Kernel Machines and Reinforcement Learning, Workshop of the 23rd ICML, Pittsburgh

Equi-gradient TD Learning

TD(λ) on adaptative Basis Functions Network

Manuel Loth,
Manuel Davy, Rémi Coulom, Philippe Preux

INRIA FUTURS *SequeL* Lille, France

June 29th 2006

Introduction

- Kernel methods
- Reinforcement learning

Introduction

- Kernel methods
  new ***Regression method*** $\sim$ kernel method
  (regularized sample-based linear approximation)

- Reinforcement learning

Introduction

- Kernel methods

  new ***Regression method*** $\sim$ kernel method
  (regularized sample-based linear approximation)

  equi-gradient descent

- Reinforcement learning

Introduction

- Kernel methods

  new ***Regression method*** $\sim$ kernel method
  (regularized sample-based linear approximation)

  equi-gradient descent

- Reinforcement learning

  Scheme for using it in ***TD(λ)***

Introduction

- Kernel methods
  new ***Regression method*** $\sim$ kernel method
  (regularized sample-based linear approximation)

  equi-gradient descent

- Reinforcement learning
  Scheme for using it in ***TD(λ)***

  equi-gradient TD(λ)

Regression

Regression

- unknown $f : \mathcal{X} \to \mathbb{R}$

Regression

- unknown $f : \mathcal{X} \rightarrow \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$

Regression

- unknown $f : \mathcal{X} \rightarrow \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

Regression

- unknown $f : \mathcal{X} \to \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

$$\mathbf{y} = (y_1, \ldots, y_n)^\mathsf{T}$$

$$\widehat{\mathbf{y}} = \left(\widehat{f}(x_1), \ldots, \widehat{f}(x_n)\right)^\mathsf{T}$$

Regression

- unknown $f : \mathcal{X} \to \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

$$\mathbf{y} = (y_1, \ldots, y_n)^\mathsf{T}$$

$$\widehat{\mathbf{y}} = \left(\widehat{f}(x_1), \ldots, \widehat{f}(x_n)\right)^\mathsf{T}$$

minimize $\mathcal{L}(\mathbf{y}, \widehat{\mathbf{y}})$

Regression

- unknown $f : \mathcal{X} \to \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

$$\mathbf{y} = (y_1, \ldots, y_n)^\mathsf{T}$$

$$\widehat{\mathbf{y}} = \left(\widehat{f}(x_1), \ldots, \widehat{f}(x_n)\right)^\mathsf{T}$$

$$\text{minimize } \mathcal{L}(\mathbf{y}, \widehat{\mathbf{y}}) + \lambda \left| \widehat{f} \right|_.$$

Regression

- unknown $f : \mathcal{X} \rightarrow \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

$$\mathbf{y} = (y_1, \ldots, y_n)^{\mathsf{T}}$$

$$\widehat{\mathbf{y}} = \left(\widehat{f}(x_1), \ldots, \widehat{f}(x_n)\right)^{\mathsf{T}}$$

$$\text{minimize } \mathcal{L}(\mathbf{y}, \widehat{\mathbf{y}}) + \lambda \left|\widehat{f}\right|_{.}$$

Regression

- unknown $f : \mathcal{X} \to \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

$$\mathbf{y} = (y_1, \ldots, y_n)^\mathsf{T}$$

$$\widehat{\mathbf{y}} = \left(\widehat{f}(x_1), \ldots, \widehat{f}(x_n)\right)^\mathsf{T}$$

$$\text{minimize } \mathcal{L}(\mathbf{y}, \widehat{\mathbf{y}}) + \lambda \left|\widehat{f}\right|_{\cdot}$$

kernel methods:

Regression

- unknown $f : \mathcal{X} \to \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

$$\mathbf{y} = (y_1, \ldots, y_n)^\mathsf{T}$$
$$\widehat{\mathbf{y}} = \left(\widehat{f}(x_1), \ldots, \widehat{f}(x_n)\right)^\mathsf{T}$$

$$\text{minimize } \mathcal{L}(\mathbf{y}, \widehat{\mathbf{y}}) + \lambda \left|\widehat{f}\right|_{.}$$

kernel methods:

- $k : \mathcal{X}^2 \to \mathbb{R} \sim$ similarity

Regression

- unknown $f : \mathcal{X} \rightarrow \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

$$\mathbf{y} = (y_1, \ldots, y_n)^\mathsf{T}$$
$$\widehat{\mathbf{y}} = \left(\widehat{f}(x_1), \ldots, \widehat{f}(x_n)\right)^\mathsf{T}$$

$$\text{minimize } \mathcal{L}(\mathbf{y}, \widehat{\mathbf{y}}) + \lambda \left| \widehat{f} \right|_.$$

kernel methods:

- $k : \mathcal{X}^2 \rightarrow \mathbb{R} \sim$ similarity
- 1 basis function / learning point: $\phi_i(x) = k(x_i, x)$

Regression

- unknown $f : \mathcal{X} \to \mathbb{R}$
- samples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- accurate and simple approximation $\widehat{f}$

$$\mathbf{y} = (y_1, \ldots, y_n)^\mathsf{T}$$
$$\widehat{\mathbf{y}} = \left(\widehat{f}(x_1), \ldots, \widehat{f}(x_n)\right)^\mathsf{T}$$

$$\text{minimize } \mathcal{L}(\mathbf{y}, \widehat{\mathbf{y}}) + \lambda \left|\widehat{f}\right|_{.}$$

kernel methods:

- $k : \mathcal{X}^2 \to \mathbb{R} \sim$ similarity
- 1 basis function / learning point: $\phi_i(x) = k(x_i, x)$
- $\widehat{f}(x) = \sum_i w_i \phi_i(x)$

regularized sample-based linear approximators

regularized sample-based linear approximators

- samples

regularized sample-based linear approximators

- samples $\rightsquigarrow$ basis functions $\phi_1, \ldots, \phi_m : \mathcal{X} \to \mathbb{R}$

regularized sample-based linear approximators

- samples $\rightsquigarrow$ basis functions $\phi_1, \ldots, \phi_m : \mathcal{X} \rightarrow \mathbb{R}$
- $\widehat{f}(x) = \sum_i w_i \phi_i(x)$

regularized sample-based linear approximators

- samples $\rightsquigarrow$ basis functions $\phi_1, \ldots, \phi_m : \mathcal{X} \to \mathbb{R}$
- $\widehat{f}(x) = \sum_i w_i \phi_i(x)$

$$\mathbf{\Phi} = \begin{pmatrix} \phi_1(x_1) & \ldots & \phi_m(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \ldots & \phi_m(x_n) \end{pmatrix}, \quad \mathbf{w} = (w_1, \ldots, w_m)^{\mathsf{T}}$$

regularized sample-based linear approximators

- samples $\rightsquigarrow$ basis functions $\phi_1, \ldots, \phi_m : \mathcal{X} \rightarrow \mathbb{R}$
- $\widehat{f}(x) = \sum_i w_i \phi_i(x)$

$$\mathbf{\Phi} = \begin{pmatrix} \phi_1(x_1) & \ldots & \phi_m(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \ldots & \phi_m(x_n) \end{pmatrix}, \quad \mathbf{w} = (w_1, \ldots, w_m)^\mathsf{T}$$

$$\widehat{\mathbf{y}} = \mathbf{\Phi}\mathbf{w}$$

regularized sample-based linear approximators

- samples $\rightsquigarrow$ basis functions $\phi_1, \dots, \phi_m : \mathcal{X} \to \mathbb{R}$
- $\widehat{f}(x) = \sum_i w_i \phi_i(x)$

$$\mathbf{\Phi} = \begin{pmatrix} \phi_1(x_1) & \dots & \phi_m(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \dots & \phi_m(x_n) \end{pmatrix}, \quad \mathbf{w} = \left(w_1, \dots, w_m\right)^{\mathsf{T}}$$

$$\widehat{\mathbf{y}} = \mathbf{\Phi w}$$

$$\mathcal{L}(\mathbf{y}, \mathbf{\Phi w}) + \lambda \left|\mathbf{\Phi w}\right|$$

LASSO

LASSO

$$\mathcal{L}(\mathbf{y}, \mathbf{\Phi w}) + \lambda \left| \mathbf{\Phi w} \right|$$

LASSO

$$(\mathbf{y} - \mathbf{\Phi w})^2 + \lambda \left| \mathbf{\Phi w} \right|$$

LASSO

$$(\mathbf{y} - \mathbf{\Phi w})^2 + \lambda \sum_i |w_i| \quad \text{(LASSO)}$$

LASSO

$$(\mathbf{y} - \mathbf{\Phi w})^2 + \lambda \sum_i |w_i| \quad \text{(LASSO)}$$

- Basis Pursuit Chen 95

LASSO

$$(\mathbf{y} - \mathbf{\Phi w})^2 + \lambda \sum_i |w_i| \quad \text{(LASSO)}$$

- Basis Pursuit Chen 95
- Adaptative Ridge Regression Granvalet 98

LASSO

$$(\mathbf{y} - \mathbf{\Phi w})^2 + \lambda \sum_i |w_i| \quad \text{(LASSO)}$$

- Basis Pursuit Chen 95
- Adaptative Ridge Regression Granvalet 98
- Matching Pursuit Mallat & Zhang 93

