi(x, y), \phi(x', y') \rangle$$

Conditional Models

Conditional Density

$$p(x|\theta) = \exp(\langle \phi(x), \theta \rangle - g(\theta))$$
$$p(y|x, \theta) = \exp(\langle \phi(x, y), \theta \rangle - g(\theta|x))$$

Log-partition function

$$g(\theta|x) = \log \int_y \exp(\langle \phi(x, y), \theta \rangle) dy$$

Sufficient Criterion

$p(x, y|\theta)$ is a member of the exponential family itself.

Key Idea

Avoid computing $\phi(x, y)$ directly, only evaluate inner products

$$k((x, y), (x', y')) := \langle \phi(x, y), \phi(x', y') \rangle$$

Conditional Models

Conditional Density

$$p(x|\theta) = \exp(\langle \phi(x), \theta \rangle - g(\theta))$$
$$p(y|x, \theta) = \exp(\langle \phi(x, y), \theta \rangle - g(\theta|x))$$

Log-partition function

$$g(\theta|x) = \log \int_y \exp(\langle \phi(x, y), \theta \rangle) dy$$

Sufficient Criterion

$p(x, y|\theta)$ is a member of the exponential family itself.

Key Idea

Avoid computing $\phi(x, y)$ directly, only evaluate inner products

$$k((x, y), (x', y')) := \langle \phi(x, y), \phi(x', y') \rangle$$

Conditional Models

Conditional Density

$$p(x|\theta) = \exp(\langle \phi(x), \theta \rangle - g(\theta))$$
$$p(y|x, \theta) = \exp(\langle \phi(x, y), \theta \rangle - g(\theta|x))$$

Log-partition function

$$g(\theta|x) = \log \int_y \exp(\langle \phi(x, y), \theta \rangle) dy$$

Sufficient Criterion

$p(x, y|\theta)$ is a member of the exponential family itself.

Key Idea

Avoid computing $\phi(x, y)$ directly, only evaluate inner products

$$k((x, y), (x', y')) := \langle \phi(x, y), \phi(x', y') \rangle$$

Conditional Distributions

Maximum a Posteriori Estimation

$$-\log p(\theta|X) = \sum_{i=1}^{m} -\langle\phi(x_i), \theta\rangle + mg(\theta) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle\phi(x_i, y_i), \theta\rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Solving the Problem

- The problem is strictly convex in θ.
- Direct solution impossible if we cannot compute $\phi(x, y)$.
- Solve convex problem in expansion coefficients.
- Expand θ in a linear combination of $\phi(x_i, y)$.

Conditional Distributions

Maximum a Posteriori Estimation

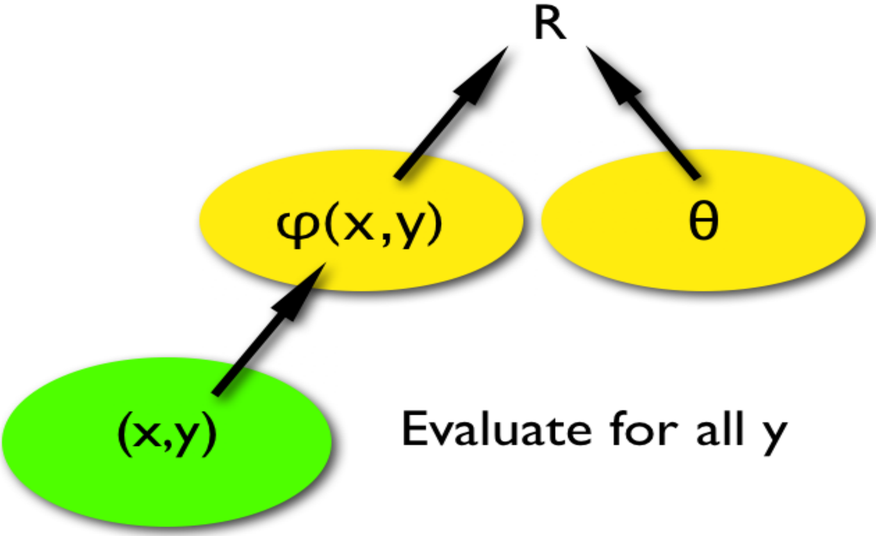
$$-\log p(\theta|X) = \sum_{i=1}^{m} -\langle \phi(x_i), \theta \rangle + mg(\theta) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Solving the Problem

- The problem is strictly convex in θ.
- Direct solution impossible if we cannot compute $\phi(x, y)$.
- Solve convex problem in expansion coefficients.
- Expand θ in a linear combination of $\phi(x_i, y)$.

Joint Feature Map



Representer Theorem

Objective Function

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Decomposition

- Decompose θ into $\theta = \theta_\| + \theta_\perp$ where

$$\theta_\| \in \mathrm{span}\{\phi(x_i, y) \text{ where } 1 \leq i \leq m \text{ and } y \in \mathcal{Y}\}$$

- Both $g(\theta|x_i)$ and $\langle \phi(x_i, y_i), \theta \rangle$ are independent of $\theta_\perp$.

Theorem

$-\log p(\theta|X, Y)$ is minimized for $\theta_\perp = 0$, hence $\theta = \theta_\|$.

Corollary

If $|\mathcal{Y}| < \infty$ we have a parametric optimization problem.

Representer Theorem

Objective Function

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Decomposition

- Decompose θ into $\theta = \theta_\| + \theta_\perp$ where

  $$\theta_\| \in \mathrm{span}\{\phi(x_i, y) \text{ where } 1 \leq i \leq m \text{ and } y \in \mathcal{Y}\}$$

- Both $g(\theta|x_i)$ and $\langle \phi(x_i, y_i), \theta \rangle$ are independent of $\theta_\perp$.

Theorem

$-\log p(\theta|X, Y)$ is minimized for $\theta_\perp = 0$, hence $\theta = \theta_\|$.

Corollary

If $|\mathcal{Y}| < \infty$ we have a parametric optimization problem.

Representer Theorem

Objective Function

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Decomposition

- Decompose θ into $\theta = \theta_\| + \theta_\perp$ where

  $$\theta_\| \in \operatorname{span}\{\phi(x_i, y) \text{ where } 1 \leq i \leq m \text{ and } y \in \mathcal{Y}\}$$

- Both $g(\theta|x_i)$ and $\langle \phi(x_i, y_i), \theta \rangle$ are independent of $\theta_\perp$.

Theorem

$-\log p(\theta|X, Y)$ *is minimized for* $\theta_\perp = 0$*, hence* $\theta = \theta_\|$.

Corollary

If $|\mathcal{Y}| < \infty$ *we have a parametric optimization problem.*

Representer Theorem

Objective Function

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Decomposition

- Decompose θ into $\theta = \theta_\parallel + \theta_\perp$ where

$$\theta_\parallel \in \operatorname{span}\{\phi(x_i, y) \text{ where } 1 \leq i \leq m \text{ and } y \in \mathcal{Y}\}$$

- Both $g(\theta|x_i)$ and $\langle \phi(x_i, y_i), \theta \rangle$ are independent of $\theta_\perp$.

Theorem

$-\log p(\theta|X, Y)$ *is minimized for* $\theta_\perp = 0$, *hence* $\theta = \theta_\parallel$.

Corollary

If $|\mathcal{Y}| < \infty$ *we have a parametric optimization problem.*

Using It

Expansion

$$\theta = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_{iy} \phi(x_i, y)$$

Inner Product

$$\langle \phi(x, y), \theta \rangle = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_{iy} k((x, y), (x_i, y))$$

Norm

$$\|\theta\|^2 = \sum_{i,j=1}^{m} \sum_{y, y' \in \mathcal{Y}} \alpha_{iy} \alpha_{jy'} k((x_i, y), (x_j, y'))$$

Log-partition function

$$g(\theta|x) = \log \sum_{y \in \mathcal{Y}} \exp\left(\langle \phi(x, y), \theta \rangle\right)$$

Using It

Expansion

$$\theta = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_{iy} \phi(x_i, y)$$

Inner Product

$$\langle \phi(x, y), \theta \rangle = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_{iy} k((x, y), (x_i, y))$$

Norm

$$\|\theta\|^2 = \sum_{i,j=1}^{m} \sum_{y,y' \in \mathcal{Y}} \alpha_{iy} \alpha_{jy'} k((x_i, y), (x_j, y'))$$

Log-partition function

$$g(\theta|x) = \log \sum_{y \in \mathcal{Y}} \exp\left(\langle \phi(x, y), \theta \rangle\right)$$

NATIONAL ICT AUSTRALIA LIMITED

Using It

Expansion

$$\theta = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_{iy} \phi(x_i, y)$$

Inner Product

$$\langle \phi(x, y), \theta \rangle = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_{iy} k((x, y), (x_i, y))$$

Norm

$$\|\theta\|^2 = \sum_{i,j=1}^{m} \sum_{y, y' \in \mathcal{Y}} \alpha_{iy} \alpha_{jy'} k((x_i, y), (x_j, y'))$$

Log-partition function

$$g(\theta|x) = \log \sum_{y \in \mathcal{Y}} \exp\left(\langle \phi(x, y), \theta \rangle\right)$$

Using It

Expansion

$$\theta = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_{iy} \phi(x_i, y)$$

Inner Product

$$\langle \phi(x, y), \theta \rangle = \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \alpha_{iy} k((x, y), (x_i, y))$$

