Kernel Learning with a Million Kernels

Ashesh Jain
IIT Delhi

SVN Vishwanathan
Purdue University

Manik Varma
Microsoft Research India

Kernel Learning

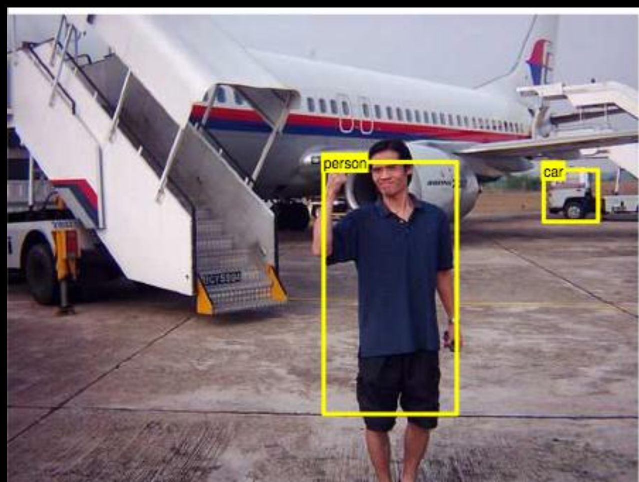
- The objective in kernel learning is to jointly learn both SVM and kernel parameters from training data.

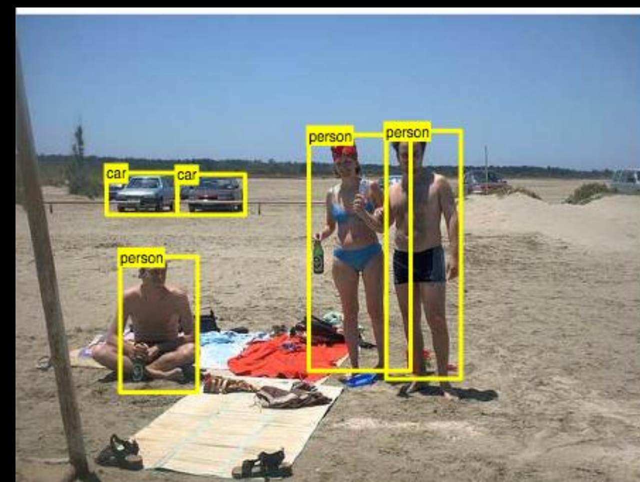
- Kernel parameterizations
  - Linear : $K = \sum_i d_i K_i$
  - Non-linear : $K = \prod_i K_i = \prod_i e^{-d_i D_i}$

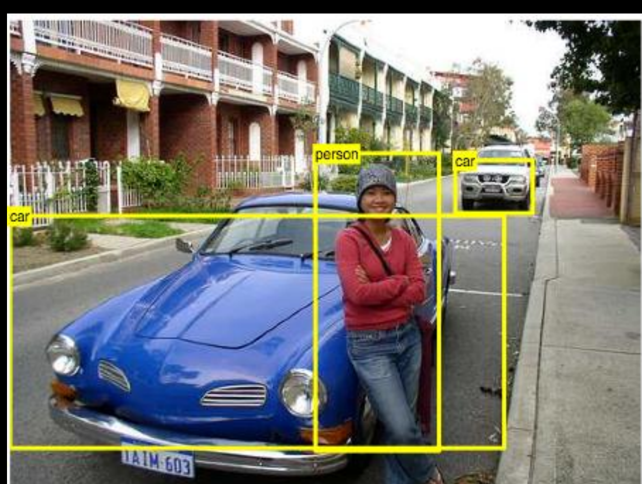
- Regularizers
  - Sparse l_1
  - Sparse and non-sparse $l_{p>1}$
  - Log determinant

Kernel Learning for Object Detection

- Vedaldi, Gulshan, Varma and Zisserman ICCV 2009

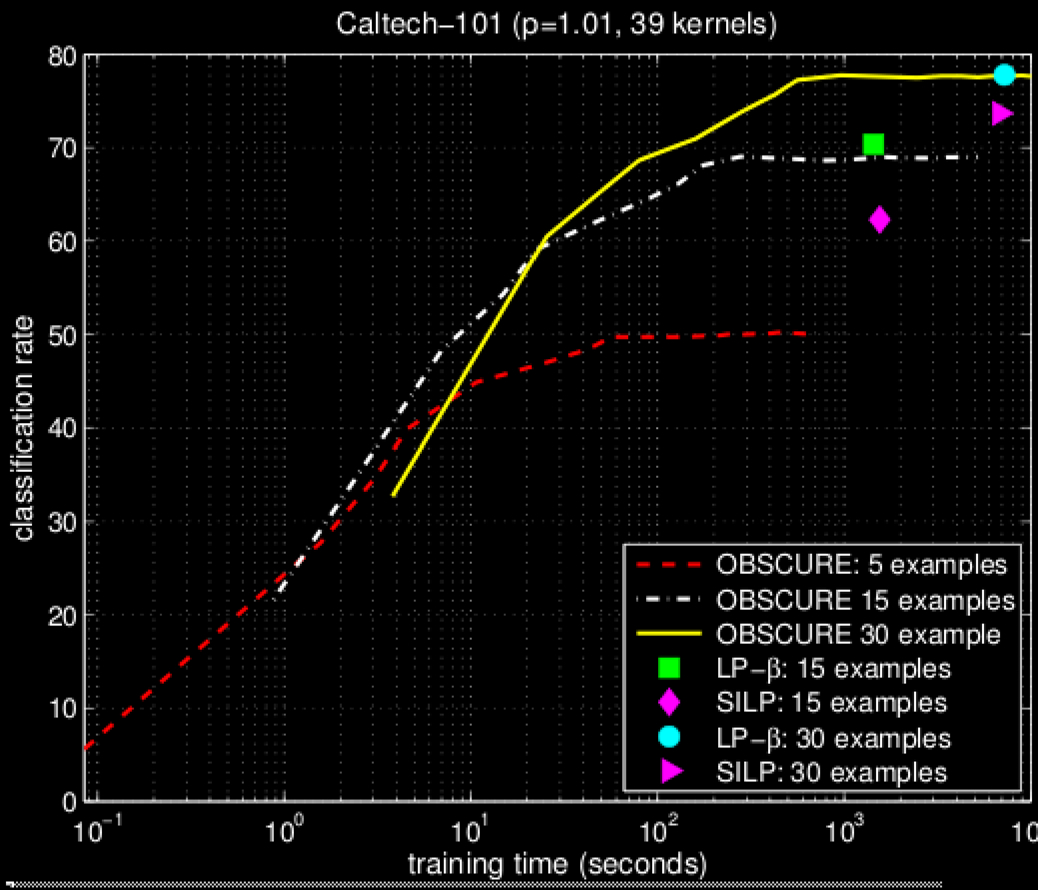






Kernel Learning for Object Recognition

- Orabona, Jie and Caputo CVPR 2010



Kernel Learning for Feature Selection

- Varma and Babu ICML 2009

FERET Gender Identification Data Set

# Feat	AdaBoost	Baluja *et al.* [IJCV 2007]	OWL-QN [ICML 2007]	LP-SVM [COA 2004]	SSVM QCQP [ICML 2007]	BAHSIC [ICML 2007]	Linear MKL	Non-Linear MKL
10	76.3 ± 0.9	79.5 ± 1.9	71.6 ± 1.4	84.9 ± 1.9	79.5 ± 2.6	81.2 ± 3.2	80.8 ± 0.2	**88.7 ± 0.8**
20	-	82.6 ± 0.6	80.5 ± 3.3	87.6 ± 0.5	85.6 ± 0.7	86.5 ± 1.3	83.8 ± 0.7	**93.2 ± 0.9**
30	-	83.4 ± 0.3	84.8 ± 0.4	89.3 ± 1.1	88.6 ± 0.2	89.4 ± 2.4	86.3 ± 1.6	**95.1 ± 0.5**
50	-	86.9 ± 1.0	88.8 ± 0.4	90.6 ± 0.6	89.5 ± 0.2	91.0 ± 1.3	89.4 ± 0.9	**95.5 ± 0.7**
80	-	88.9 ± 0.6	90.4 ± 0.2	-	90.6 ± 1.1	92.4 ± 1.4	90.5 ± 0.2	**-**
100	-	89.5 ± 0.2	90.6 ± 0.3	-	90.5 ± 0.2	94.1 ± 1.3	91.3 ± 1.3	**-**
150	-	91.3 ± 0.5	90.3 ± 0.8	-	90.7 ± 0.2	94.5 ± 0.7	-	**-**
252	-	93.1 ± 0.5	-	-	90.8 ± 0.0	94.3 ± 0.1	-	**-**
	76.3(12.6)	-	91 (221.3)	91 (58.3)	90.8 (252)	-	91.6(146.3)	**95.5 (69.6)**