LASSO

$$(\mathbf{y} - \mathbf{\Phi w})^2 + \lambda \sum_i |w_i| \quad \text{(LASSO)}$$

- Basis Pursuit Chen 95
- Adaptative Ridge Regression Granvalet 98
- Matching Pursuit Mallat & Zhang 93
- iterative gradient descent Osborne *et al.* 00

Equi-gradient descent

Least-Angle Regression Stagewise/laSSO

Equi-gradient descent

Least-Angle Regression Stagewise/laSSO

- Efron 2002 variable selection

Equi-gradient descent

Least-Angle Regression Stagewise/laSSO

- Efron 2002 variable selection
- Guigue 2005 kernelization

Equi-gradient descent

Least-Angle Regression Stagewise/laSSO

- Efron 2002 variable selection
- Guigue 2005 kernelization
- Generalization, simplification $\longrightarrow$ ***Equi-gradient descent***

Equi-gradient descent

Equi-gradient descent

Given λ

Equi-gradient descent

Given λ
Given $\mathrm{sgn}(\mathbf{w}) = (0, 0, -1, 0, 1, \ldots)^\mathsf{T}$

Equi-gradient descent

Given λ
Given $\mathrm{sgn}(\mathbf{w}) = (0, 0, -1, 0, 1, \ldots)^{\mathsf{T}}$
$\mathbf{w}_{\mathrm{a}}$ = ***active*** (non-zero) weights

Equi-gradient descent

Given λ

Given $\text{sgn}(\mathbf{w}) = (0, 0, -1, 0, 1, \ldots)^{\mathsf{T}}$

$\mathbf{w}_{\mathrm{a}}$ = ***active*** (non-zero) weights

$\boldsymbol{\Phi}_{\mathrm{a}}$ = active basis functions

Equi-gradient descent

Given λ

Given $\mathrm{sgn}(\mathbf{w}) = (0, 0, -1, 0, 1, \ldots)^{\mathsf{T}}$

$\mathbf{w}_{\mathrm{a}}$ = ***active*** (non-zero) weights

$\mathbf{\Phi}_{\mathrm{a}}$ = active basis functions

$\widehat{\mathbf{y}} = \mathbf{\Phi}_{\mathrm{a}}\mathbf{w}_{\mathrm{a}}$

Equi-gradient descent

Given λ
Given $\mathrm{sgn}(\mathbf{w}) = (0, 0, -1, 0, 1, \ldots)^\mathsf{T}$
$\mathbf{w}_\mathrm{a}$ = ***active*** (non-zero) weights
$\mathbf{\Phi}_\mathrm{a}$ = active basis functions
$\widehat{\mathbf{y}} = \mathbf{\Phi}_\mathrm{a}\mathbf{w}_\mathrm{a}$

$$\text{minimize } (\mathbf{y} - \mathbf{\Phi}_\mathrm{a}\mathbf{w}_\mathrm{a})^2 + \lambda \sum_i |w_i|$$

Equi-gradient descent

Given λ
Given $\mathrm{sgn}(\mathbf{w}) = (0, 0, -1, 0, 1, \ldots)^{\mathsf{T}}$
$\mathbf{w}_{\mathrm{a}} =$ ***active*** (non-zero) weights
$\mathbf{\Phi}_{\mathrm{a}} =$ active basis functions
$\widehat{\mathbf{y}} = \mathbf{\Phi}_{\mathrm{a}}\mathbf{w}_{\mathrm{a}}$

$$\text{minimize } (\mathbf{y} - \mathbf{\Phi}_{\mathrm{a}}\mathbf{w}_{\mathrm{a}})^2 + \lambda \sum_i |w_i|$$

$$\mathbf{\Phi}_{\mathrm{a}}^{\mathsf{T}}(\mathbf{y} - \mathbf{\Phi}_{\mathrm{a}}\mathbf{w}_{\mathrm{a}}) = \lambda \mathbf{sgn}_{\mathrm{a}}$$

Equi-gradient descent

Given λ

Given $\text{sgn}(\mathbf{w}) = (0, 0, -1, 0, 1, \ldots)^T$

$\mathbf{w}_a$ = ***active*** (non-zero) weights

$\mathbf{\Phi}_a$ = active basis functions

$\widehat{\mathbf{y}} = \mathbf{\Phi}_a \mathbf{w}_a$

$$\text{minimize } (\mathbf{y} - \mathbf{\Phi}_a \mathbf{w}_a)^2 + \lambda \sum_i |w_i|$$

$$\mathbf{\Phi}_a^T (\mathbf{y} - \mathbf{\Phi}_a \mathbf{w}_a) = \lambda \mathbf{sgn}_a$$

$$\text{for all active } w_i, \qquad \left| \frac{\partial \mathcal{L}}{\partial w_i} \right| = \lambda$$

Equi-gradient descent

Given λ
Given $\mathrm{sgn}(\mathbf{w}) = (0, 0, -1, 0, 1, \ldots)^{\mathsf{T}}$
$\mathbf{w}_{\mathrm{a}}$ = ***active*** (non-zero) weights
$\mathbf{\Phi}_{\mathrm{a}}$ = active basis functions
$\widehat{\mathbf{y}} = \mathbf{\Phi}_{\mathrm{a}}\mathbf{w}_{\mathrm{a}}$

$$\text{minimize } (\mathbf{y} - \mathbf{\Phi}_{\mathrm{a}}\mathbf{w}_{\mathrm{a}})^2 + \lambda \sum_i |w_i|$$

$$\mathbf{\Phi}_{\mathrm{a}}^{\mathsf{T}}(\mathbf{y} - \mathbf{\Phi}_{\mathrm{a}}\mathbf{w}_{\mathrm{a}}) = \lambda \mathbf{sgn}_{\mathrm{a}}$$

$$\text{for all active } w_i, \qquad \left|\frac{\partial \mathcal{L}}{\partial w_i}\right| = \lambda$$

$$\text{for all inactive } w_i, \qquad \left|\frac{\partial \mathcal{L}}{\partial w_i}\right| \leq \lambda$$

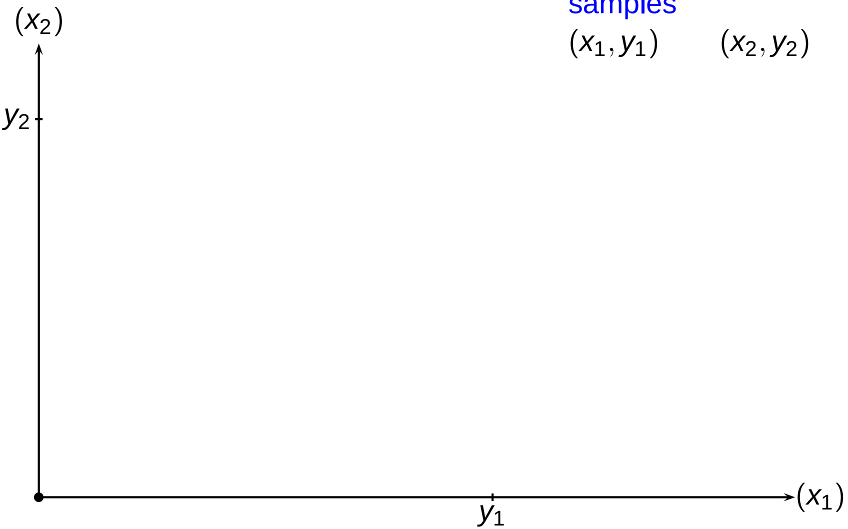
Regularization path

Regularization path

samples

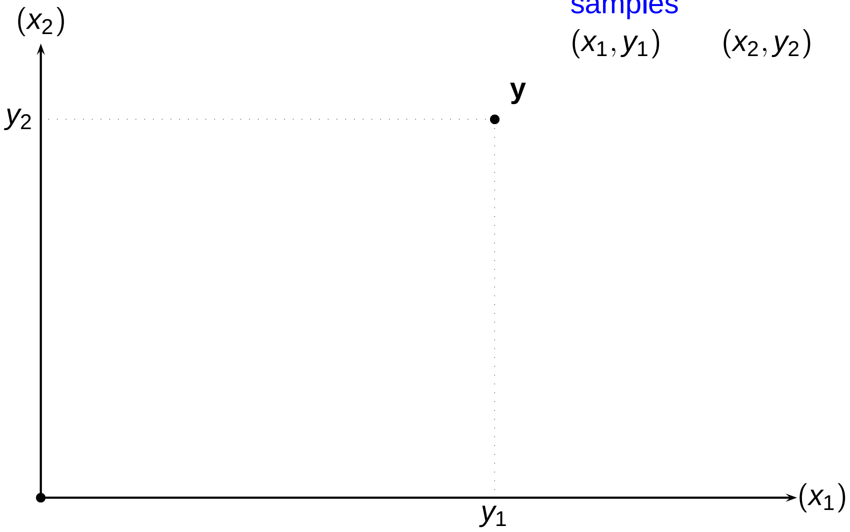
(x_1, y_1) (x_2, y_2)

Regularization path



samples
(x_1, y_1) (x_2, y_2)