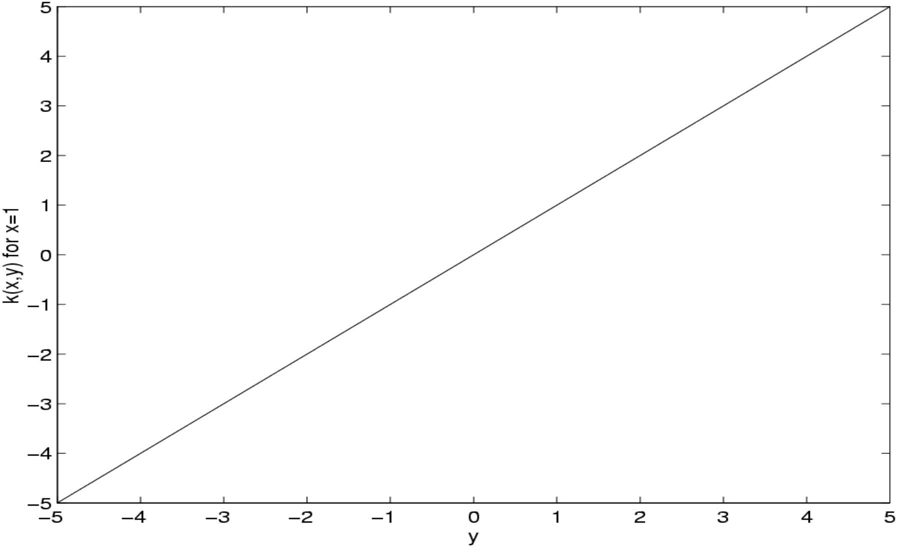
Norm

$$\|\theta\|^2 = \sum_{i,j=1}^{m} \sum_{y, y' \in \mathcal{Y}} \alpha_{iy} \alpha_{jy'} k((x_i, y), (x_j, y'))$$

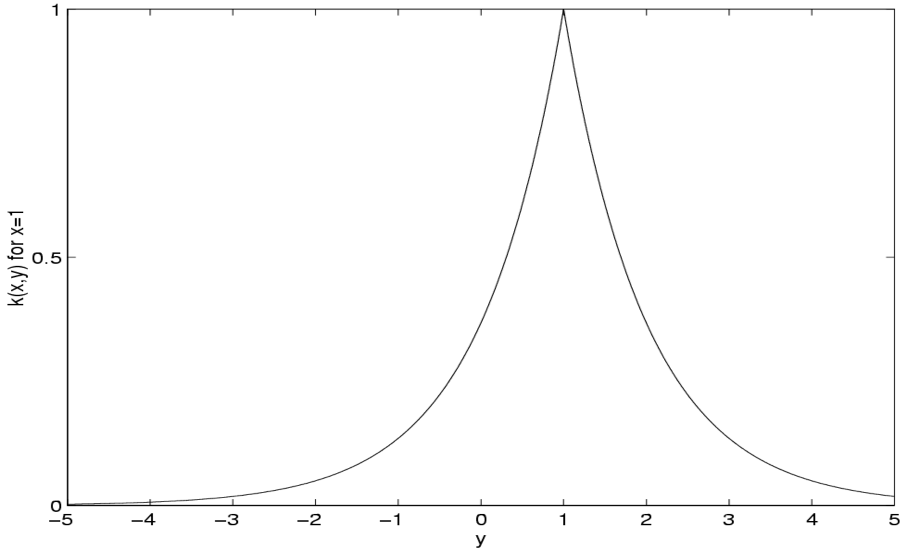
Log-partition function

$$g(\theta|x) = \log \sum_{y \in \mathcal{Y}} \exp\left(\langle \phi(x, y), \theta \rangle\right)$$

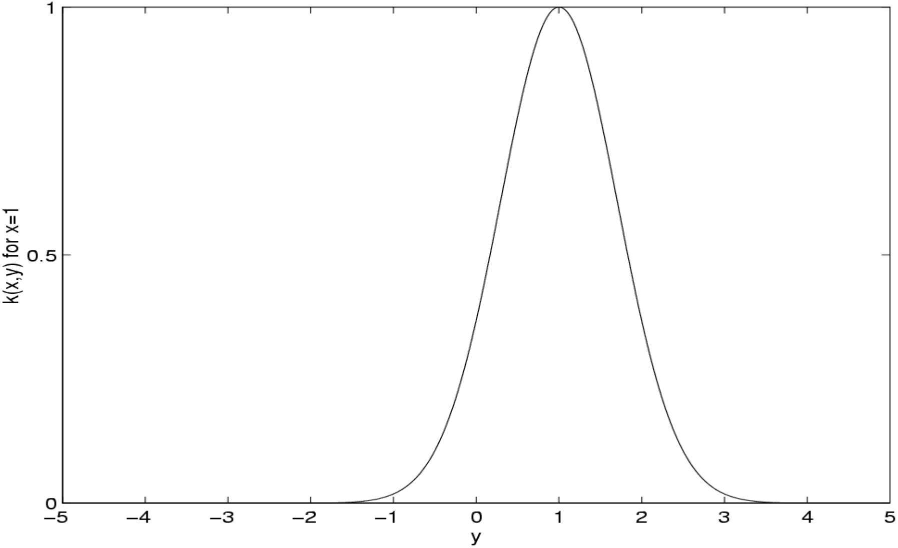
Linear Kernel



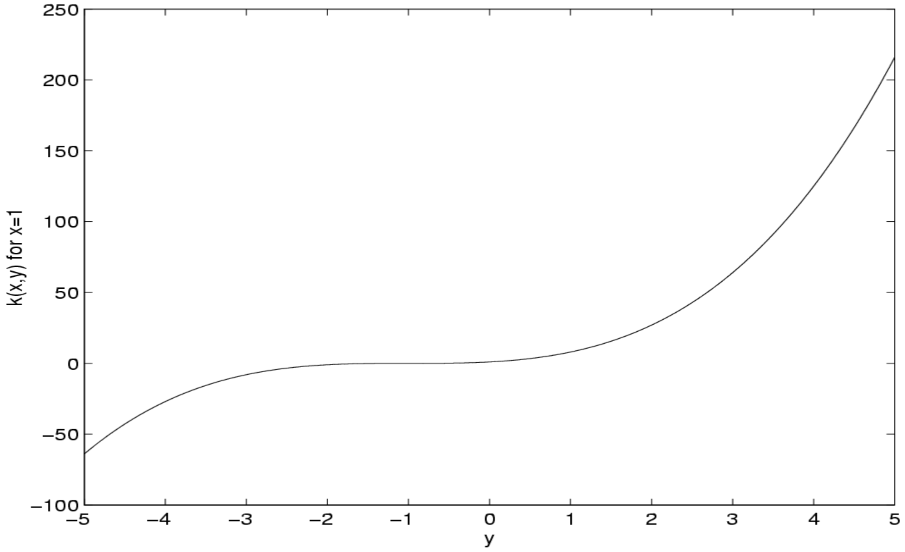
Laplace Kernel Covariance



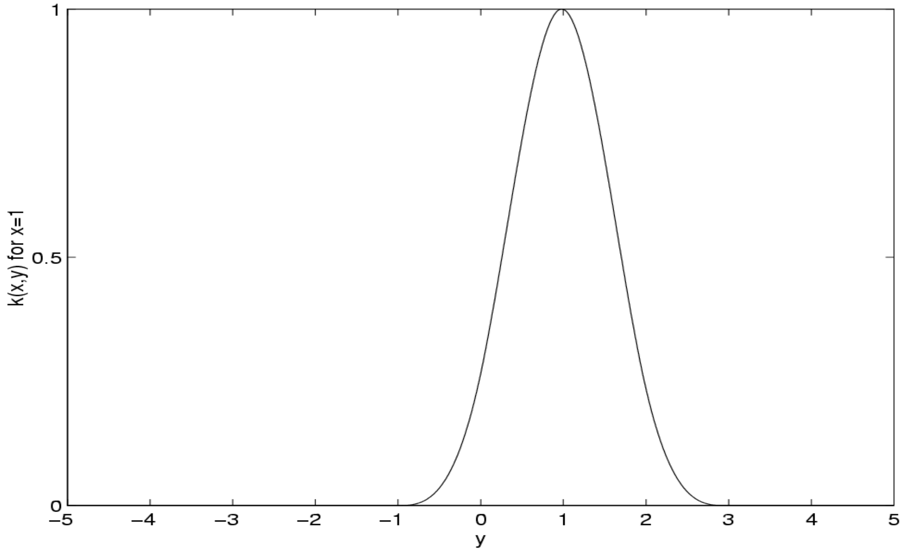
Gaussian Kernel



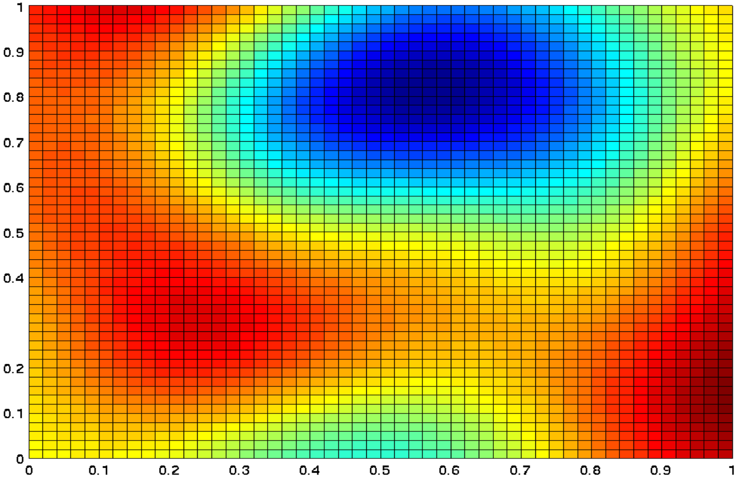
Polynomial (Order 3)