The GMKL Primal Formulation

$P = \text{Min}_{\mathbf{w},b,\mathbf{d}} \;\; \frac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_i L(\mathbf{w}^t\boldsymbol{\phi}_{\mathbf{d}}(\mathbf{x}_i) + b, y_i) + r(\mathbf{d})$

$\text{s. t.} \;\; \mathbf{d} \in \boldsymbol{D}$

- $K_{\mathbf{d}}(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}_{\mathbf{d}}^t(\mathbf{x}_i)\boldsymbol{\phi}_{\mathbf{d}}(\mathbf{x}_j) \succ 0 \;\; \forall \mathbf{d} \in D$
- $\nabla_{\mathbf{d}} K$ and $\nabla_{\mathbf{d}} r$ exist and are continuous

The GMKL Primal Formulation

- The GMKL primal formulation for binary classification.

$P = \text{Min}_{\mathbf{w},b,\mathbf{d},\xi}$ $\frac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_i \xi_i + r(\mathbf{d})$

s. t. $y_i(\mathbf{w}^t\boldsymbol{\phi}_{\mathbf{d}}(\mathbf{x}_i) + b) \geq 1 - \xi_i$

$\xi_i \geq 0$ & $\mathbf{d} \in \boldsymbol{D}$

The GMKL Primal Formulation

- The GMKL primal formulation for binary classification.

$$P = \text{Min}_{\mathbf{w},b,\mathbf{d},\xi} \quad \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_i \xi_i + r(\mathbf{d})$$

$$\text{s. t.} \quad y_i(\mathbf{w}^t\boldsymbol{\phi}_{\mathbf{d}}(\mathbf{x}_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0 \ \& \ \mathbf{d} \in \boldsymbol{D}$$

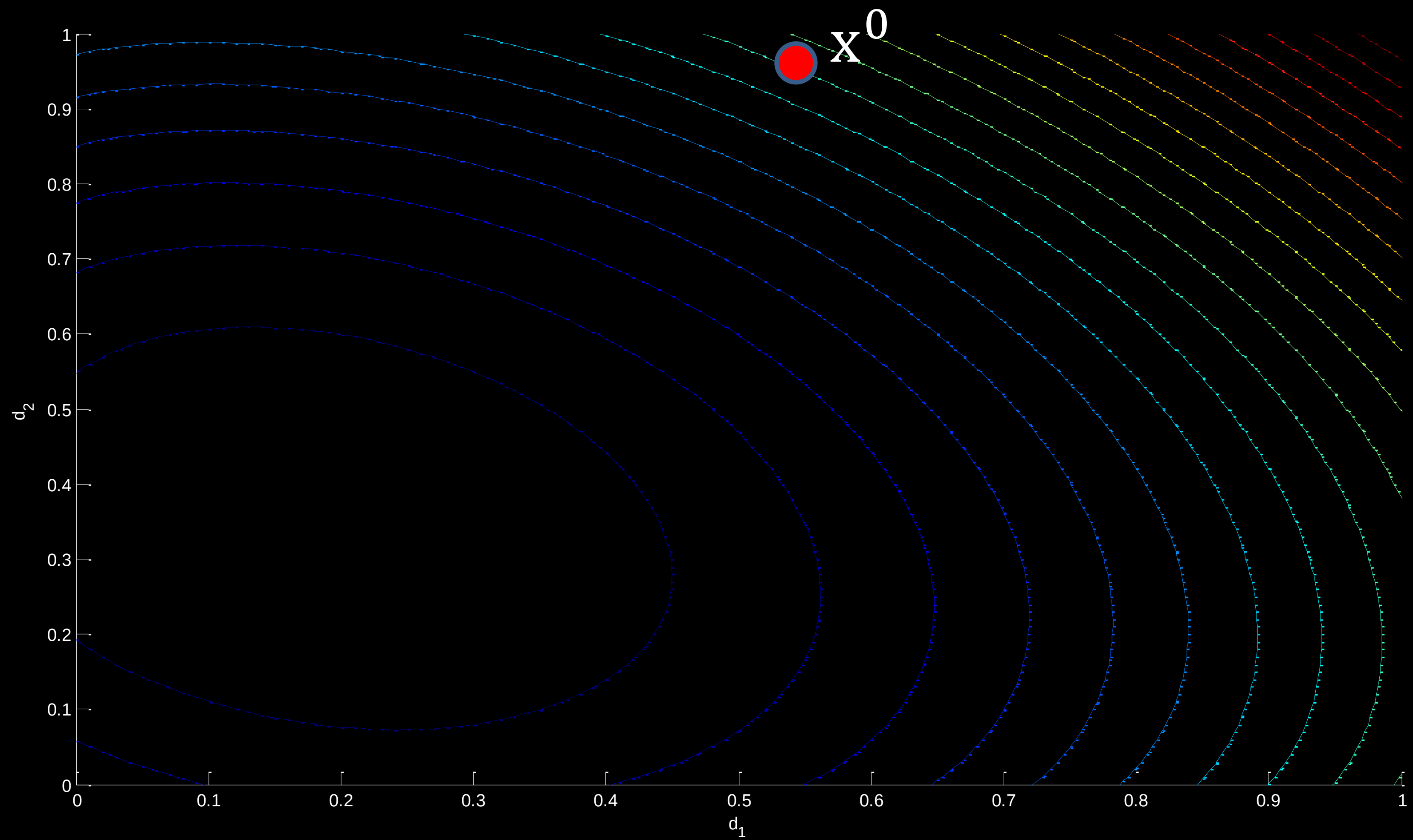
- Intermediate Dual

$$D = \text{Min}_{\mathbf{d}} \, \text{Max}_{\alpha} \quad \mathbf{1}^t\alpha - \tfrac{1}{2}\alpha^t\mathbf{Y}\mathbf{K}_{\mathbf{d}}\mathbf{Y}\alpha + r(\mathbf{d})$$