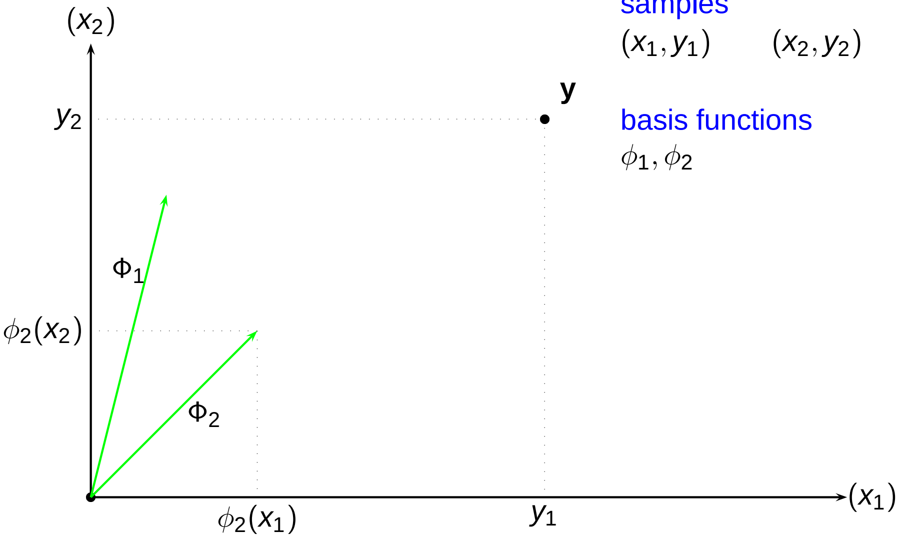
Regularization path



samples

(x_1, y_1) (x_2, y_2)

Regularization path



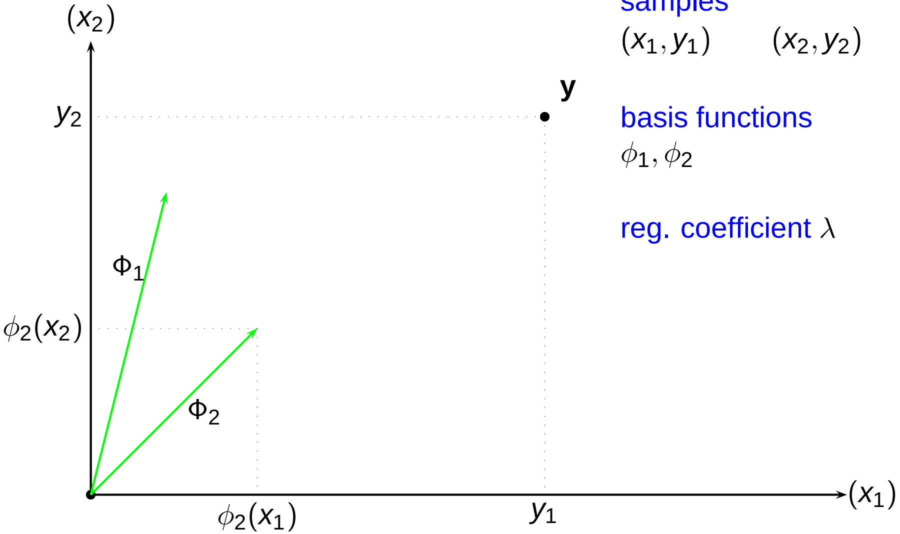
samples

(x_1, y_1) (x_2, y_2)

basis functions

ϕ_1, ϕ_2

Regularization path



samples

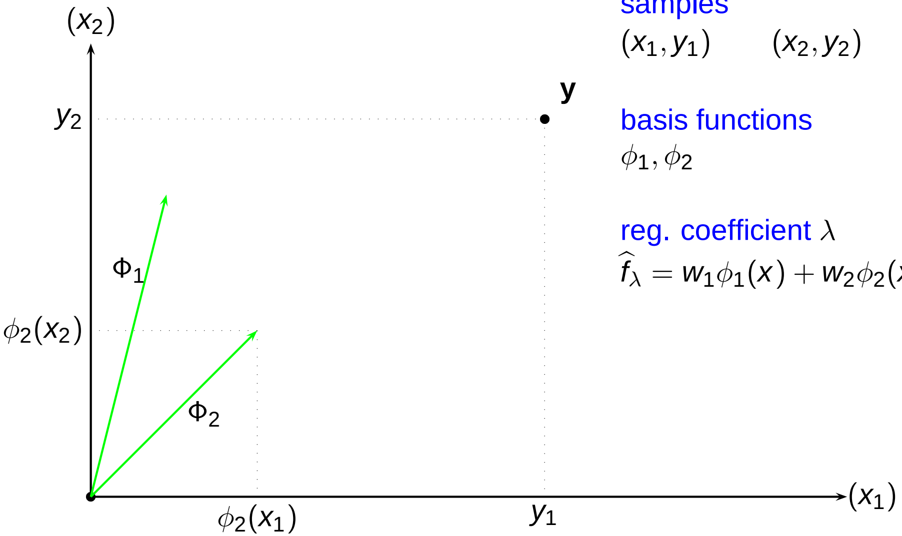
(x_1, y_1) (x_2, y_2)

basis functions

ϕ_1, ϕ_2

reg. coefficient λ

Regularization path



samples

$(x_1, y_1) \qquad (x_2, y_2)$

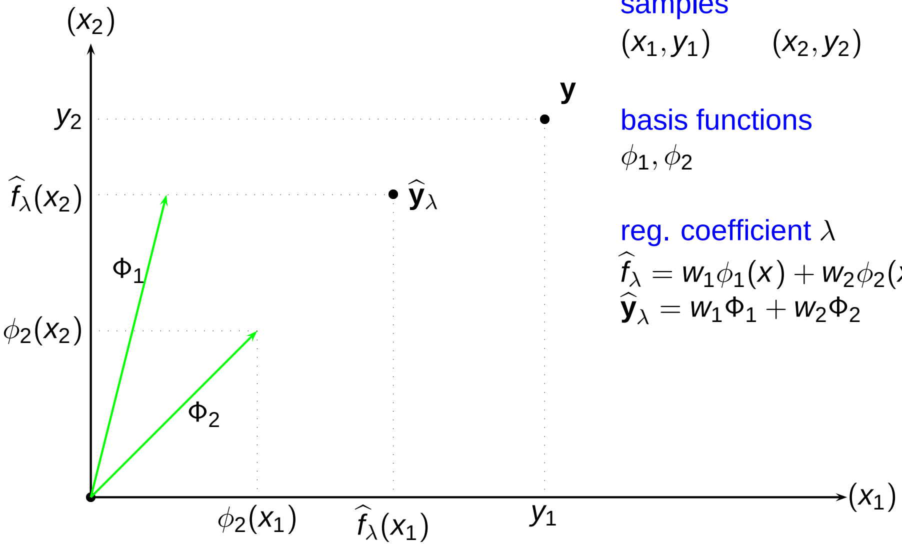
basis functions

ϕ_1, ϕ_2

reg. coefficient λ

$\widehat{f}_\lambda = w_1\phi_1(x) + w_2\phi_2(x)$

Regularization path



samples

$(x_1, y_1) \qquad (x_2, y_2)$

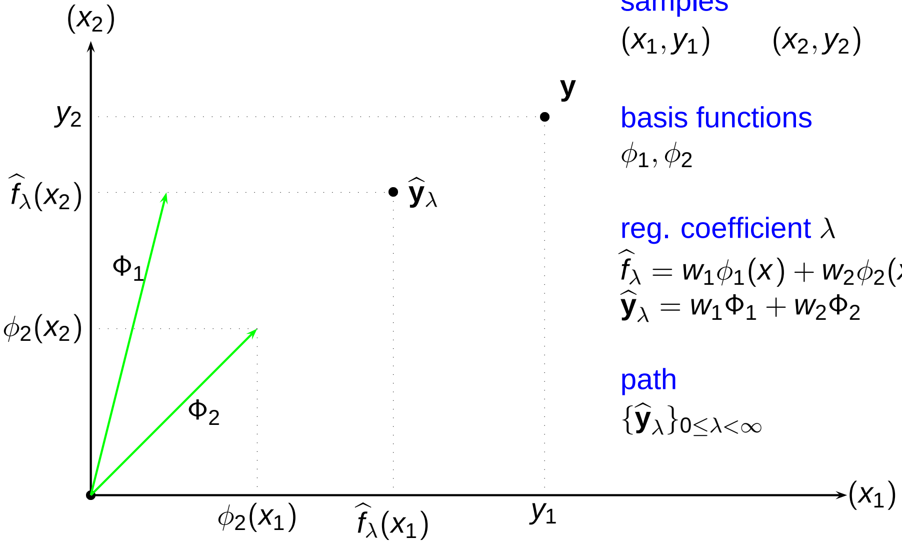
basis functions

ϕ_1, ϕ_2

reg. coefficient λ

$$\widehat{f}_\lambda = w_1 \phi_1(x) + w_2 \phi_2(x)$$
$$\widehat{\mathbf{y}}_\lambda = w_1 \Phi_1 + w_2 \Phi_2$$