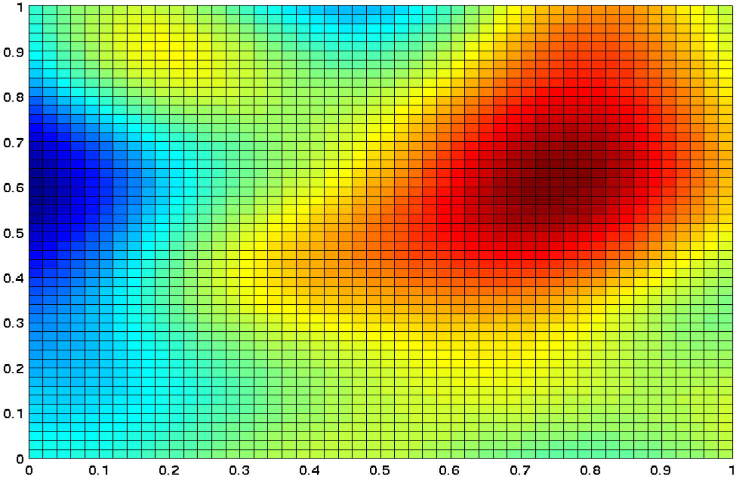
B_3-Spline Kernel



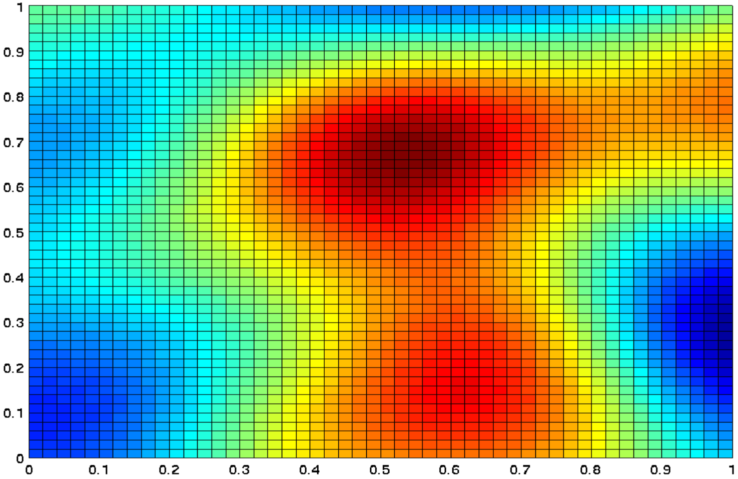
Sample from Gaussian RBF



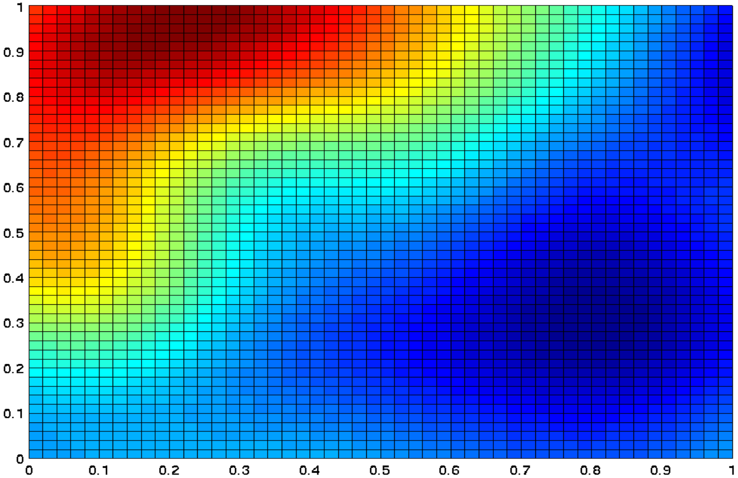
Sample from Gaussian RBF



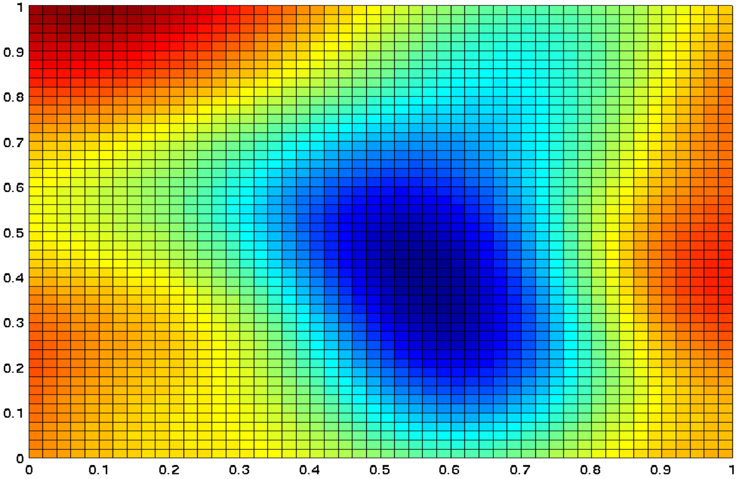
Sample from Gaussian RBF



Sample from Gaussian RBF

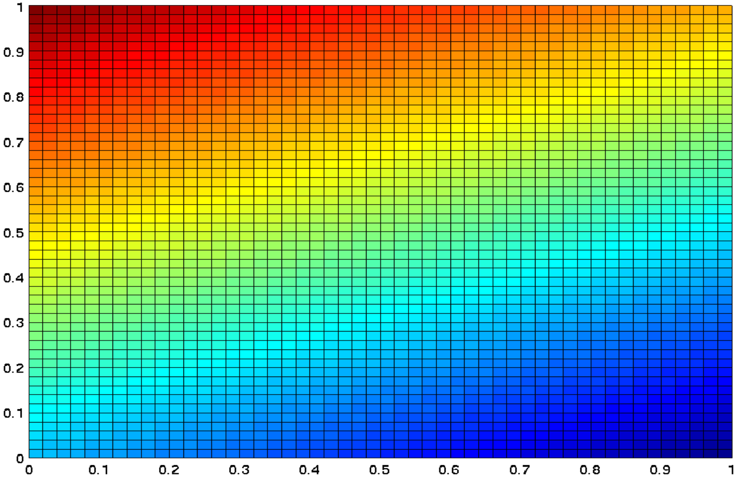


Sample from Gaussian RBF

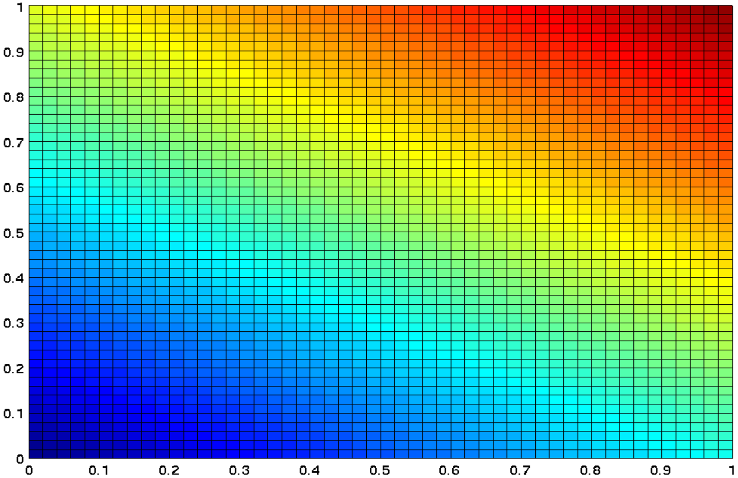


NATIONAL ICT AUSTRALIA LIMITED

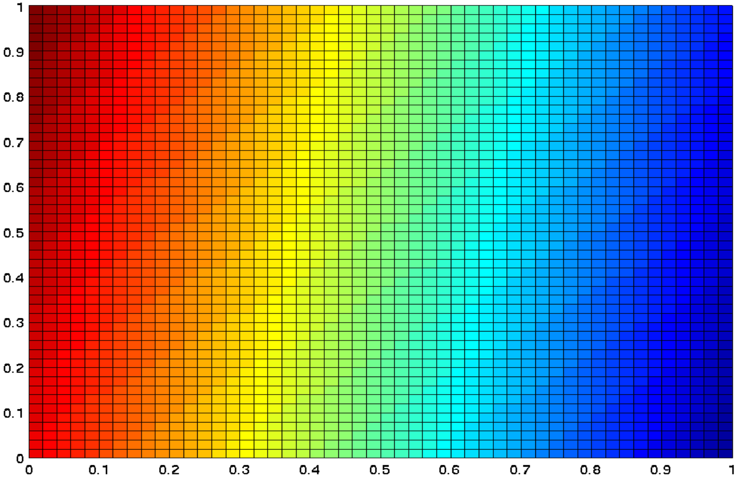
Sample from linear kernel



Sample from linear kernel

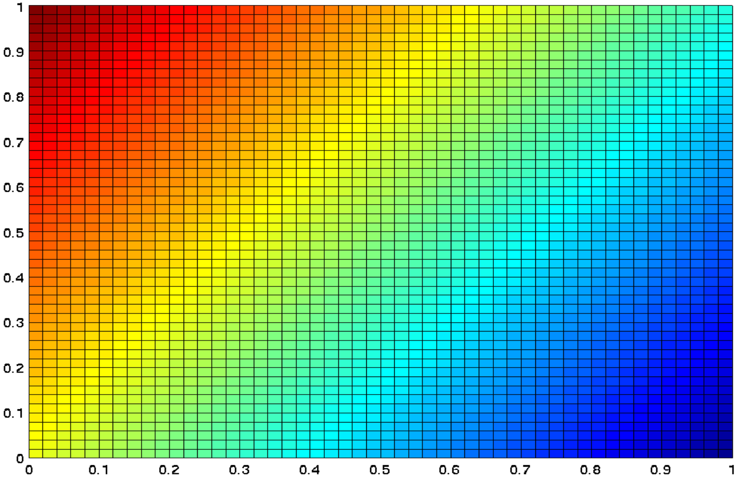


Sample from linear kernel

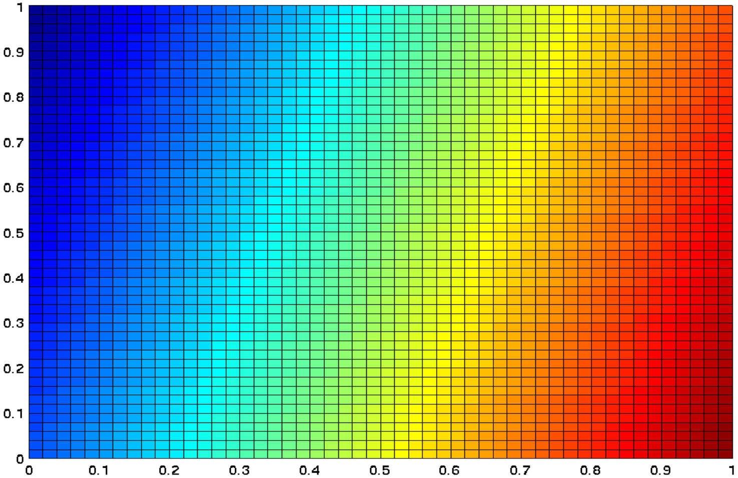


NATIONAL ICT AUSTRALIA LIMITED

Sample from linear kernel



Sample from linear kernel



Mini Summary

Choose a suitable sufficient statistic $\phi(x, y)$

- Conditionally multinomial distribution leads to Gaussian Process multiclass classifier.
- Conditionally Gaussian leads to Gaussian Process regression. **Note:** we estimate mean and variance.
- Conditionally Poisson distributions yield spatial Poisson regression.

Solve the optimization problem

This is typically convex.

The bottom line

Instead of choosing $k(x, x')$ choose $k((x, y), (x', y'))$.

Mini Summary

Choose a suitable sufficient statistic $\phi(x, y)$

- Conditionally multinomial distribution leads to Gaussian Process multiclass classifier.
- Conditionally Gaussian leads to Gaussian Process regression. **Note:** we estimate mean and variance.
- Conditionally Poisson distributions yield spatial Poisson regression.

Solve the optimization problem

This is typically convex.

The bottom line

Instead of choosing $k(x, x')$ choose $k((x, y), (x', y'))$.

Mini Summary

Choose a suitable sufficient statistic $\phi(x, y)$

- Conditionally multinomial distribution leads to Gaussian Process multiclass classifier.
- Conditionally Gaussian leads to Gaussian Process regression. **Note:** we estimate mean and variance.
- Conditionally Poisson distributions yield spatial Poisson regression.

Solve the optimization problem

This is typically convex.

The bottom line

Instead of choosing $k(x, x')$ choose $k((x, y), (x', y'))$.

Mini Summary

Choose a suitable sufficient statistic $\phi(x, y)$

- Conditionally multinomial distribution leads to Gaussian Process multiclass classifier.
- Conditionally Gaussian leads to Gaussian Process regression. **Note:** we estimate mean and variance.
- Conditionally Poisson distributions yield spatial Poisson regression.

Solve the optimization problem

This is typically convex.

The bottom line

Instead of choosing $k(x, x')$ choose $k((x, y), (x', y'))$.

Mini Summary

Choose a suitable sufficient statistic $\phi(x, y)$