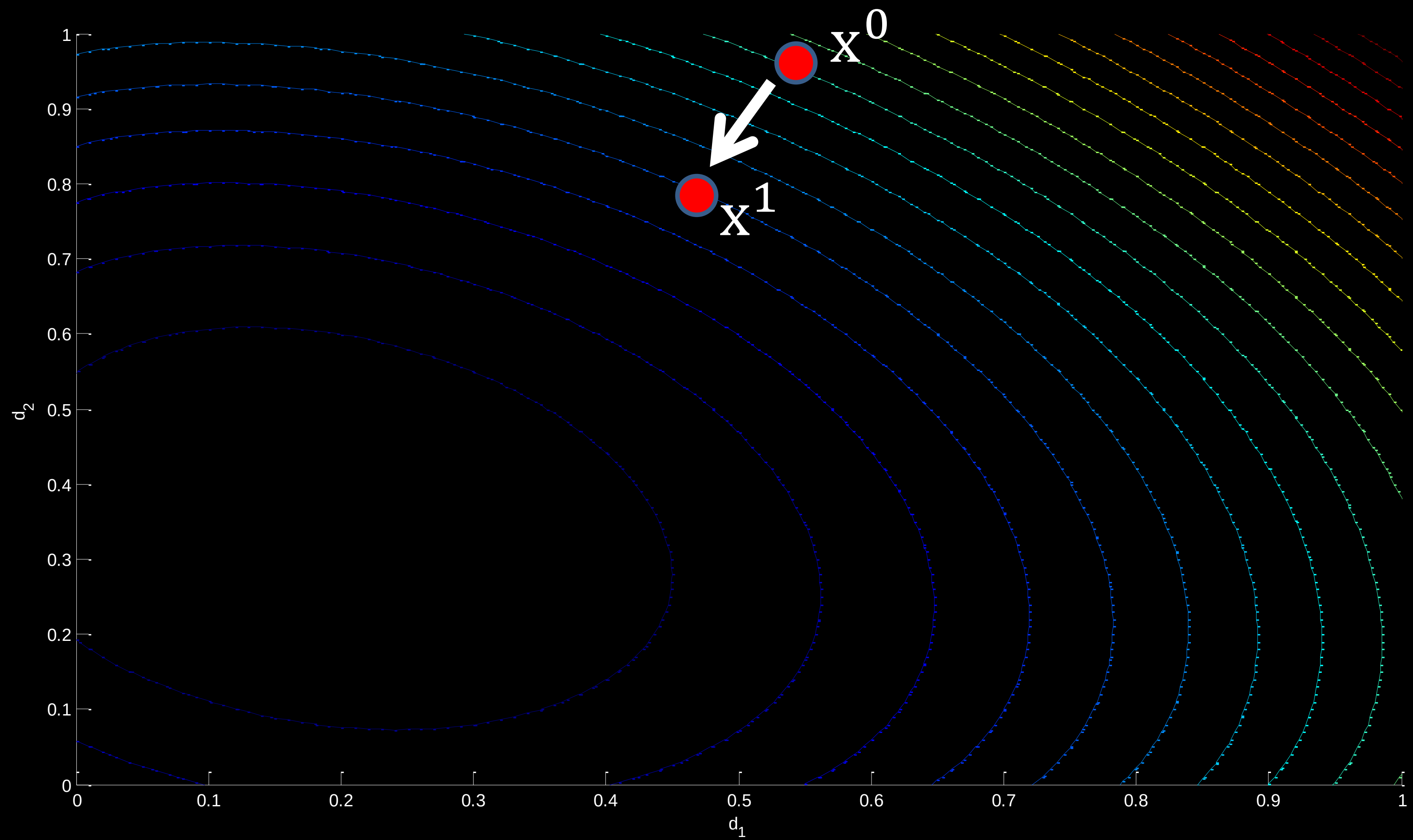
$$\text{s. t.} \quad \mathbf{1}^t\mathbf{Y}\alpha = 0$$