Regularization path



samples

(x_1, y_1) $\quad$ (x_2, y_2)

basis functions

ϕ_1, ϕ_2

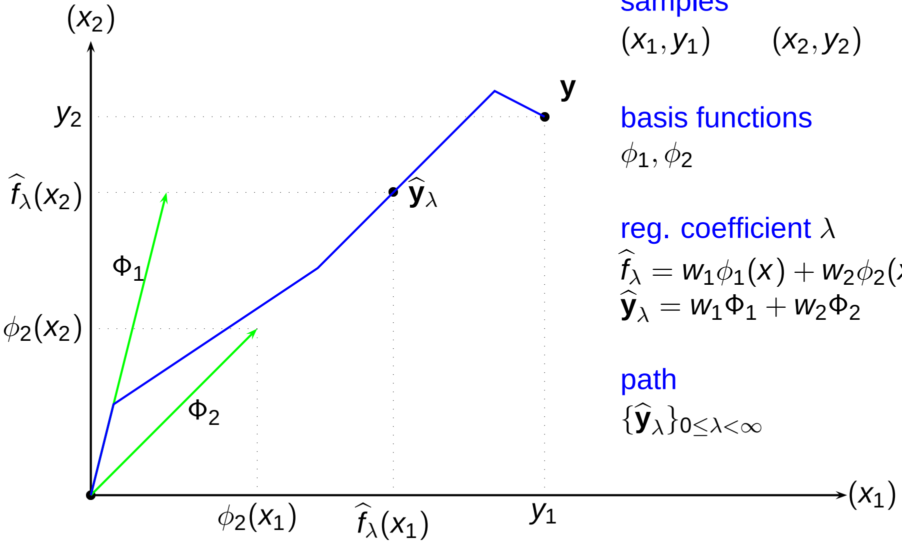
reg. coefficient λ

$$\widehat{f}_\lambda = w_1\phi_1(x) + w_2\phi_2(x)$$
$$\widehat{\mathbf{y}}_\lambda = w_1\Phi_1 + w_2\Phi_2$$

path

$\{\widehat{\mathbf{y}}_\lambda\}_{0\leq\lambda<\infty}$

Regularization path



samples

$(x_1, y_1) \qquad (x_2, y_2)$

basis functions

ϕ_1, ϕ_2

reg. coefficient λ

$\widehat{f}_\lambda = w_1\phi_1(x) + w_2\phi_2(x)$

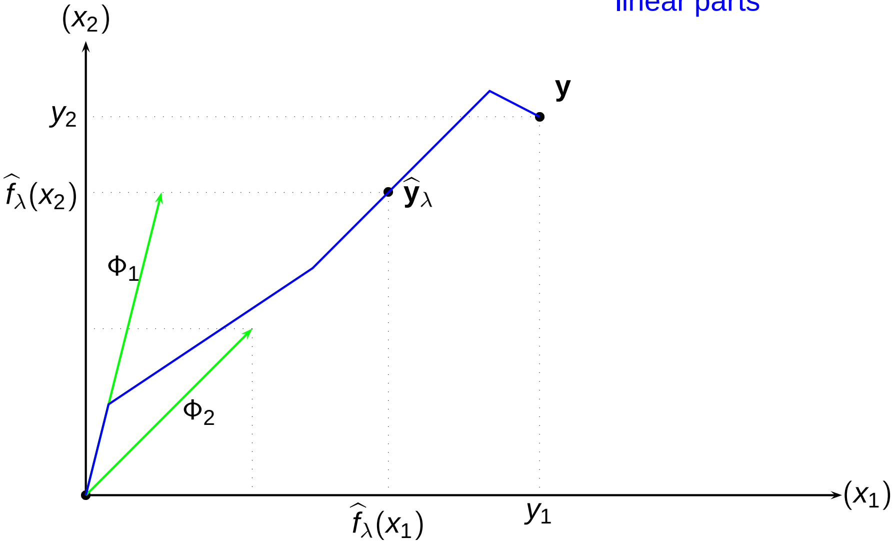
$\widehat{\mathbf{y}}_\lambda = w_1\Phi_1 + w_2\Phi_2$

path

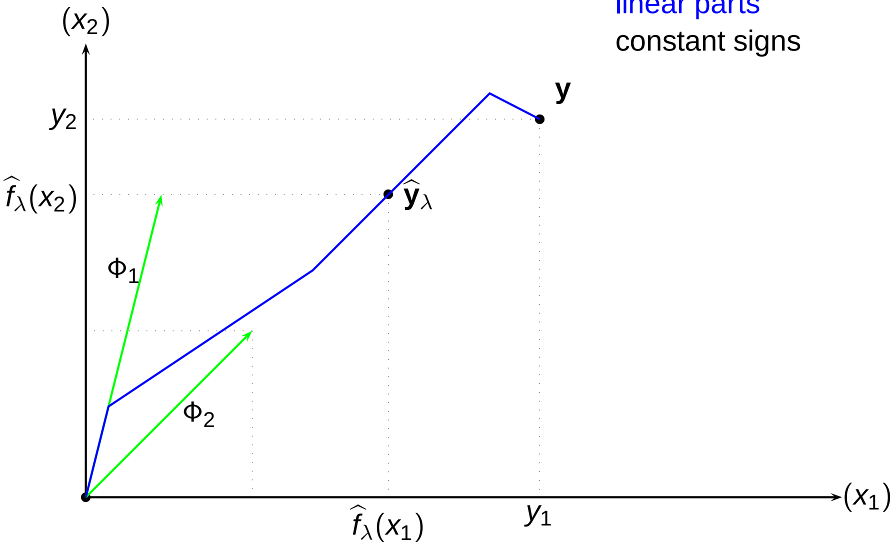
$\{\widehat{\mathbf{y}}_\lambda\}_{0\leq\lambda<\infty}$