- Conditionally multinomial distribution leads to Gaussian Process multiclass classifier.
- Conditionally Gaussian leads to Gaussian Process regression. **Note:** we estimate mean and variance.
- Conditionally Poisson distributions yield spatial Poisson regression.

Solve the optimization problem

This is typically convex.

The bottom line

Instead of choosing $k(x, x')$ choose $k((x, y), (x', y'))$.

Mini Summary

Choose a suitable sufficient statistic $\phi(x, y)$

- Conditionally multinomial distribution leads to Gaussian Process multiclass classifier.
- Conditionally Gaussian leads to Gaussian Process regression. **Note:** we estimate mean and variance.
- Conditionally Poisson distributions yield spatial Poisson regression.

Solve the optimization problem

This is typically convex.

The bottom line

Instead of choosing $k(x, x')$ choose $k((x, y), (x', y'))$.

Outline

Gaussian Process Classification

Sufficient Statistic

We pick $\phi(x, y) = \phi(x) \otimes e_y$, that is

$$k((x, y), (x', y')) = k(x, x')\delta_{yy'} \text{ where } y, y' \in \{1, \ldots, n\}$$

Kernel Expansion

By the representer theorem we get that

$$\theta = \sum_{i=1}^{m} \sum_{y} \alpha_{iy} \phi(x_i, y)$$

Optimization Problem

Not too messy and convex.

Gaussian Process Classification

Sufficient Statistic

We pick $\phi(x, y) = \phi(x) \otimes e_y$, that is

$$k((x, y), (x', y')) = k(x, x')\delta_{yy'} \text{ where } y, y' \in \{1, \ldots, n\}$$

Kernel Expansion

By the representer theorem we get that

$$\theta = \sum_{i=1}^{m} \sum_{y} \alpha_{iy} \phi(x_i, y)$$

Optimization Problem

Not too messy and convex.

Gaussian Process Classification

Sufficient Statistic

We pick $\phi(x, y) = \phi(x) \otimes e_y$, that is

$$k((x, y), (x', y')) = k(x, x')\delta_{yy'} \text{ where } y, y' \in \{1, \ldots, n\}$$

Kernel Expansion

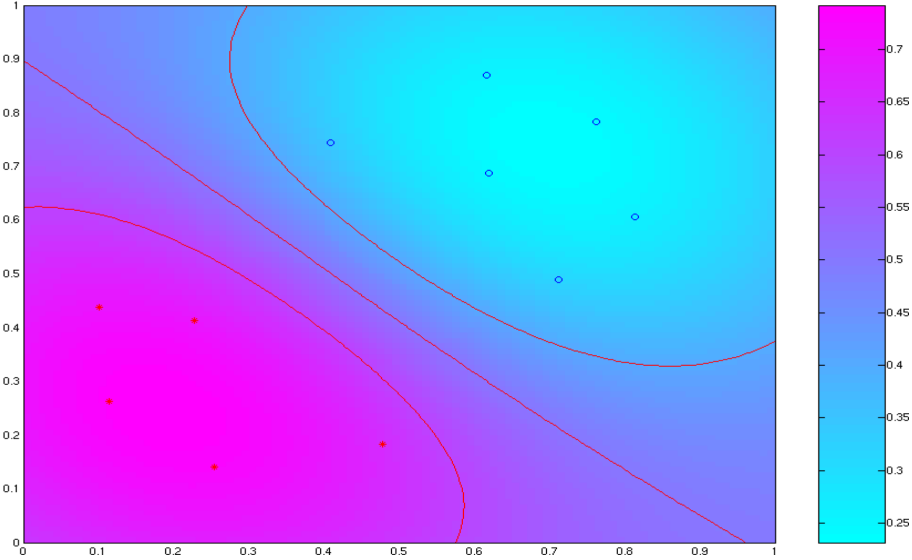
By the representer theorem we get that

$$\theta = \sum_{i=1}^{m} \sum_{y} \alpha_{iy} \phi(x_i, y)$$

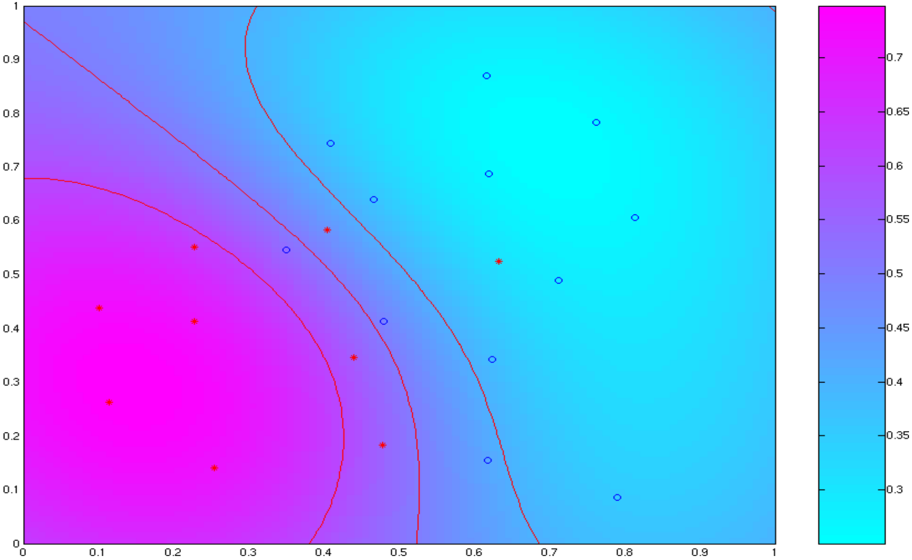
Optimization Problem

Not too messy and convex.

A Toy Example



Noisy Data



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3

Mini Summary

Feature Map

- Conditionally multinomial models $y|x$
- Feature map is $e_y \otimes \phi(x)$
- Kernel $k((x, y), (x', y')) = \delta_{y,y'} k(x, x')$
- Could use different interaction between labels.

Optimization Problem

- Convex problem
- Solve in dual space by Newton's method
- Could solve in primal space if $\phi(x, y)$ can be computed efficiently.

Caveat

- True posterior is only approximated by mode of posterior.
- Would need sampling methods for exact inference.