$$\mathbf{0} \leq \alpha \leq \mathbf{C} \ \& \ \mathbf{d} \in \boldsymbol{D}$$

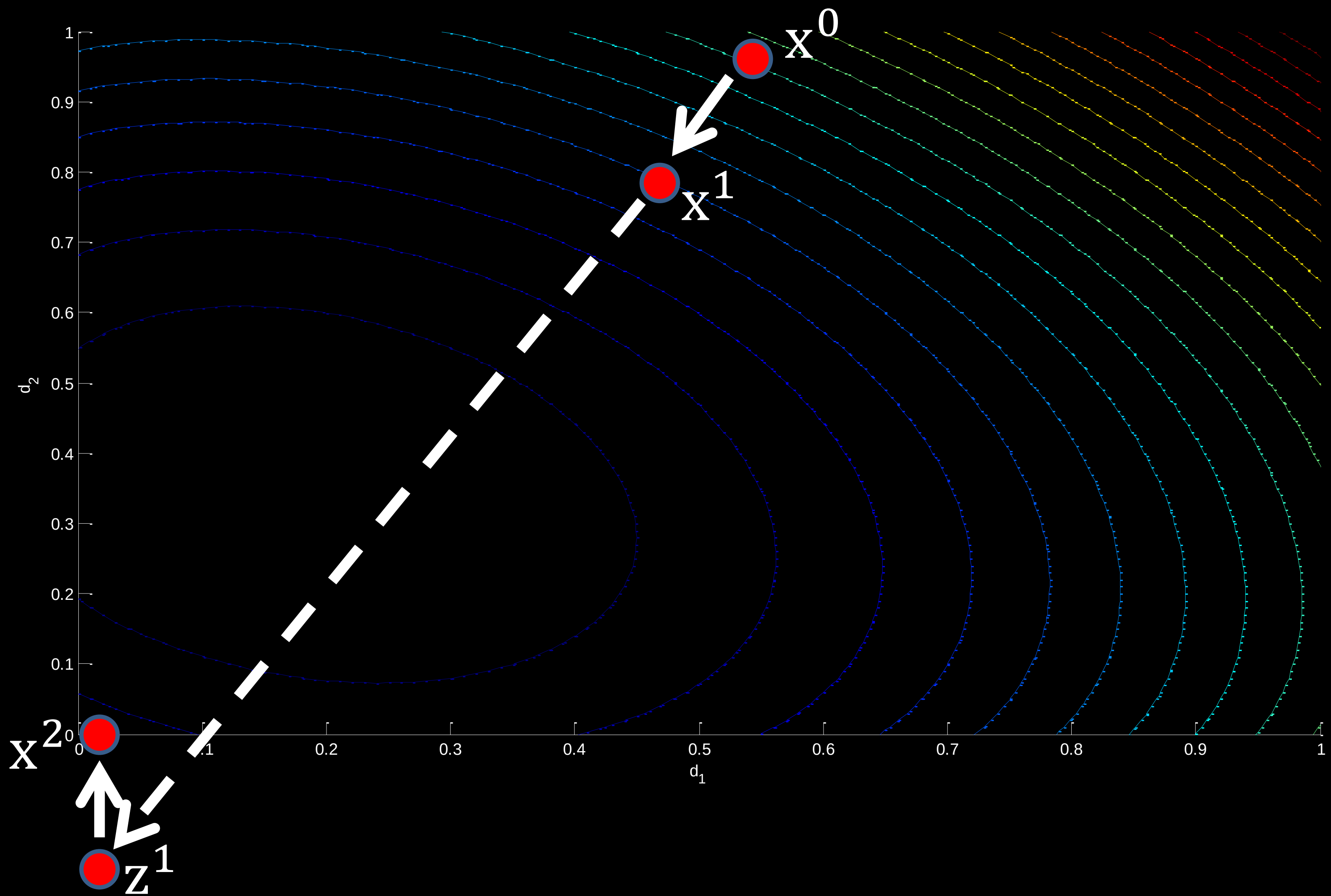
Projected Gradient Descent



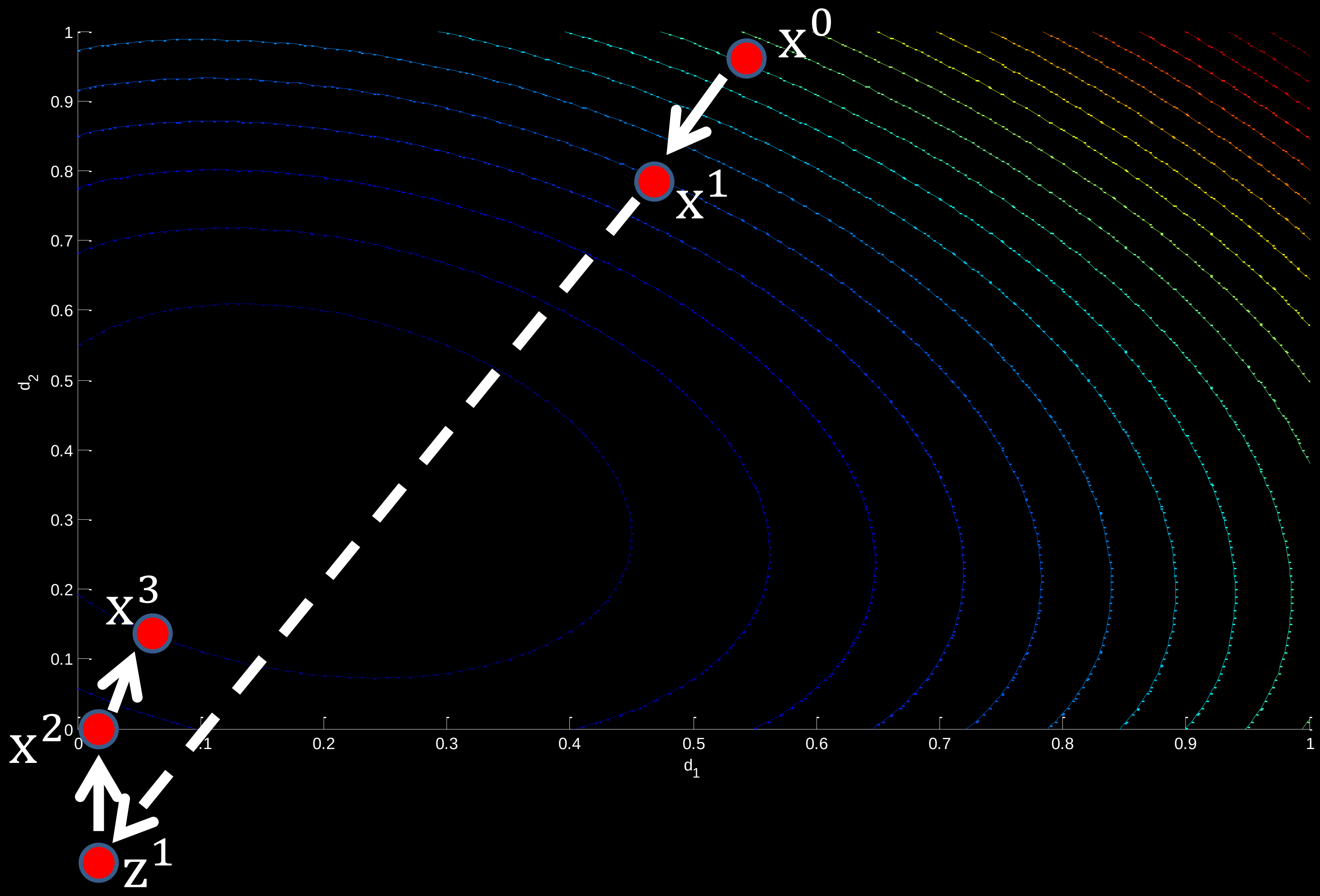
Projected Gradient Descent



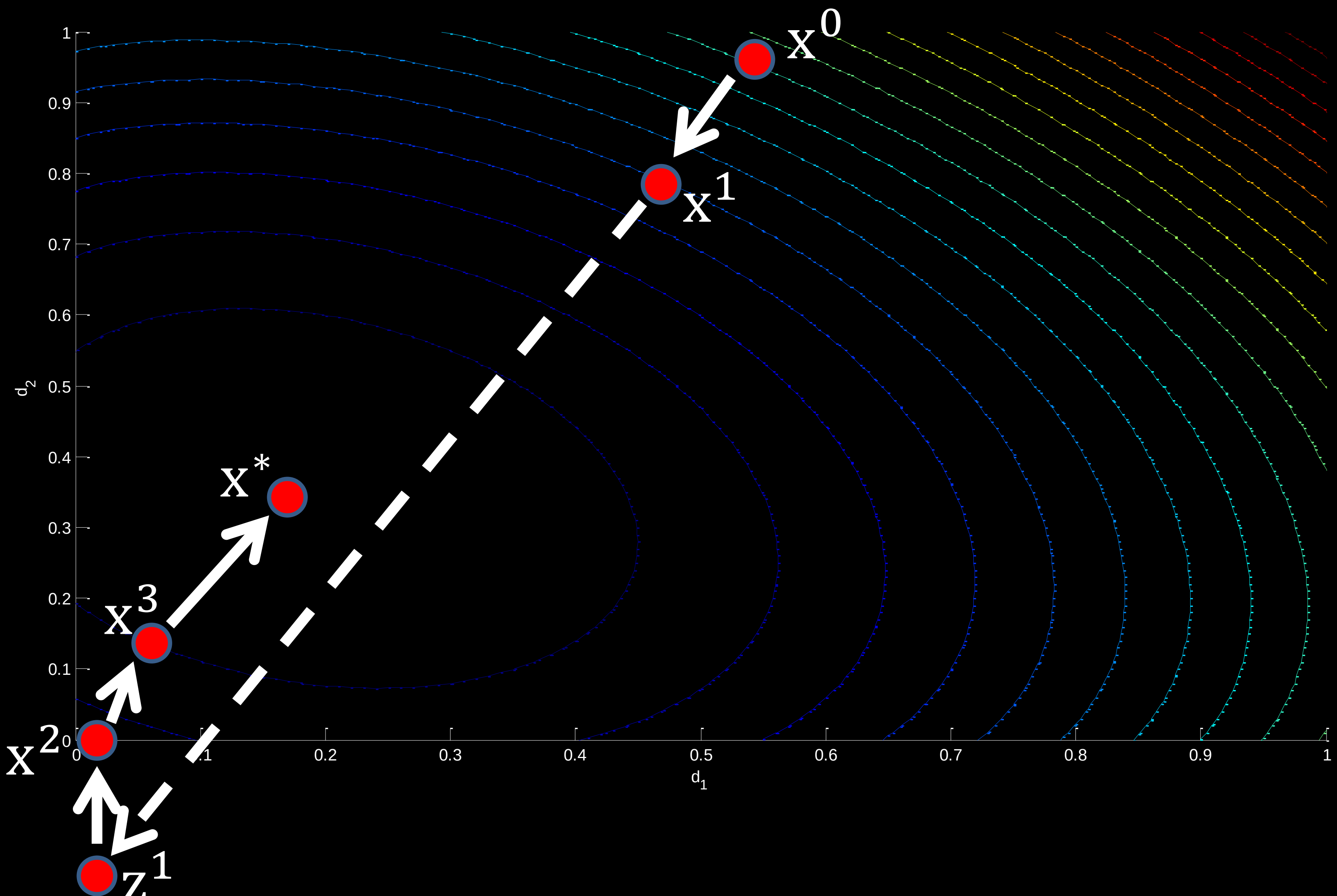
Projected Gradient Descent



Projected Gradient Descent



Projected Gradient Descent



PGD Limitations

- PGD requires many function and gradient evaluations as
  - No step size information is available.
  - The Armijo rule might reject many step size proposals.
  - Inaccurate gradient values can lead to many tiny steps.

PGD Limitations

- PGD requires many function and gradient evaluations as
  - No step size information is available.
  - The Armijo rule might reject many step size proposals.
  - Inaccurate gradient values can lead to many tiny steps.

- Noisy function and gradient values can cause PGD to converge to points far away from the optimum.

PGD Limitations

- PGD requires many function and gradient evaluations as
  - No step size information is available.
  - The Armijo rule might reject many step size proposals.
  - Inaccurate gradient values can lead to many tiny steps.

- Noisy function and gradient values can cause PGD to converge to points far away from the optimum.

- Solving SVMs to high precision to obtain accurate function and gradient values is very expensive.

PGD Limitations

- PGD requires many function and gradient evaluations as
  - No step size information is available.
  - The Armijo rule might reject many step size proposals.
  - Inaccurate gradient values can lead to many tiny steps.