Regularization path

linear parts



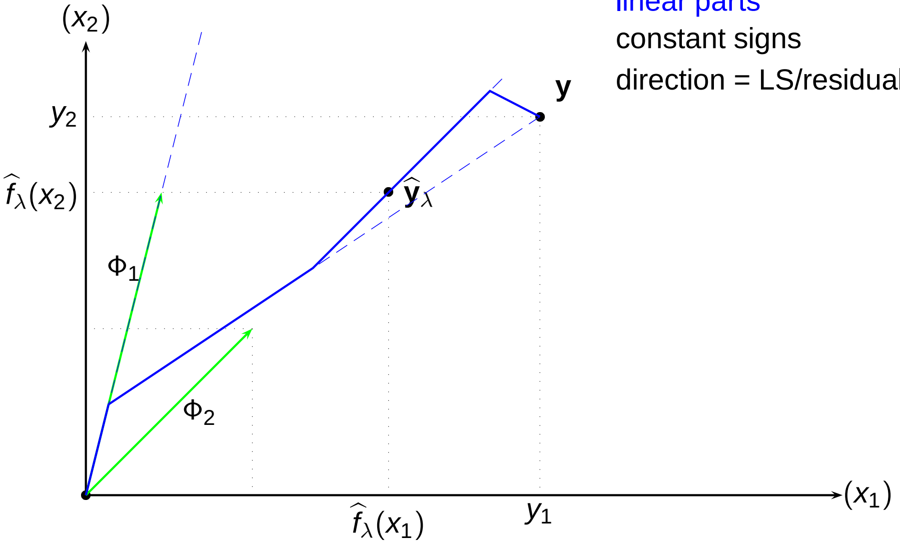
Regularization path



linear parts

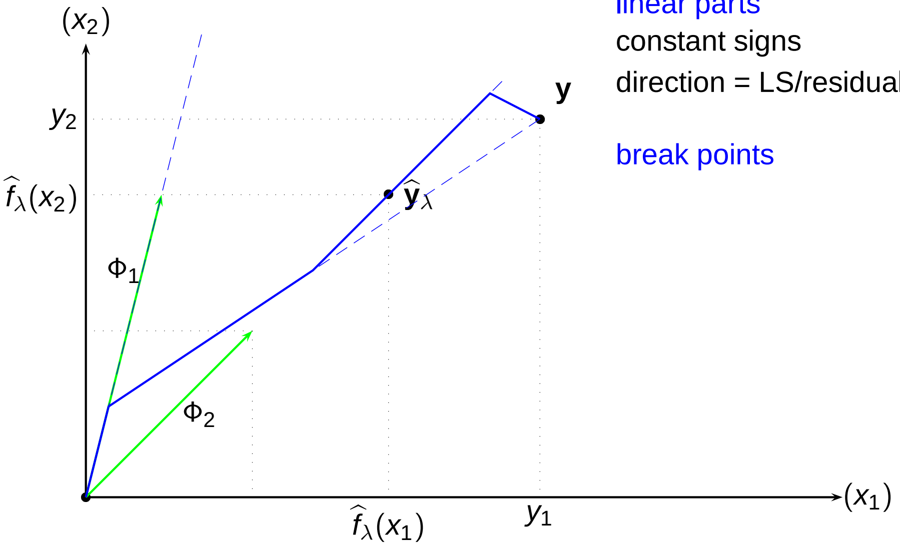
constant signs

Regularization path



linear parts
constant signs
direction = LS/residual

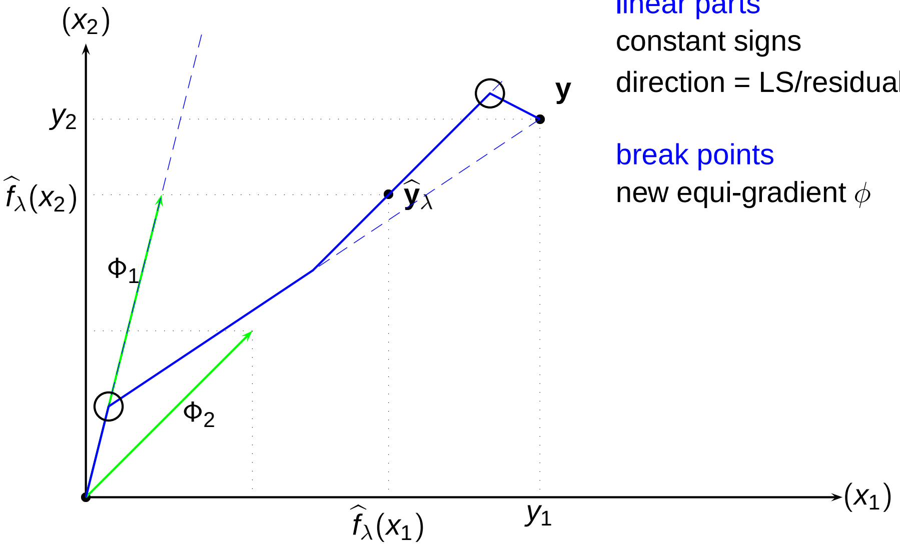
Regularization path



linear parts
constant signs
direction = LS/residual

break points

Regularization path



linear parts

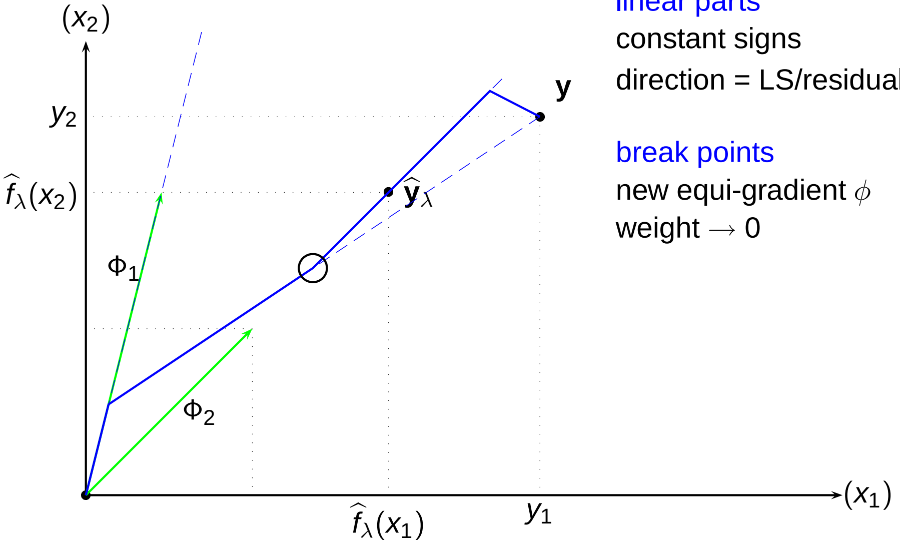
constant signs

direction = LS/residual

break points

new equi-gradient ϕ

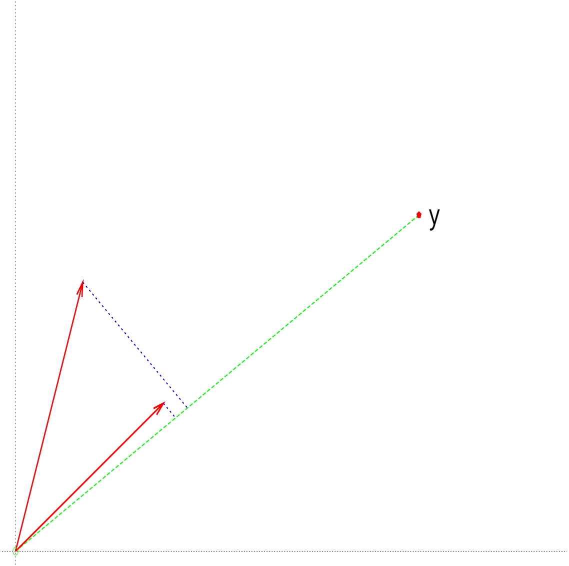
Regularization path



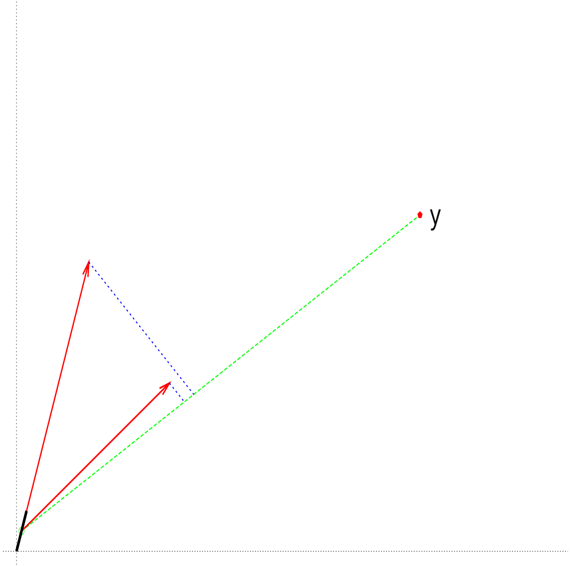
linear parts
constant signs
direction = LS/residual