Outline

1. Conditional Models
   - Log-partition Function, Densities, and Expectations
   - Inner Products and Kernels
   - Examples of Kernels
2. Gaussian Process Classification
   - Feature map
   - Examples
3. **Gaussian Process Regression**
   - Homoscedastic Model
   - Heteroscedastic Model
4. Conditional Random Fields
   - Model Structure
   - Kernel Expansion
   - Connections to Hidden Markov Models

Regression

Recall: Maximum a Posteriori Estimation

$$-\log p(\theta|X,Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Domain

- Continuous domain of observations $\mathcal{Y} = \mathbb{R}$
- We want to have a conditionally normal distribution $y|x$.
- Log-partition function $g(\theta|x)$ easy to compute **in closed form** as normal distribution.

Regression

Recall: Maximum a Posteriori Estimation

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Domain

- Continuous domain of observations $\mathcal{Y} = \mathbb{R}$
- We want to have a conditionally normal distribution $y|x$.
- Log-partition function $g(\theta|x)$ easy to compute **in closed form** as normal distribution.

Regression

Recall: Maximum a Posteriori Estimation

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2} \|\theta\|^2 + c$$

Domain

- Continuous domain of observations $\mathcal{Y} = \mathbb{R}$
- We want to have a conditionally normal distribution $y|x$.
- Log-partition function $g(\theta|x)$ easy to compute **in closed form** as normal distribution.

Regression

Recall: Maximum a Posteriori Estimation

$$-\log p(\theta|X, Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Domain

- Continuous domain of observations $\mathcal{Y} = \mathbb{R}$
- We want to have a conditionally normal distribution $y|x$.
- Log-partition function $g(\theta|x)$ easy to compute **in closed form** as normal distribution.

Regression

Recall: Maximum a Posteriori Estimation

$$-\log p(\theta|X,Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

Domain

- Continuous domain of observations $\mathcal{Y} = \mathbb{R}$
- We want to have a conditionally normal distribution $y|x$.
- Log-partition function $g(\theta|x)$ easy to compute **in closed form** as normal distribution.

Standard Model

Key Idea

- For fixed parameter θ we want to have a normal distribution with fixed variance.
- Exponential family model of $y|x$ with $c(x)y - \frac{1}{2\sigma^2}y^2$ in exponent.

Sufficient Statistic

Pick $\phi(x, y) = (y\phi(x), y^2)$, that is

$$k((x, y), (x', y')) = k(x, x')yy' + y^2 y'^2 \text{ where } y, y' \in \mathbb{R}$$

Traditionally the variance is fixed.

Inference Problem

After straightforward algebra we get standard GP regression model.

Standard Model

Key Idea

- For fixed parameter θ we want to have a normal distribution with fixed variance.
- Exponential family model of $y|x$ with $c(x)y - \frac{1}{2\sigma^2}y^2$ in exponent.

Sufficient Statistic

Pick $\phi(x, y) = (y\phi(x), y^2)$, that is

$$k((x, y), (x', y')) = k(x, x')yy' + y^2y'^2 \text{ where } y, y' \in \mathbb{R}$$

Traditionally the variance is fixed.

Inference Problem

After straightforward algebra we get standard GP regression model.

NATIONAL ICT AUSTRALIA LIMITED

Standard Model

Key Idea

- For fixed parameter θ we want to have a normal distribution with fixed variance.
- Exponential family model of $y|x$ with $c(x)y - \frac{1}{2\sigma^2}y^2$ in exponent.

Sufficient Statistic

Pick $\phi(x, y) = (y\phi(x), y^2)$, that is

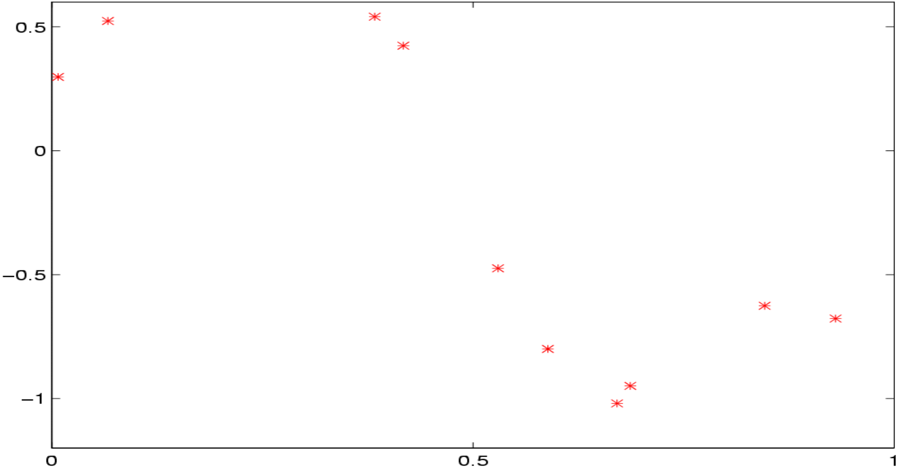
$$k((x, y), (x', y')) = k(x, x')yy' + y^2y'^2 \text{ where } y, y' \in \mathbb{R}$$

Traditionally the variance is fixed.

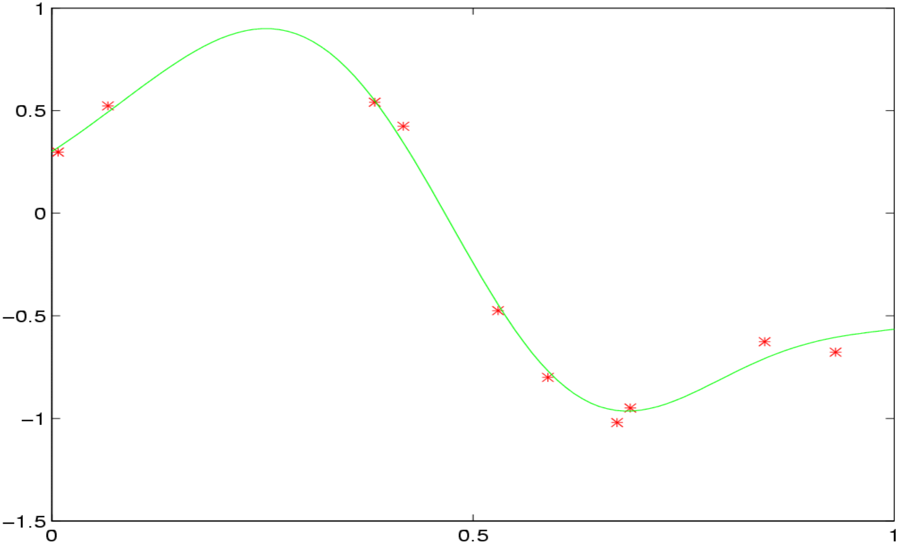
Inference Problem

After straightforward algebra we get standard GP regression model.

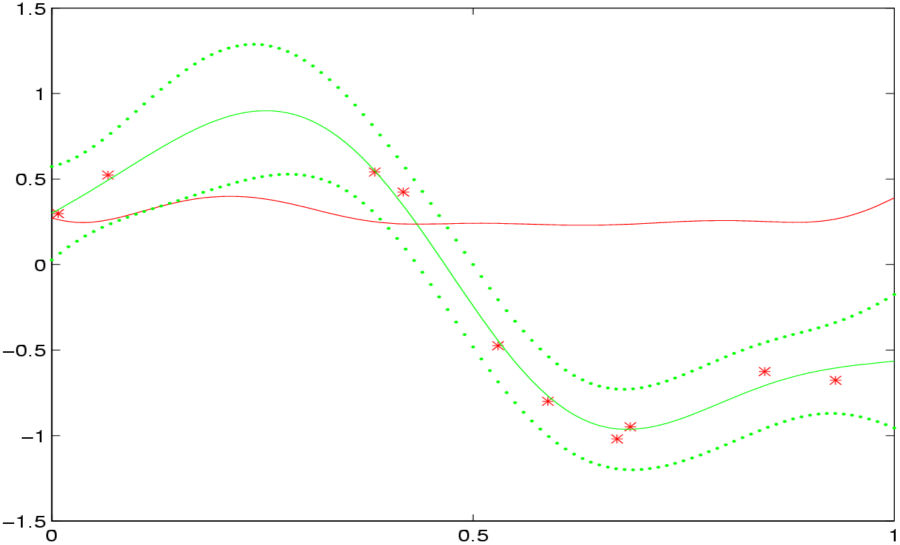
Training Data



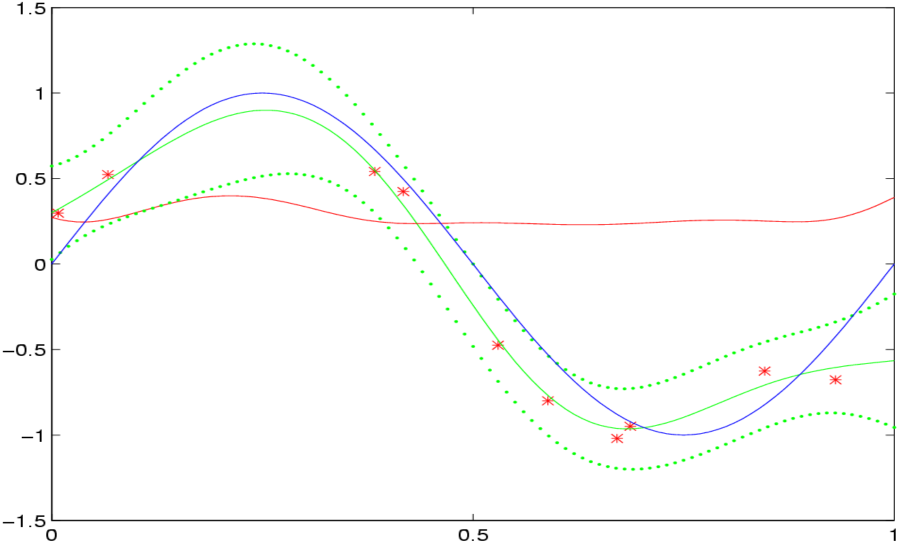
Mean $\vec{k}^\top(x)(K + \sigma^2\mathbf{1})^{-1}y$