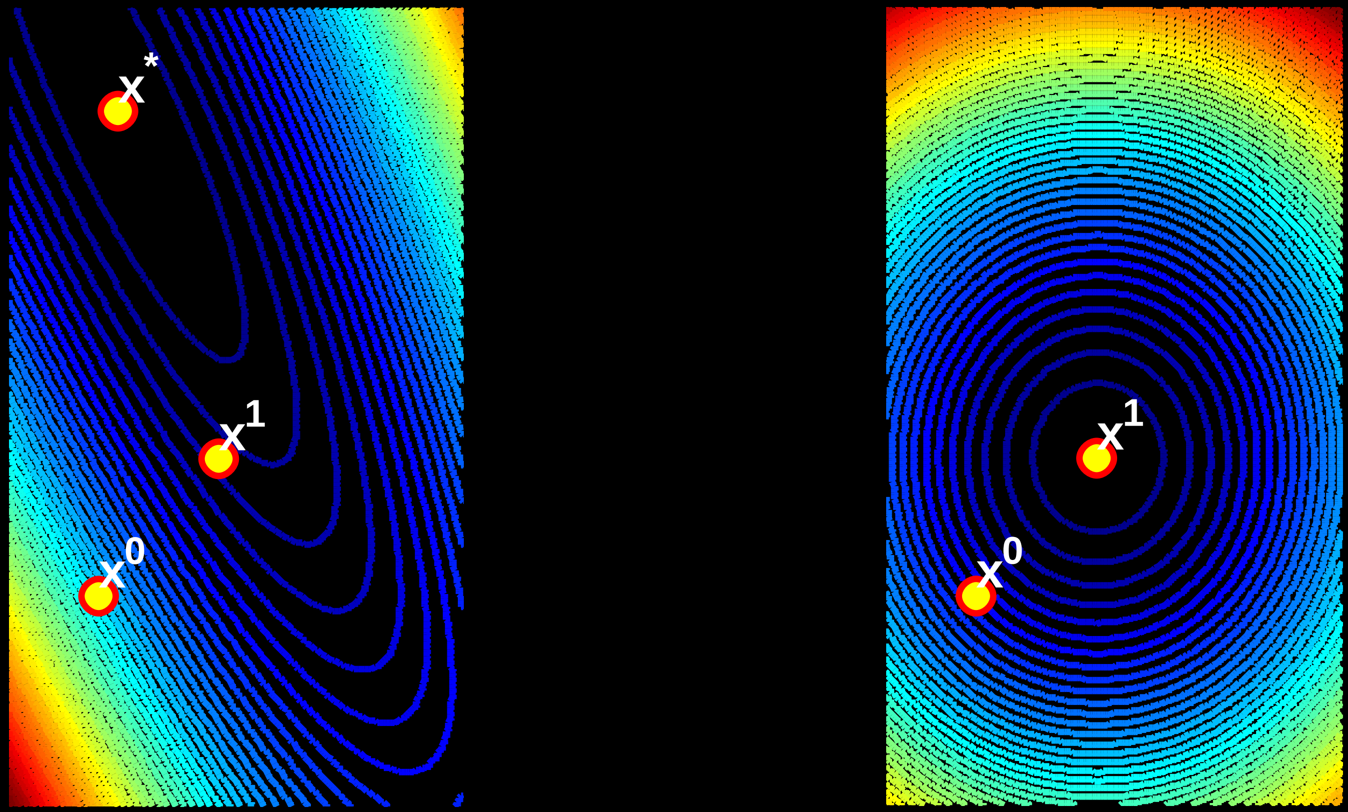
- Noisy function and gradient values can cause PGD to converge to points far away from the optimum.

- Solving SVMs to high precision to obtain accurate function and gradient values is very expensive.

- Repeated projection onto the feasible set might also be expensive.

SPG Solution – Spectral Step Length

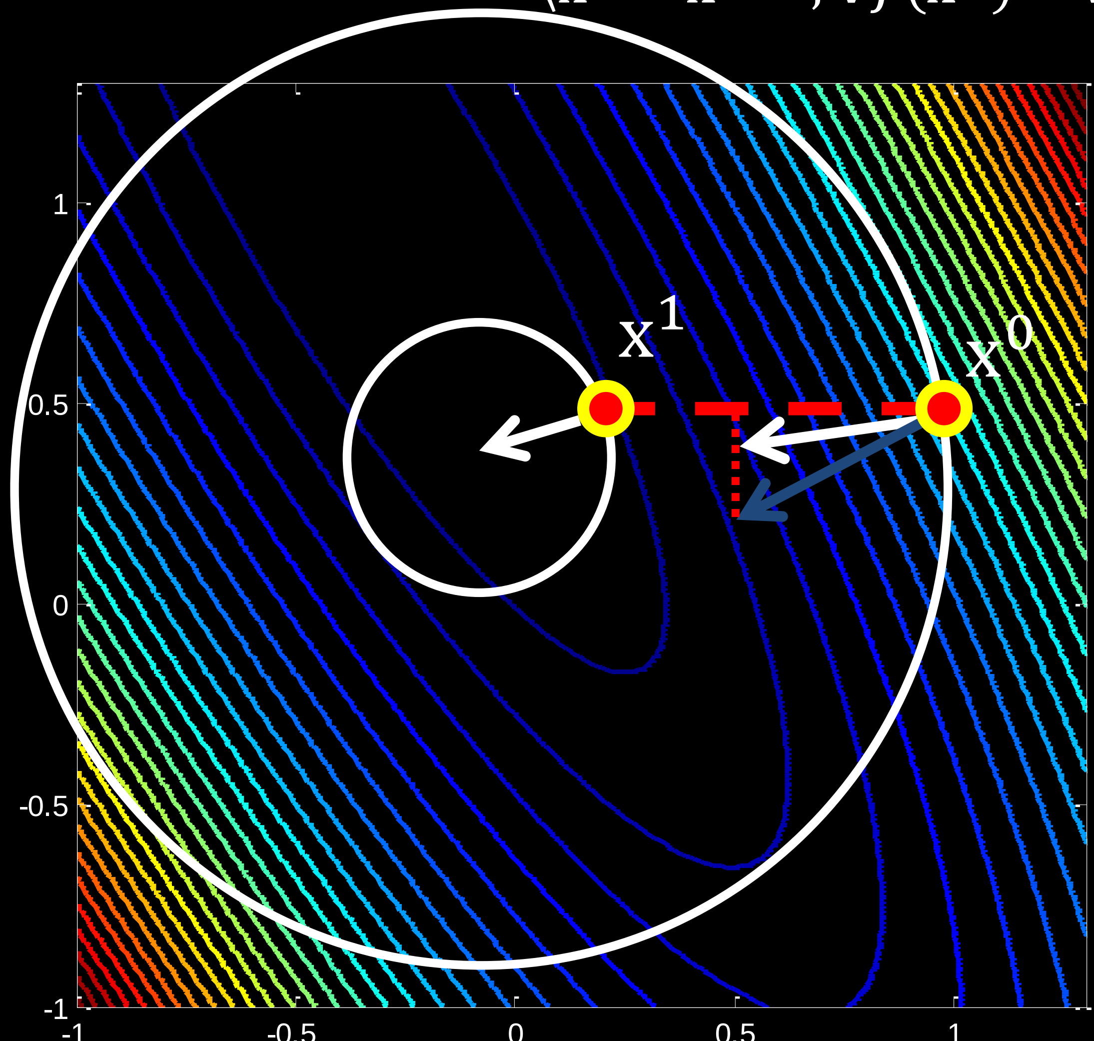
- Quadratic approximation : $\frac{1}{2}\lambda^{-1}\mathbf{x}^t\mathbf{x} + \mathbf{c}^t\mathbf{x} + d$
- Spectral step length : $\lambda_{SPG} = \dfrac{\langle \mathbf{x}^n - \mathbf{x}^{n-1}, \mathbf{x}^n - \mathbf{x}^{n-1} \rangle}{\langle \mathbf{x}^n - \mathbf{x}^{n-1}, \nabla f(\mathbf{x}^n) - \nabla f(\mathbf{x}^{n-1}) \rangle}$