break points
new equi-gradient ϕ
weight $\rightarrow$ 0

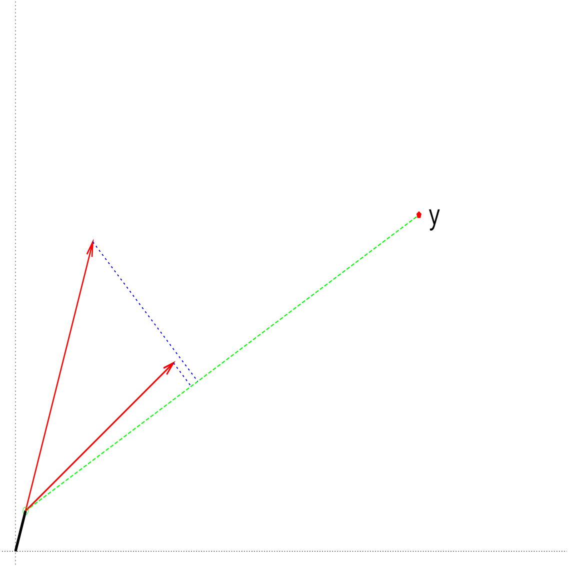
Regularization path



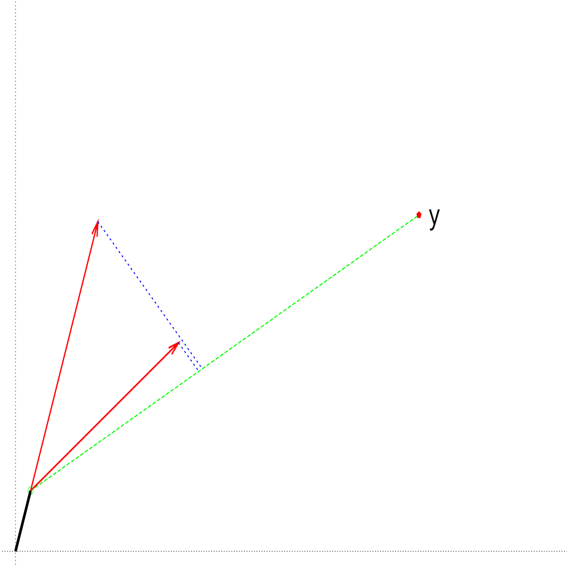
Regularization path



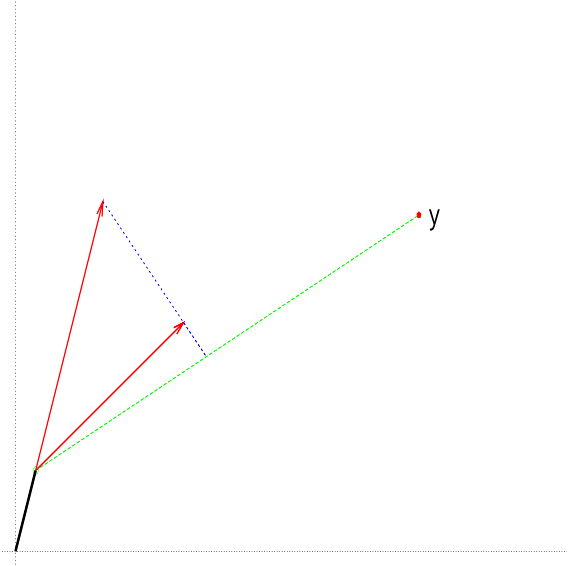
Regularization path



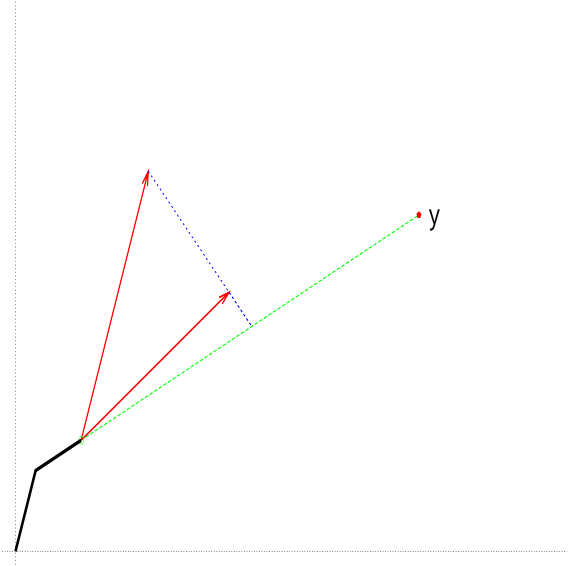
Regularization path



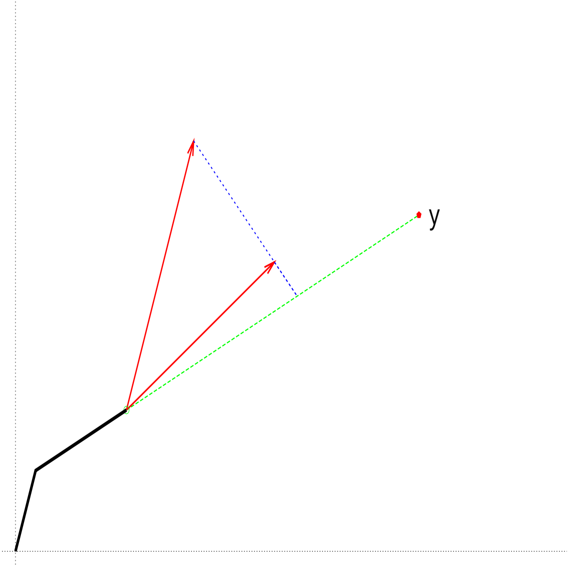
Regularization path



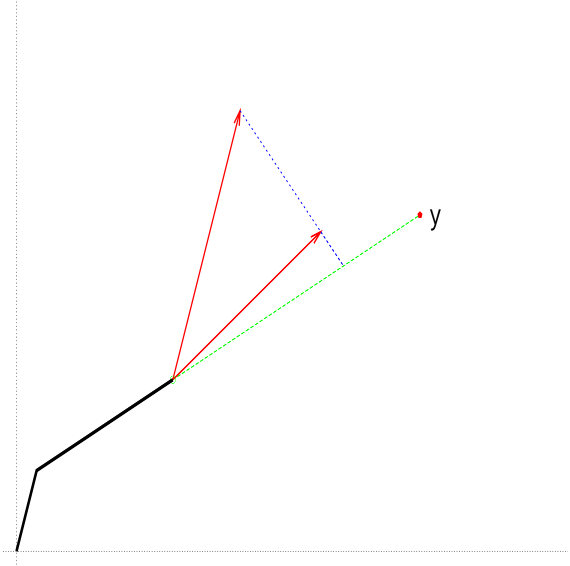
Regularization path



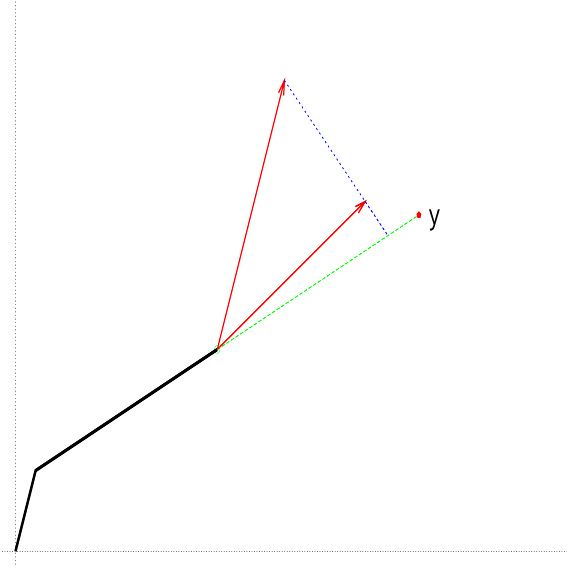
Regularization path



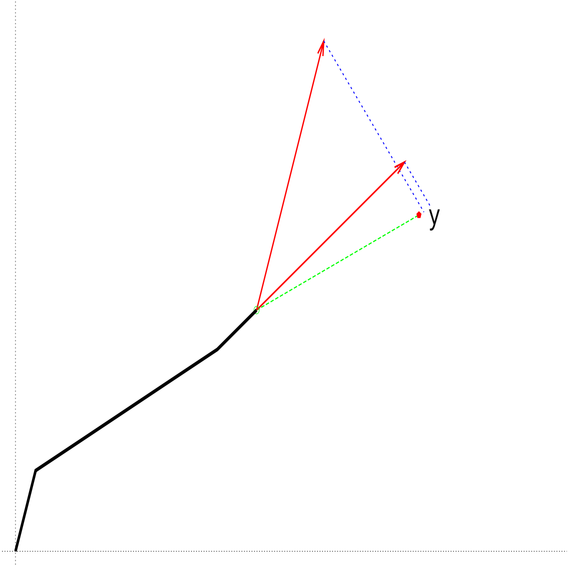
Regularization path



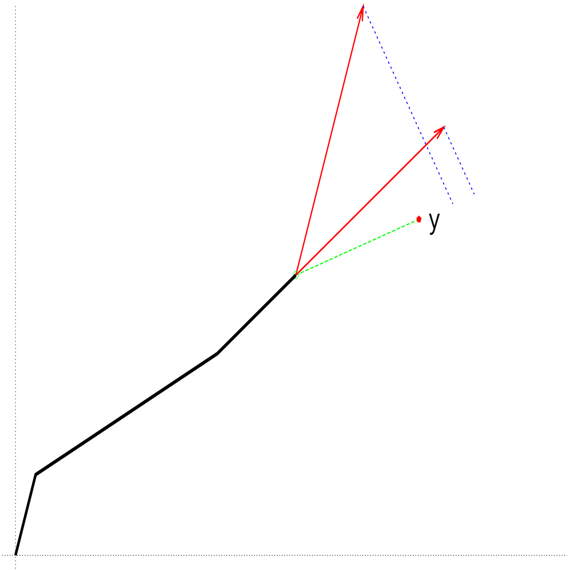
Regularization path



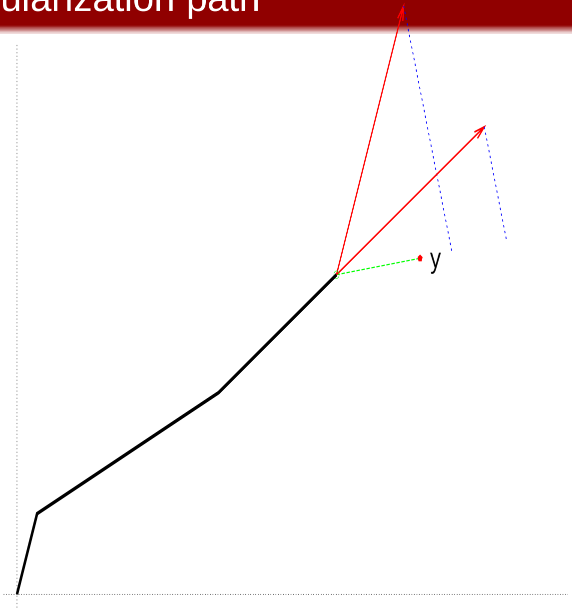
Regularization path



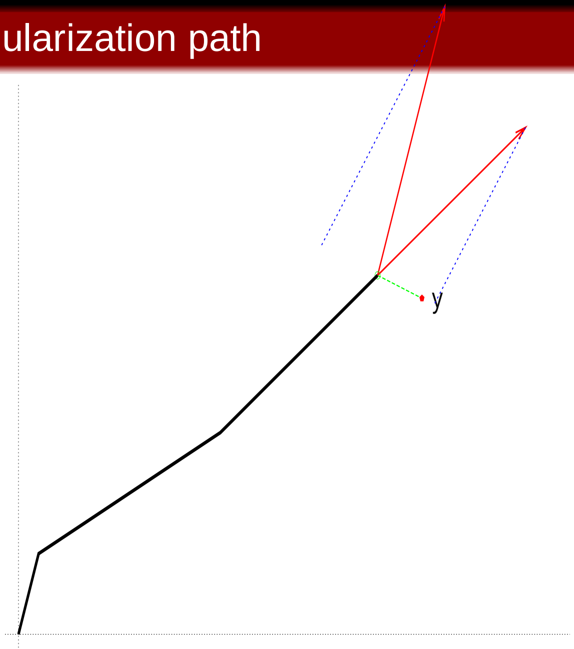
Regularization path



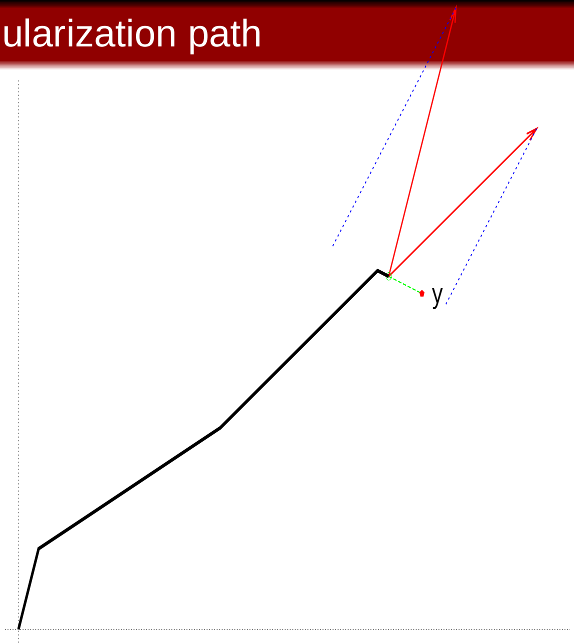
Regularization path



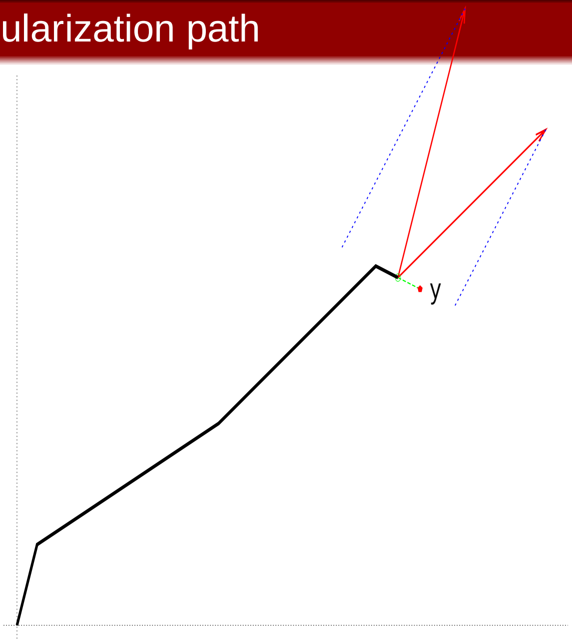
Regularization path



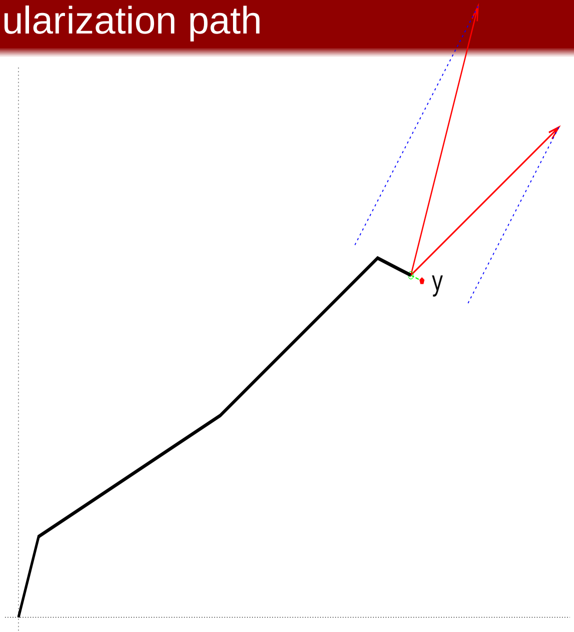