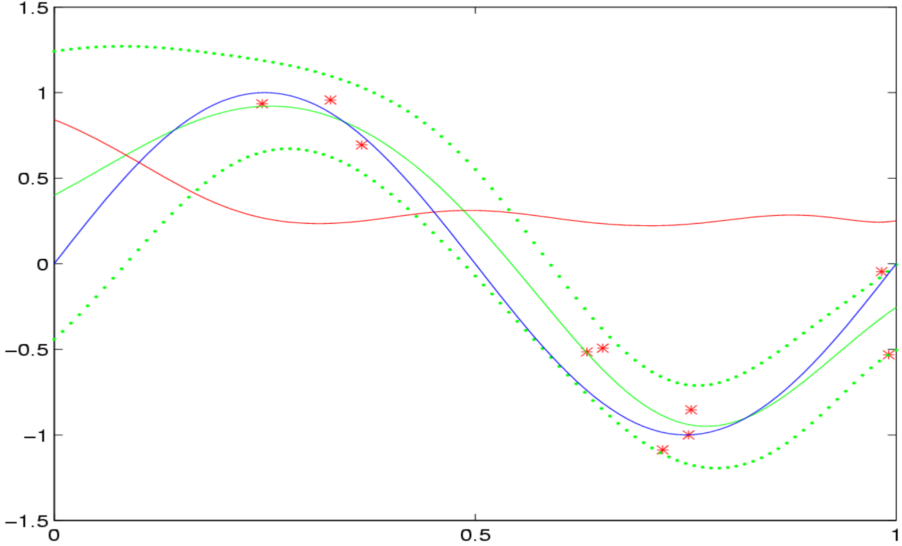
Variance $k(x,x) + \sigma^2 - \vec{k}^\top(x)(K + \sigma^2 \mathbf{1})^{-1}\vec{k}(x)$



Putting everything together . . .



Another Example



Heteroscedastic Regression

Key Idea

Make both linear and quadratic term in $y|x$ dependent on x.

Sufficient Statistic

Pick $\phi(x, y) = (y\phi_1(x), y^2\phi_2(x))$, that is

$$k((x, y), (x', y')) = k_1(x, x')yy' + k_2(x, x')y^2y'^2 \text{ where } y, y' \in \mathbb{R}$$

We estimate mean and variance **simultaneously**.

Kernel Expansion

By the representer theorem (and more algebra) we get

$$\theta = \left(\sum_{i=1}^{m} \alpha_{i1}\phi_1(x_i), \sum_{i=1}^{m} \alpha_{i2}\phi_2(x_i)\right)$$

Heteroscedastic Regression

Key Idea

Make both linear and quadratic term in $y|x$ dependent on x.

Sufficient Statistic

Pick $\phi(x, y) = (y\phi_1(x), y^2\phi_2(x))$, that is

$$k((x, y), (x', y')) = k_1(x, x')yy' + k_2(x, x')y^2 y'^2 \text{ where } y, y' \in \mathbb{R}$$

We estimate mean and variance **simultaneously**.

Kernel Expansion

By the representer theorem (and more algebra) we get

$$\theta = \left(\sum_{i=1}^{m} \alpha_{i1}\phi_1(x_i), \sum_{i=1}^{m} \alpha_{i2}\phi_2(x_i) \right)$$

Heteroscedastic Regression

Key Idea

Make both linear and quadratic term in $y|x$ dependent on x.

Sufficient Statistic

Pick $\phi(x, y) = (y\phi_1(x), y^2\phi_2(x))$, that is

$$k((x, y), (x', y')) = k_1(x, x')yy' + k_2(x, x')y^2y'^2 \text{ where } y, y' \in \mathbb{R}$$

We estimate mean and variance **simultaneously**.

Kernel Expansion

By the representer theorem (and more algebra) we get

$$\theta = \left(\sum_{i=1}^{m} \alpha_{i1}\phi_1(x_i), \sum_{i=1}^{m} \alpha_{i2}\phi_2(x_i)\right)$$

Heteroscedastic Regression

Optimization Problem

$$\sum_{i=1}^{m}\Big[-\frac{1}{4}\big[\sum_{j=1}^{m}\alpha_{1j}k_1(x_i,x_j)\big]^{\top}\big[\sum_{j=1}^{m}\alpha_{2j}k_2(x_i,x_j)\big]^{-1}\big[\sum_{j=1}^{m}\alpha_{1j}k_1(x_i,x_j)\big]$$

$$-\frac{1}{2}\log\det -2\big[\sum_{j=1}^{m}\alpha_{2j}k_2(x_i,x_j)\big] - \sum_{j=1}^{m}\big[y_i^{\top}\alpha_{1j}k_1(x_i,x_j) + (y_j^{\top}\alpha_{2j}y_j)k$$

$$+\frac{1}{2\sigma^2}\sum_{i,j}\alpha_{1i}^{\top}\alpha_{1j}k_1(x_i,x_j) + \mathrm{tr}\,\left[\alpha_{2i}\alpha_{2j}^{\top}\right]k_2(x_i,x_j).$$

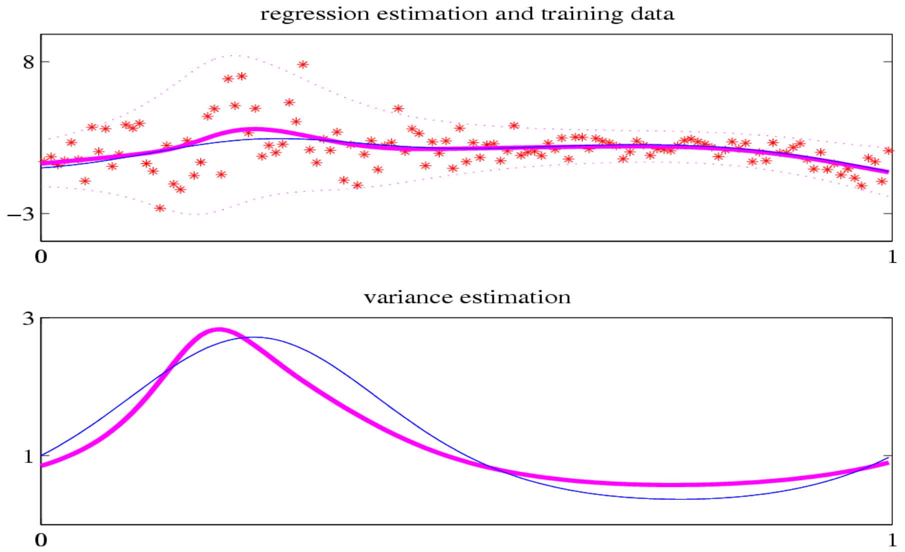
subject to $0 \succ \sum_{i=1}^{m}\alpha_{2i}k(x_i,x_j)$

Heteroscedastic Regression

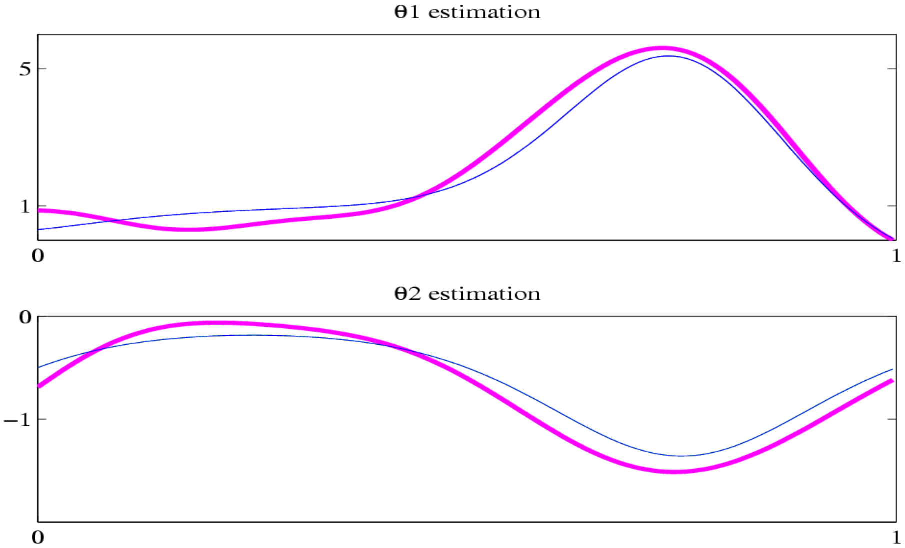
Optimization Problem

- The problem is convex
- The log-determinant from the normalization of the Gaussian distribution acts as a **barrrier function**.
- We get a semidefinite program.
- Because of the barrier function we can solve it by Newton's method.

Heteroscedastic Regression



Natural Parameters



Mini Summary

Sufficient Statistics

- Conditionally normal model explained if $\phi(y, x)$ has linear and quadratic terms.
- For homoscedastic model we only need to estimate the linear term. Quadratic term is fixed.
- Second order kernel in y, arbitrary kernel in x.

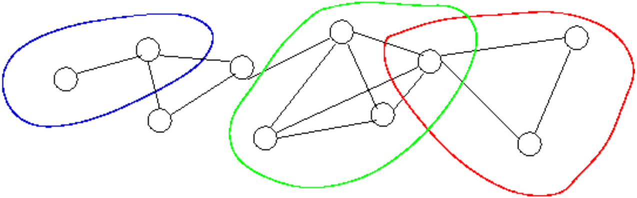
Optimization

- Linear system is all we need for fixed variance
- Semidefinite program for heteroscedastic estimation
- Can be solved by Newton's method, as the log-determinant acts as barrier function.