Original Function　　　　Approximation

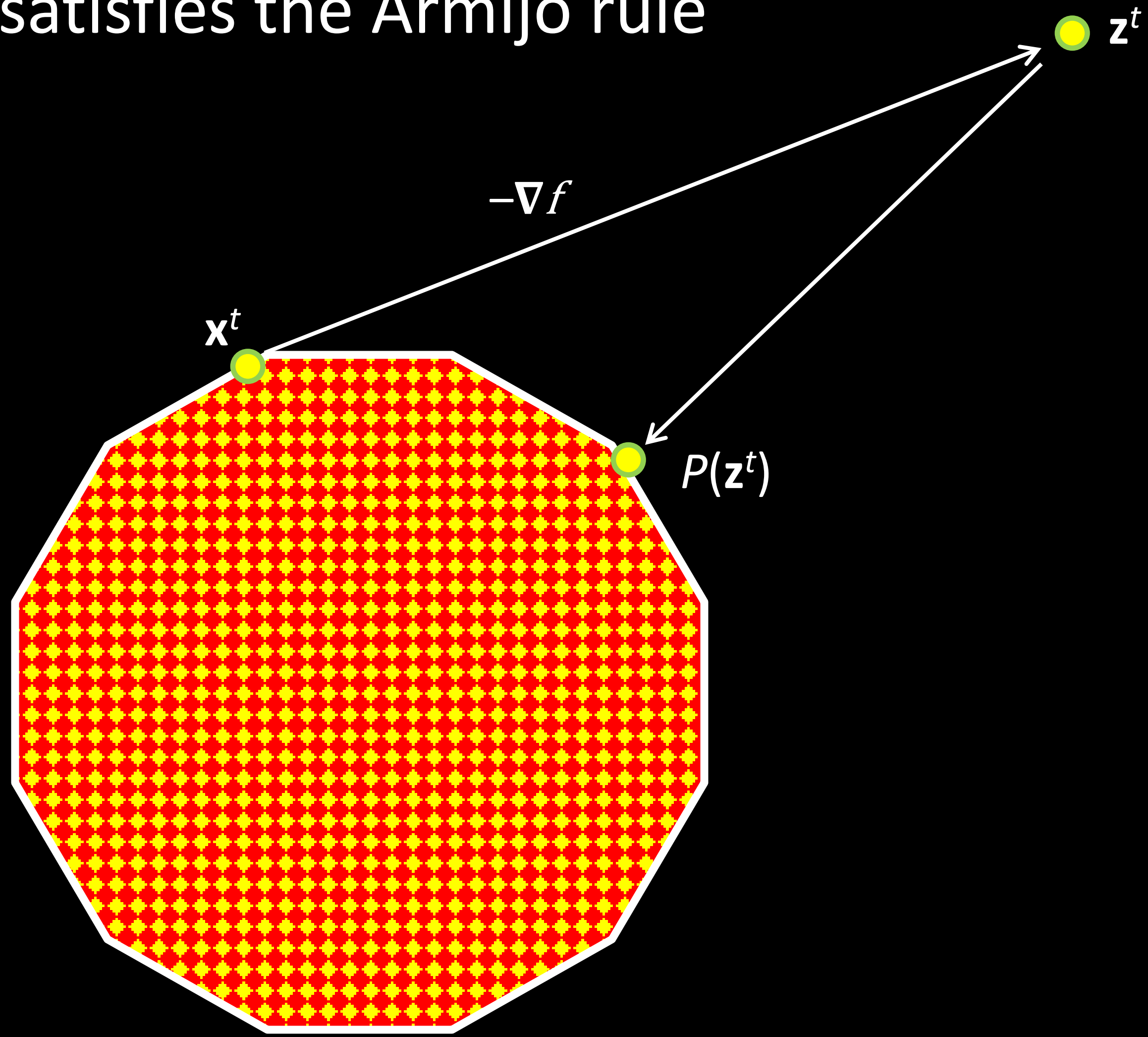
SPG Solution – Spectral Step Length

- Spectral step length : $\lambda_{SPG} = \dfrac{\langle \mathbf{x}^n - \mathbf{x}^{n-1}, \mathbf{x}^n - \mathbf{x}^{n-1} \rangle}{\langle \mathbf{x}^n - \mathbf{x}^{n-1}, \nabla f(\mathbf{x}^n) - \nabla f(\mathbf{x}^{n-1}) \rangle}$