Regularization path



Regularization path



Regularization path



Regularization path

Complexity

Each step (activation/deactivation) is $O(|\mathcal{D}| + |\mathcal{A}|^2)$

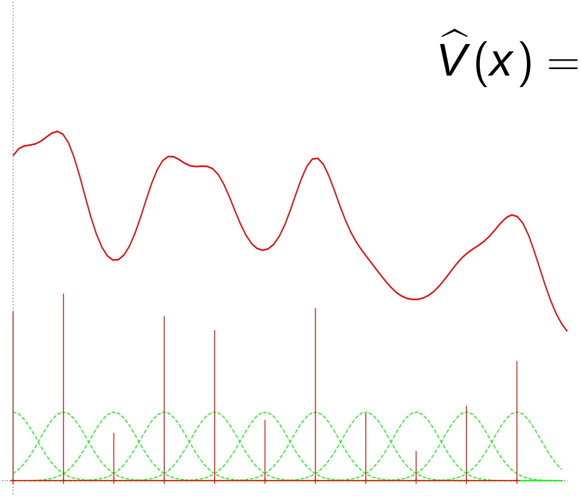
- linear in dictionary size
- quadratic in number of active basis functions

Demo

Reinforcement Learning

- TD(λ)
- continuous state space
- discrete time
- updates after each trajectory

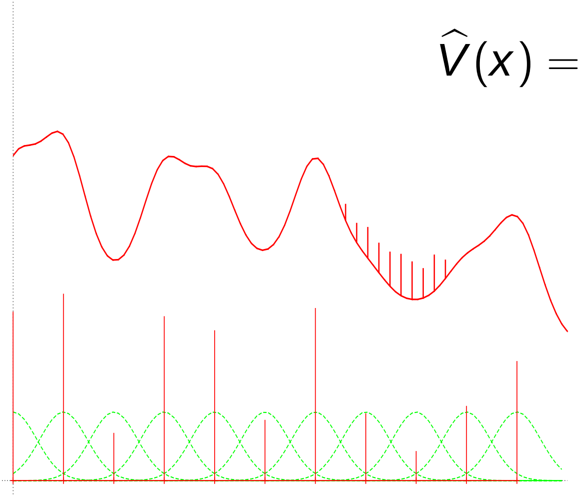
Radial Basis Functions Network



$$\widehat{V}(x) = \sum w_i \phi_i(x)$$

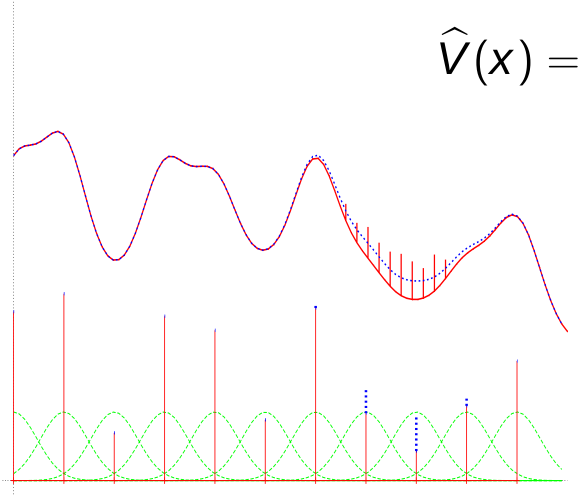
Radial Basis Functions Network

$$\widehat{V}(x) = \sum w_i \phi_i(x)$$

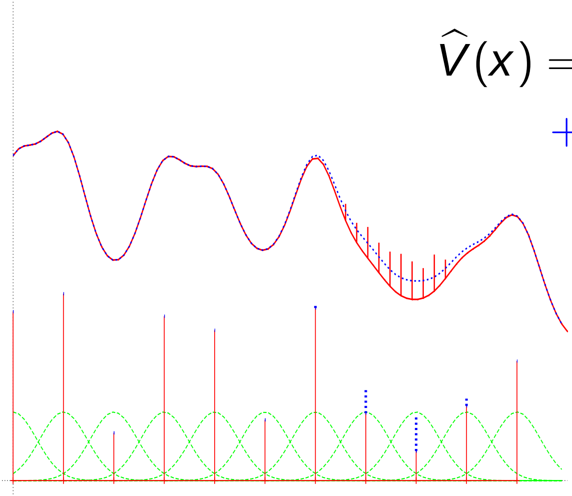


Update

$$\widehat{V}(x) = \sum (w_i + \Delta_{w_i}) \phi_i(x)$$

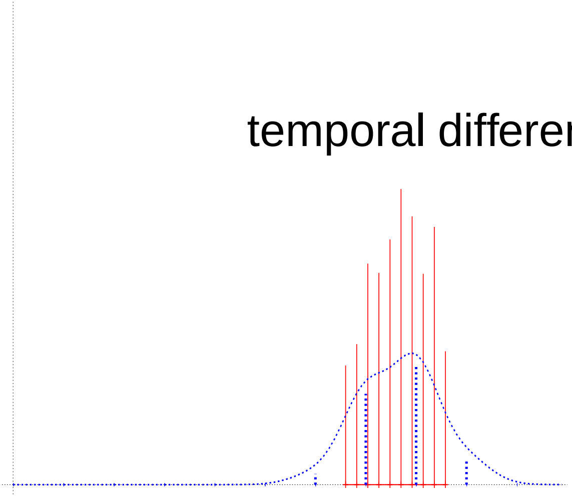


Update



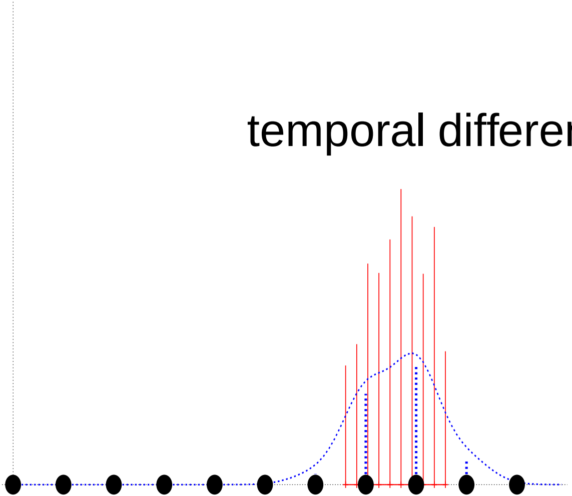
$$\hat{V}(x) = \sum w_i \phi_i(x) + \sum \Delta_{w_i} \phi_i(x)$$