Outline

1. Conditional Models
   - Log-partition Function, Densities, and Expectations
   - Inner Products and Kernels
   - Examples of Kernels
2. Gaussian Process Classification
   - Feature map
   - Examples
3. Gaussian Process Regression
   - Homoscedastic Model
   - Heteroscedastic Model
4. **Conditional Random Fields**
   - Model Structure
   - Kernel Expansion
   - Connections to Hidden Markov Models

Graphical Models



Corollary

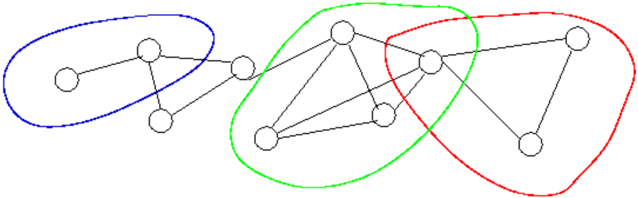
The conditional density $p(y|x)$ can be written in terms of potential functions defined on the maximal cliques in y.

Corollary

Featuremap $\phi(x)$ decomposes via $\phi(x) = (\ldots, \phi_c(x_c), \ldots)$. Consequently we can write the kernel via

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = \sum_c \langle \phi_c(x_c), \phi_c(x'_c) \rangle = \sum_c k_c(x_c, x'_c)$$

Graphical Models



Corollary

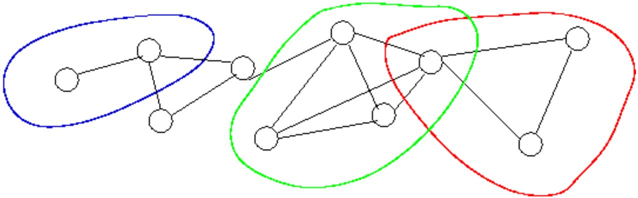
The conditional density $p(y|x)$ can be written in terms of potential functions defined on the maximal cliques in y.

Corollary

Featuremap $\phi(x)$ decomposes via $\phi(x) = (\ldots, \phi_c(x_c), \ldots)$. Consequently we can write the kernel via

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = \sum_c \langle \phi_c(x_c), \phi_c(x'_c) \rangle = \sum_c k_c(x_c, x'_c)$$

Graphical Models



Corollary

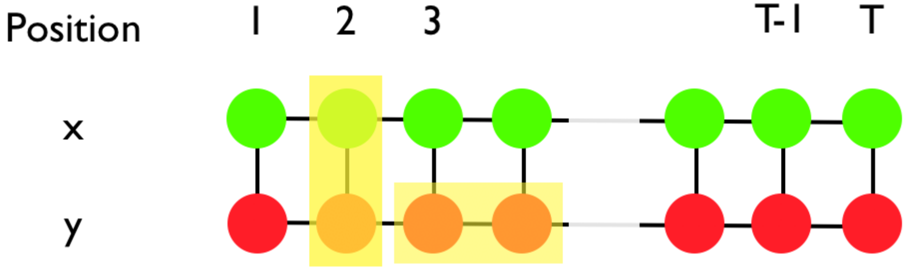
The conditional density $p(y|x)$ can be written in terms of potential functions defined on the maximal cliques in y.

Corollary

Featuremap $\phi(x)$ decomposes via $\phi(x) = (\ldots, \phi_c(x_c), \ldots)$. Consequently we can write the kernel via

$$k(x, x') = \langle \phi(x), \phi(x') \rangle = \sum_c \langle \phi_c(x_c), \phi_c(x'_c) \rangle = \sum_c k_c(x_c, x'_c)$$

Conditional Random Fields



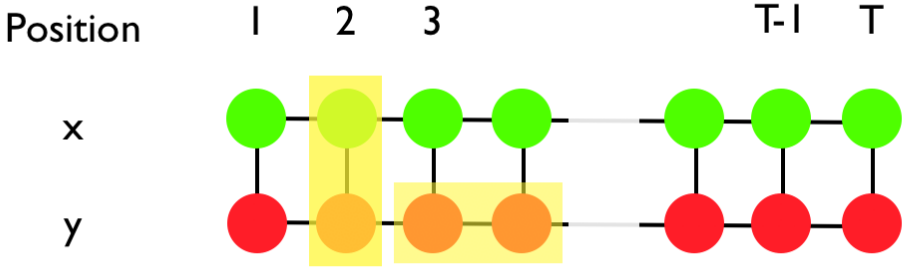
Key Points

- Cliques are (x_t, y_t), (x_t, x_{t+1}), and (y_t, y_{t+1})
- We can drop cliques in (x_t, x_{t+1})

$$p(y|x,\theta) = \exp\Big(\sum_t \langle \phi_{xy}(x_t, y_t), \theta_{xy,t}\rangle + \langle \phi_{yy}(y_t, y_{t+1}), \theta_{yy,t}\rangle$$
$$+ \langle \phi_{xx}(x_t, x_{t+1}), \theta_{xx,t}\rangle - g(\theta|x)\Big)$$

NATIONAL ICT AUSTRALIA LIMITED

Conditional Random Fields



Key Points

- Cliques are $(x_t, y_t), (x_t, x_{t+1})$, and (y_t, y_{t+1})
- We can drop cliques in (x_t, x_{t+1})

$$p(y|x,\theta) = \exp\Big(\sum_t \langle \phi_{xy}(x_t, y_t), \theta_{xy,t}\rangle + \langle \phi_{yy}(y_t, y_{t+1}), \theta_{yy,t}\rangle + \langle \phi_{xx}(x_t, x_{t+1}), \theta_{xx,t}\rangle - g(\theta|x)\Big)$$

Computational Issues

Key Points

- Compute $g(\theta|x)$ via dynamic programming
- Assume stationarity of the model, that is θ_c does not depend on the position of the

Dynamic Programming

$$
\begin{aligned}
& g(\theta|x) \\
&= \log \sum_{y_1,\ldots,y_T} \prod_{t=1}^{T} \underbrace{\exp\left(\langle \phi_{xy}(x_t, y_t), \theta_{xy}\rangle + \langle \phi_{yy}(y_t, y_{t+1}), \theta_{yy}\rangle\right)}_{M_t(y_t, y_{t+1})} \\
&= \log \sum_{y_1} \sum_{y_2} M_1(y_1, y_2) \sum_{y_3} M_2(y_2, y_3) \ldots \sum_{y_T} M_T(y_{T-1}, y_T)
\end{aligned}
$$

Efficient computation of $g(\theta|x), p(y_t|x, \theta)$ and $p(y_t, y_{t+1}|x, \theta)$.

Computational Issues

Key Points

- Compute $g(\theta|x)$ via dynamic programming
- Assume stationarity of the model, that is θ_c does not depend on the position of the

Dynamic Programming

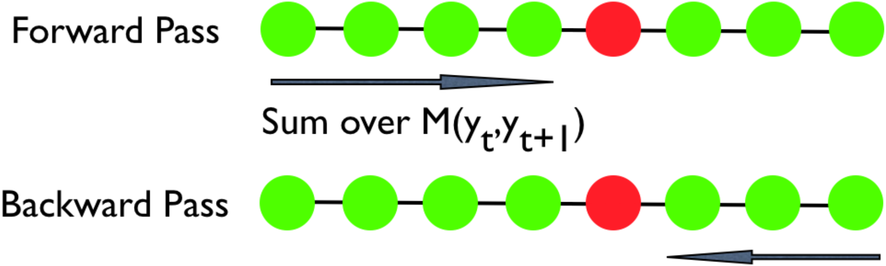
$$g(\theta|x)$$

$$= \log \sum_{y_1,\ldots,y_T} \prod_{t=1}^{T} \underbrace{\exp\left(\langle \phi_{xy}(x_t, y_t), \theta_{xy}\rangle + \langle \phi_{yy}(y_t, y_{t+1}), \theta_{yy}\rangle\right)}_{M_t(y_t, y_{t+1})}$$

$$= \log \sum_{y_1} \sum_{y_2} M_1(y_1, y_2) \sum_{y_3} M_2(y_2, y_3) \ldots \sum_{y_T} M_T(y_{T-1}, y_T)$$

Efficient computation of $g(\theta|x), p(y_t|x,\theta)$ and $p(y_t, y_{t+1}|x,\theta)$.

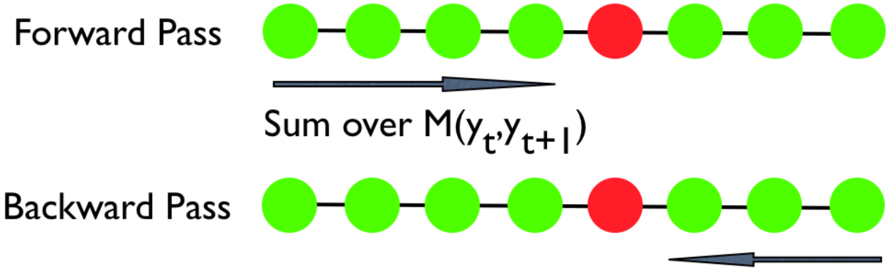
Forward Backward Algorithm



Key Idea

- Store sum over all $y_1, \ldots, y_{t-1}$ (forward pass) and over all $y_{t+1}, \ldots, y_T$ as intermediate values
- We get those values for all positions t in one sweep.
- Extend this to message passing (when we have trees).