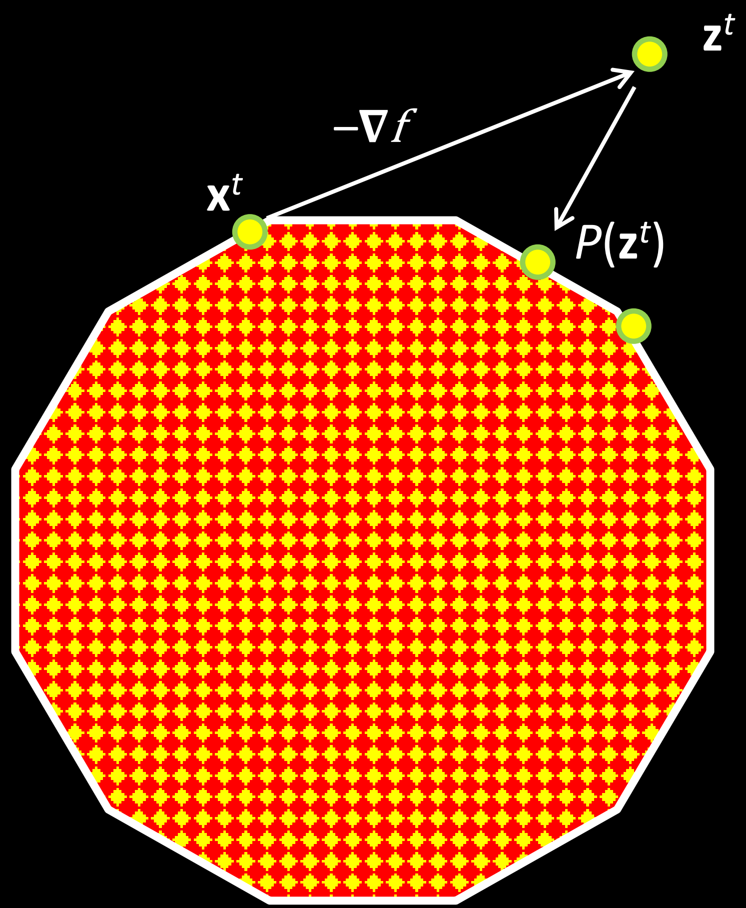
PGD Limitations – Repeated Projections

- Accept $P(\mathbf{z}^t)$ if it satisfies the Armijo rule



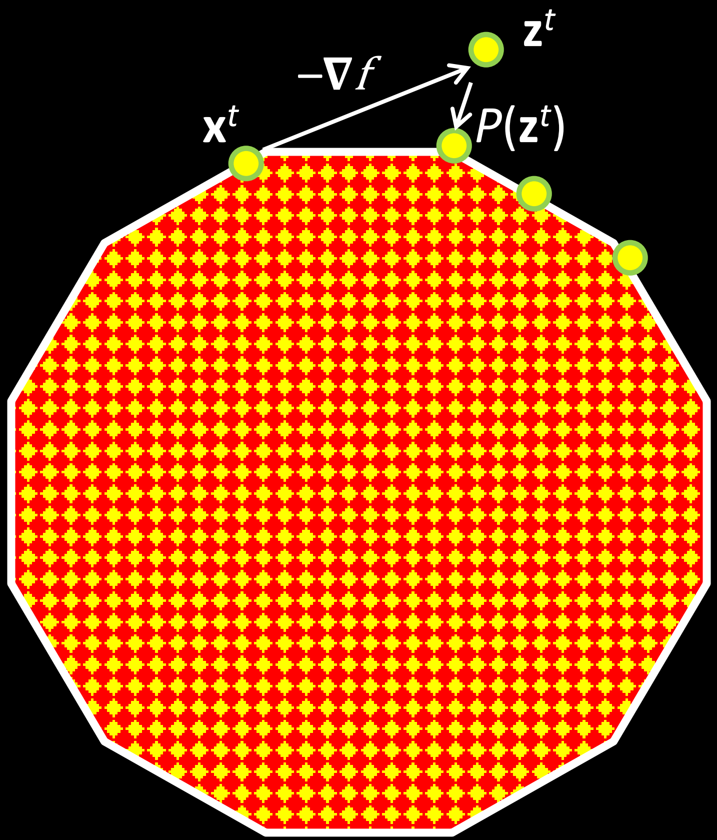
PGD Limitations – Repeated Projections

- Accept $P(\mathbf{z}^t)$ if it satisfies the Armijo rule



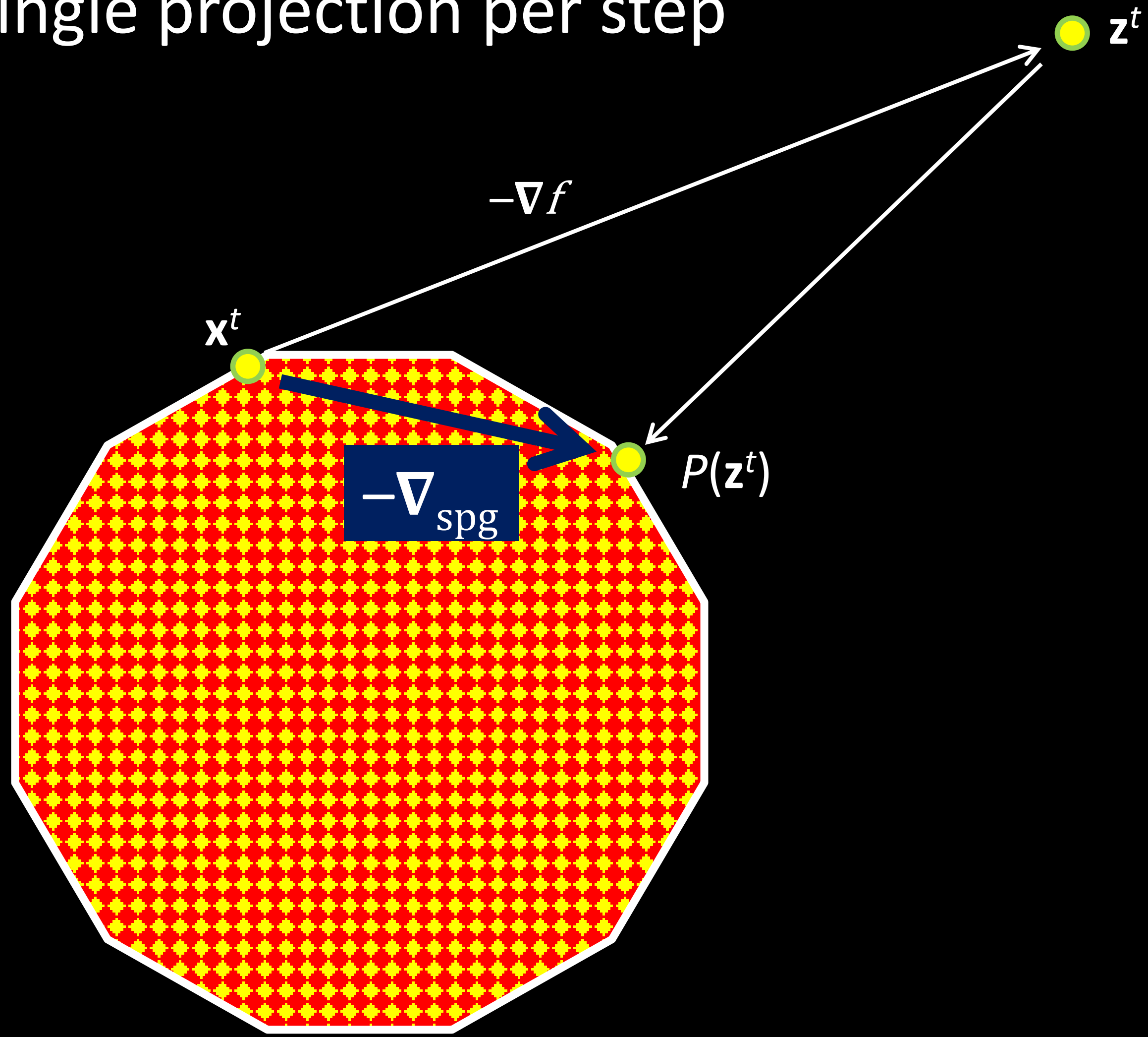
PGD Limitations – Repeated Projections

- PGD might require many projections before accepting a point



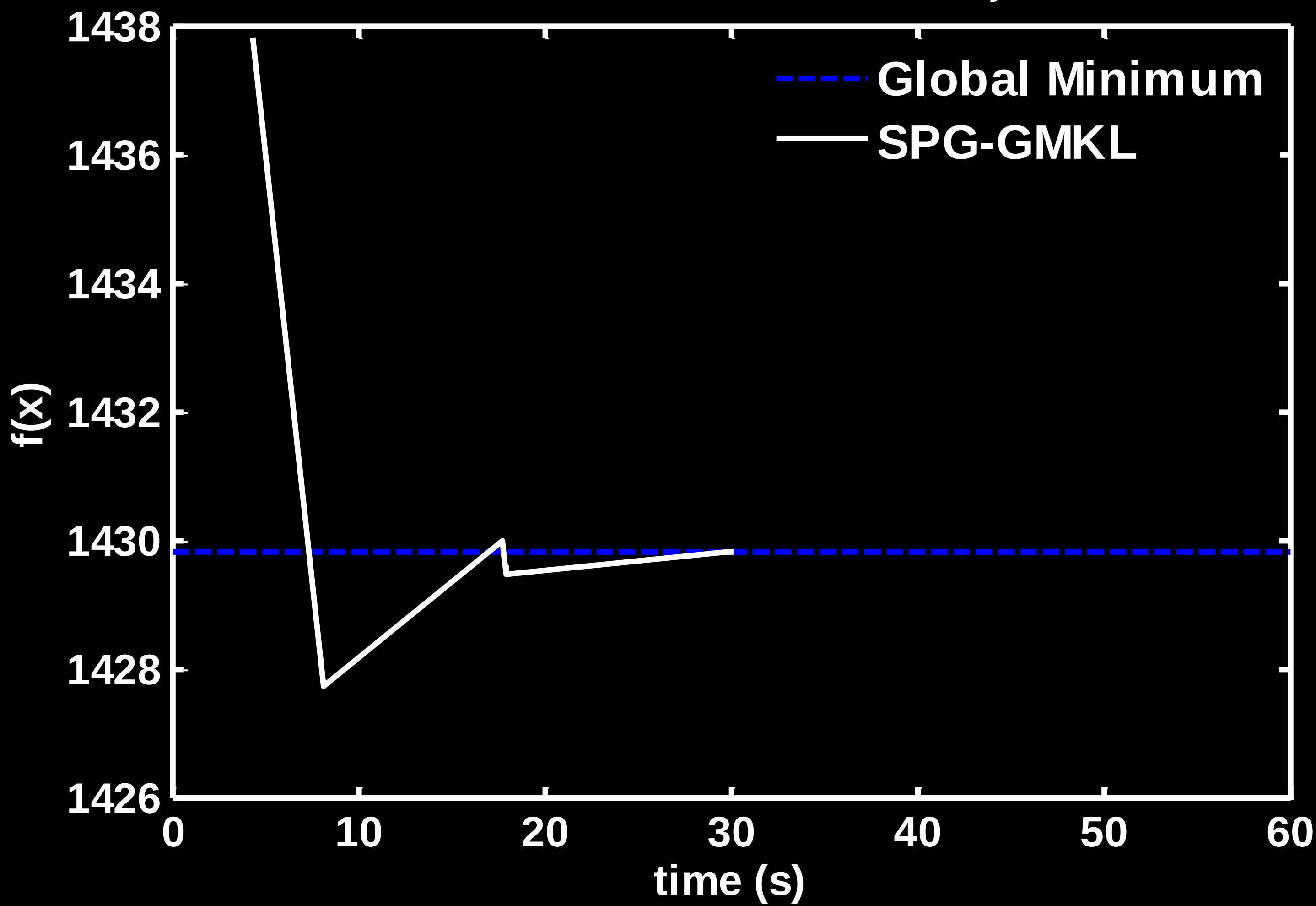
SPG Solution – Spectral Proj Gradient

- SPG requires a single projection per step



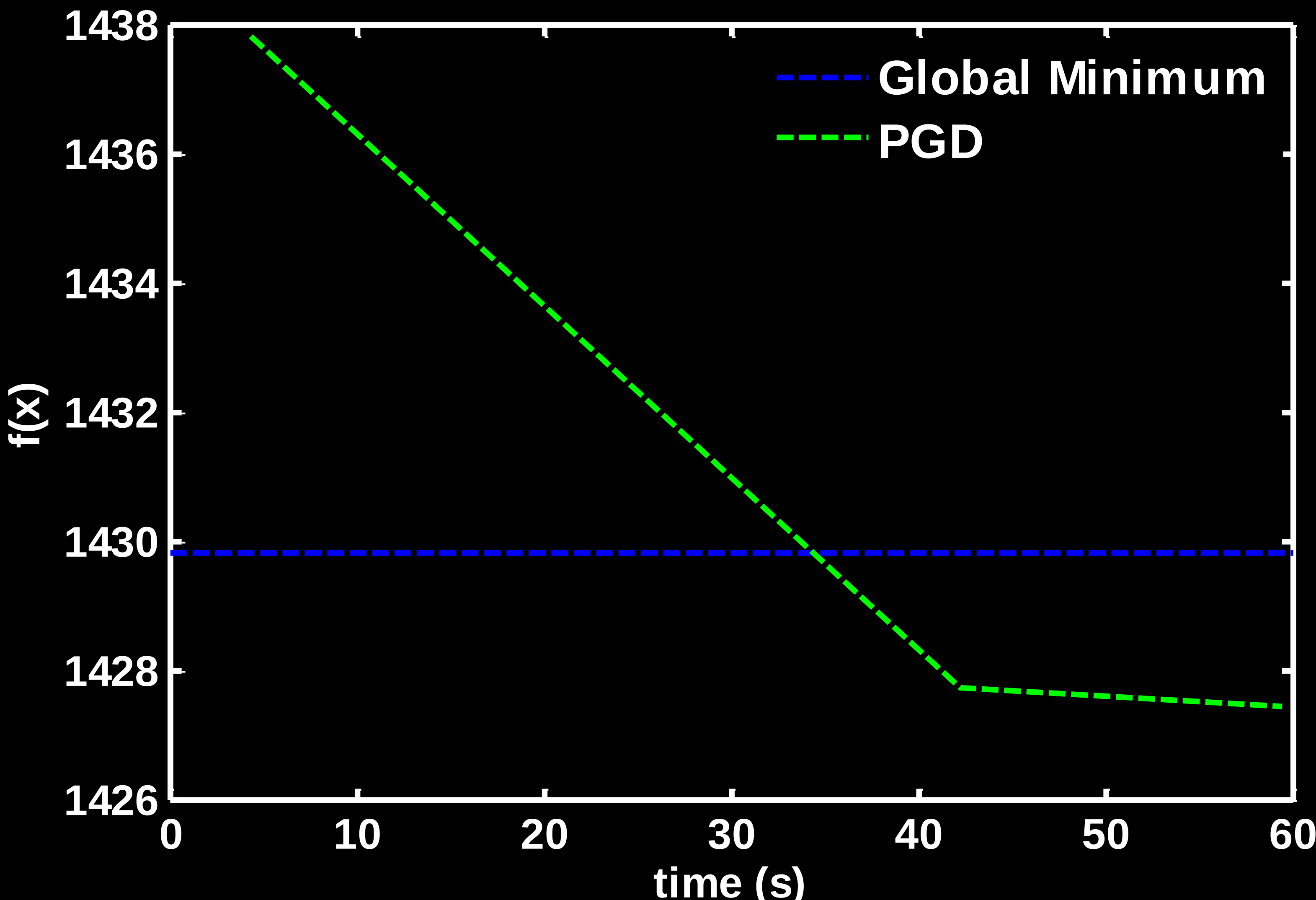
SPG Solution – Non-Monotone Rule

- Handling function and gradient noise.
- Non-monotone rule: $f(x^t - s\nabla f(x^t)) \leq \max_{0 \leq j \leq M} f\left(x^{t-j}\right) - \gamma s \|\nabla f(x^t)\|_2^2$