Independant regression

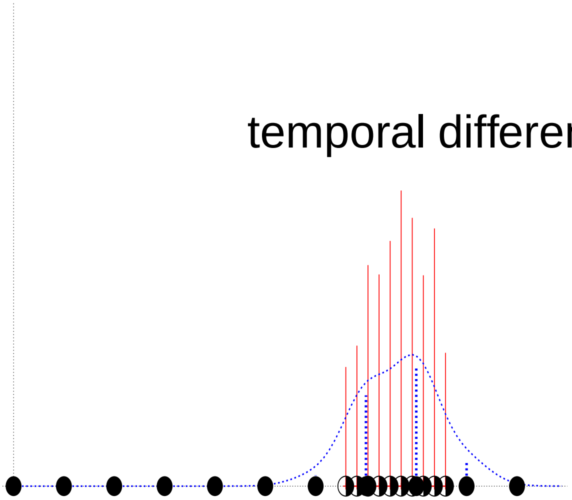


Gradient descent on fixed basis network

temporal differences



Equi-gradient descent on extended network



Temporal regularization

Temporal regularization

Continuously add basis functions?!

Temporal regularization

Continuously add basis functions?!

- Put a preference on existing basis functions

Temporal regularization

Continuously add basis functions?!

- Put a preference on existing basis functions
- Remove zero-weighted basis functions

Temporal regularization

Continuously add basis functions?!

- Put a preference on existing basis functions
  $\phi_i \leftarrow \rho\phi_i$
- Remove zero-weighted basis functions

Temporal regularization

Continuously add basis functions?!

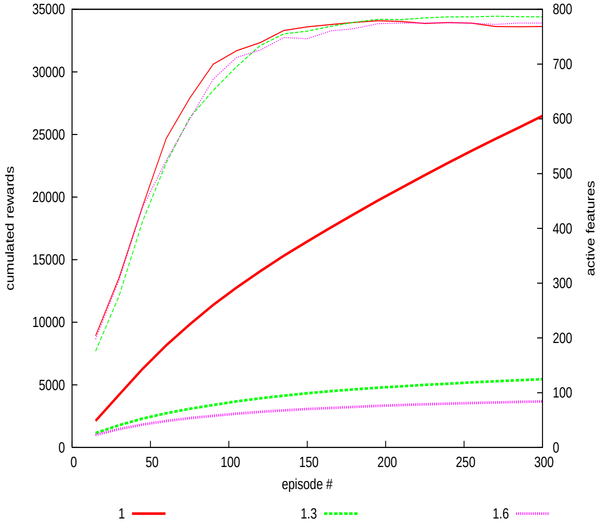
- Put a preference on existing basis functions
  $\phi_i \leftarrow \rho\phi_i \Rightarrow w_i \leftarrow \frac{1}{\rho}w_i$
- Remove zero-weighted basis functions

Preliminary simple experiments

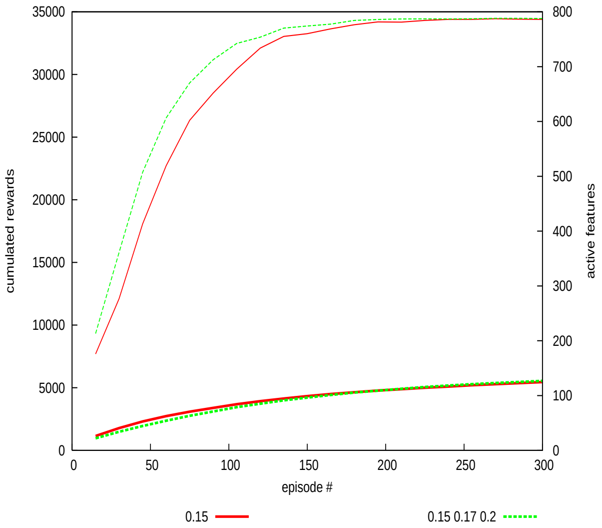
Preliminary simple experiments

- Inverted pendulum
- Gaussian basis functions on normalized state space
- Updates after each episode
- Stopping EG descents at $|\widehat{\mathbf{y}}|^2 = 70\%|\mathbf{y}|^2$

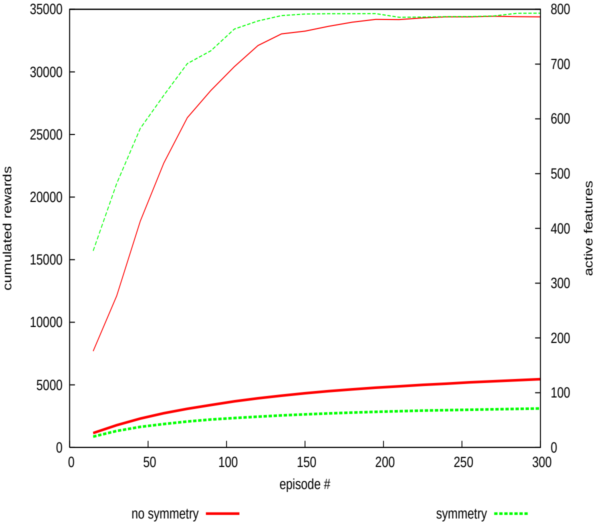
Experiments: preference



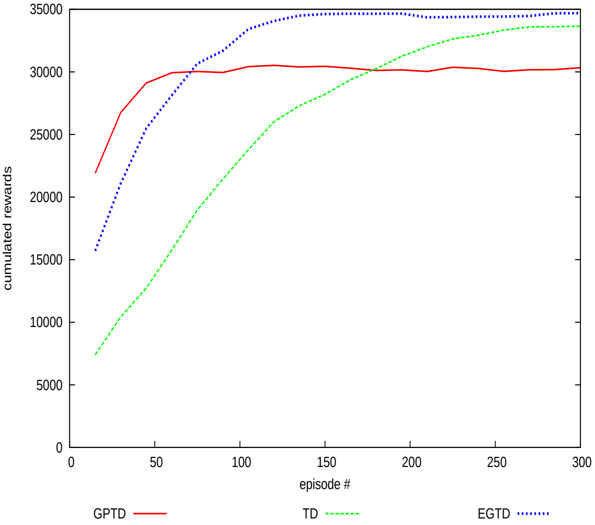
Experiments: multi-kernels



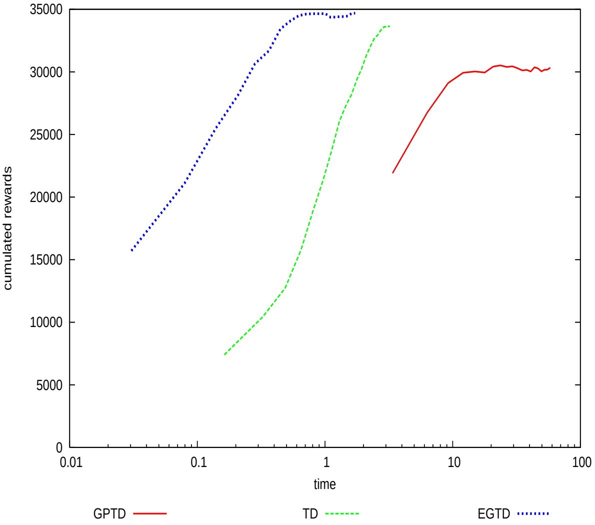
Experiments: symmetry



Experiments: comparison



Experiments: comparison



Conclusion

Conclusion

Summary:

Conclusion

Summary:

- ***efficient*** and ***easy*** way to select basis functions in TD(λ)

Conclusion

Summary:

- ***efficient*** and ***easy*** way to select basis functions in TD(λ)
- ***robust***, no unintuitive parameters

Conclusion

Summary:

- ***efficient*** and ***easy*** way to select basis functions in TD(λ)
- ***robust***, no unintuitive parameters

Perspectives:

Conclusion

Summary:

- ***efficient*** and ***easy*** way to select basis functions in TD(λ)
- ***robust***, no unintuitive parameters

Perspectives:

- experiments on other problems

Conclusion

Summary:

- ***efficient*** and ***easy*** way to select basis functions in TD(λ)
- ***robust***, no unintuitive parameters

Perspectives:

- experiments on other problems
- automatically build basis function dictionary based on topology, TD variance, . . . (wavelets, low-dimensional projections,. . .)

Take home message

- Feature selection is easy!
- Basic use of it in RL $\longrightarrow$ efficient & easy-to-tune TD(λ)