Forward Backward Algorithm



Key Idea

- Store sum over all $y_1, \ldots, y_{t-1}$ (forward pass) and over all $y_{t+1}, \ldots, y_T$ as intermediate values
- We get those values for all positions t in one sweep.
- Extend this to message passing (when we have trees).

Minimization

Objective Function

$$-\log p(\theta|X,Y) = \sum_{i=1}^{m} -\langle \phi(x_i,y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

$$\partial_\theta - \log p(\theta|X,Y) = \sum_{i=1}^{m} -\phi(x_i,y_i) + \mathbf{E}\left[\phi(x_i,y_i)|x_i\right] + \frac{1}{\sigma^2}\theta$$

We only need $\mathbf{E}\left[\phi_{xy}(x_{it},y_{it})|x_i\right]$ and $\mathbf{E}\left[\phi_{yy}(y_{it},y_{i(t+1)})|x_i\right]$.

Kernel Trick

- Conditional expectations of $\Phi(x_{it},y_{it})$ cannot be computed explicitly **but** inner products can.

$$\langle \phi_{xy}(x_t',y_t'), \mathbf{E}\left[\phi_{xy}(x_t,y_t)|x\right] = \mathbf{E}\left[k((x_t',y_t'),(x_t,y_t)|x\right]$$

- Only need marginals $p(y_t|x,\theta)$ and $p(y_t,y_{t+1}|x,\theta)$, which we get via dynamic programming.

Minimization

Objective Function

$$-\log p(\theta|X,Y) = \sum_{i=1}^{m} -\langle \phi(x_i, y_i), \theta \rangle + g(\theta|x_i) + \frac{1}{2\sigma^2}\|\theta\|^2 + c$$

$$\partial_\theta - \log p(\theta|X,Y) = \sum_{i=1}^{m} -\phi(x_i, y_i) + \mathbf{E}\left[\phi(x_i, y_i)|x_i\right] + \frac{1}{\sigma^2}\theta$$

We only need $\mathbf{E}\left[\phi_{xy}(x_{it}, y_{it})|x_i\right]$ and $\mathbf{E}\left[\phi_{yy}(y_{it}, y_{i(t+1)})|x_i\right]$.

Kernel Trick

- Conditional expectations of $\Phi(x_{it}, y_{it})$ cannot be computed explicitly **but** inner products can.

$$\langle \phi_{xy}(x'_t, y'_t), \mathbf{E}\left[\phi_{xy}(x_t, y_t)|x\right] = \mathbf{E}\left[k((x'_t, y'_t), (x_t, y_t)|x\right]$$

- Only need marginals $p(y_t|x,\theta)$ and $p(y_t, y_{t+1}|x,\theta)$, which we get via dynamic programming.

NATIONAL ICT AUSTRALIA LIMITED

Subspace Representer Theorem

Representer Theorem

Solutions of the MAP problem are given by

$$\theta \in \mathrm{span}\{\phi(x_i, y) \text{ for all } y \in \mathcal{Y} \text{ and } 1 \leq i \leq n\}$$

Big Problem

$|\mathcal{Y}|$ could be huge, e.g. for sequence annotation 2^n.

Solution

- Exploit decomposition of $\phi(x, y)$ into sufficient statistics on cliques.
- Restriction of $\mathcal{Y}$ to cliques is much smaller.

$$\theta_c \in \mathrm{span}\{\phi_c(x_{ci}, y_c) \text{ for all } y_c \in \mathcal{Y}_c \text{ and } 1 \leq i \leq n\}$$

Rather than 2^n we now get $2^{|c|}$.

Subspace Representer Theorem

Representer Theorem

Solutions of the MAP problem are given by

$$\theta \in \operatorname{span}\{\phi(x_i, y) \text{ for all } y \in \mathcal{Y} \text{ and } 1 \leq i \leq n\}$$

Big Problem

$|\mathcal{Y}|$ could be huge, e.g. for sequence annotation 2^n.

Solution

- Exploit decomposition of $\phi(x, y)$ into sufficient statistics on cliques.
- Restriction of $\mathcal{Y}$ to cliques is much smaller.

$$\theta_c \in \operatorname{span}\{\phi_c(x_{ci}, y_c) \text{ for all } y_c \in \mathcal{Y}_c \text{ and } 1 \leq i \leq n\}$$

Rather than 2^n we now get $2^{|c|}$.

Subspace Representer Theorem

Representer Theorem

Solutions of the MAP problem are given by

$$\theta \in \mathrm{span}\{\phi(x_i, y) \text{ for all } y \in \mathcal{Y} \text{ and } 1 \leq i \leq n\}$$

Big Problem

$|\mathcal{Y}|$ could be huge, e.g. for sequence annotation 2^n.

Solution

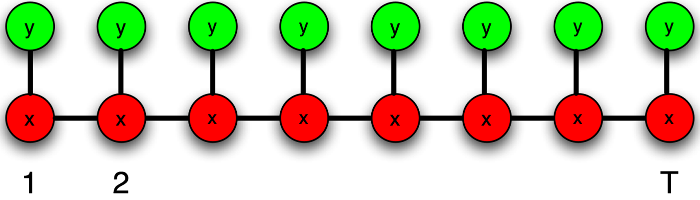
- Exploit decomposition of $\phi(x, y)$ into sufficient statistics on cliques.
- Restriction of $\mathcal{Y}$ to cliques is much smaller.

$$\theta_c \in \mathrm{span}\{\phi_c(x_{ci}, y_c) \text{ for all } y_c \in \mathcal{Y}_c \text{ and } 1 \leq i \leq n\}$$

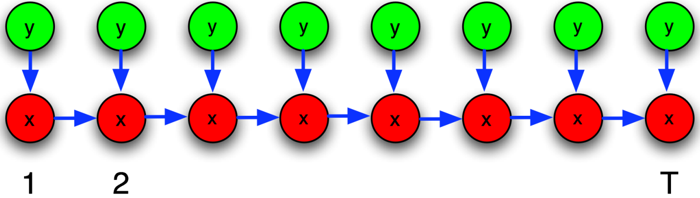
Rather than 2^n we now get $2^{|c|}$.

CRFs and HMMs

Conditional Random Field: maximize $p(y|x, \theta)$

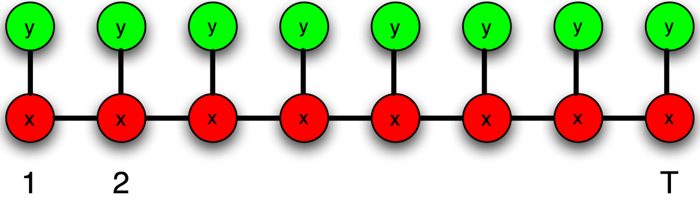


Hidden Markov Model: maximize $p(x, y|\theta)$

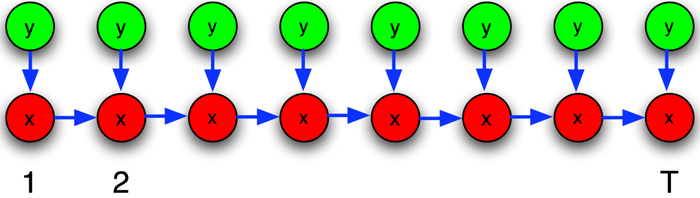


CRFs and HMMs

Conditional Random Field: maximize $p(y|x, \theta)$



Hidden Markov Model: maximize $p(x, y|\theta)$



Equivalence Theorem

Theorem

CRFs and HMMs yield identical probability estimates for $p(y|x, \theta)$, if the set of functions is equally expressive.

Proof.

- Write out $p_{\mathrm{CRF}}(y|x, \theta)$ and $p_{\mathrm{HMM}}(x, y|\theta)$, and show that they only differ in the normalization.
- This disappears when computing $p_{\mathrm{HMM}}(y|x, \theta)$.

□

Consequence

Differential training for current HMM implementations.

Equivalence Theorem

Theorem

CRFs and HMMs yield identical probability estimates for $p(y|x,\theta)$, if the set of functions is equally expressive.

Proof.

- Write out $p_{\mathrm{CRF}}(y|x,\theta)$ and $p_{\mathrm{HMM}}(x,y|\theta)$, and show that they only differ in the normalization.
- This disappears when computing $p_{\mathrm{HMM}}(y|x,\theta)$.

□

Consequence

Differential training for current HMM implementations.

Equivalence Theorem

Theorem

CRFs and HMMs yield identical probability estimates for $p(y|x,\theta)$, if the set of functions is equally expressive.

Proof.

- Write out $p_{\mathrm{CRF}}(y|x,\theta)$ and $p_{\mathrm{HMM}}(x,y|\theta)$, and show that they only differ in the normalization.
- This disappears when computing $p_{\mathrm{HMM}}(y|x,\theta)$.

□

Consequence

Differential training for current HMM implementations.

Mini Summary

Graphical Model Structure

- Same decomposition as in unconditional models.
- Only need to take cliques in y into account.

Kernel Expansion

- Representer theorem is still intractable (exponential number of terms).
- Decompose along cliques (we have a representer theorem per clique).
- For some parts primal space optimization is more efficient (cliques in y_i alone).

Connection to Hidden Markov Models

- HMMs optimize generative performance.
- CRFs optimize a discriminative model.
- Can re-optimize HMMs for discriminative performance.

Summary

1. **Conditional Models**
   - Log-partition Function, Densities, and Expectations
   - Inner Products and Kernels
   - Examples of Kernels
2. **Gaussian Process Classification**
   - Feature map
   - Examples
3. **Gaussian Process Regression**
   - Homoscedastic Model
   - Heteroscedastic Model
4. **Conditional Random Fields**
   - Model Structure
   - Kernel Expansion
   - Connections to Hidden Markov Models