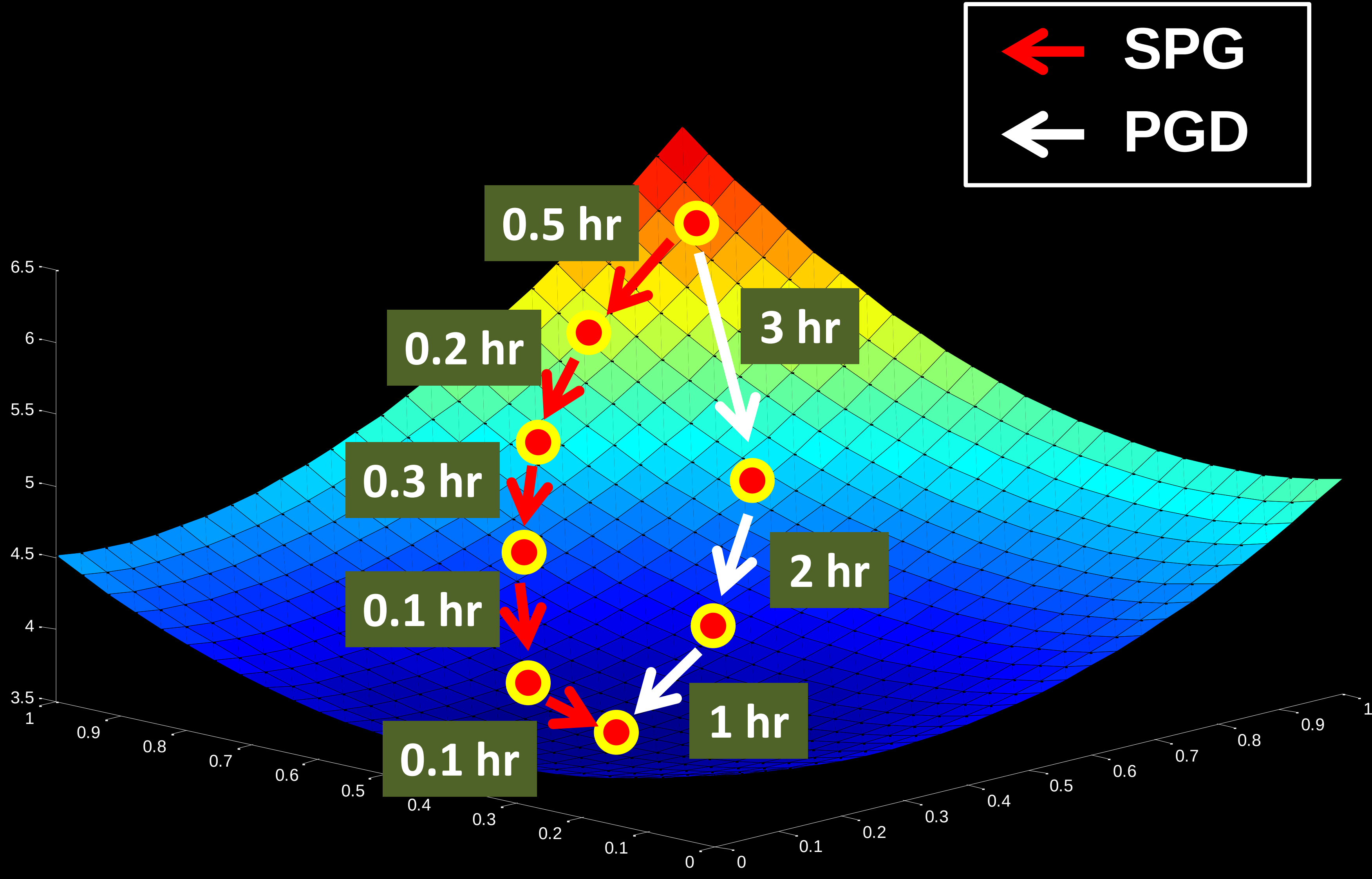


PGD Limitations – Step Size Selection

- The Armijo rule might get stuck due to noisy function values



SPG Solution – SVM Precision Tuning



SPG Advantages

- SPG requires fewer function and gradient evaluations due to
  - The 2nd order spectral step length estimation.
  - The non-monotone line search criterion.

- SPG is more robust to noisy function and gradient values due to the non-monotone line search criterion.

- SPG never needs to solve an SVM with high precision due to our precision tuning strategy.

- SPG needs to perform only a single projection per step.

SPG Algorithm

$n \leftarrow 0$

Initialize $\mathbf{d}^0$ randomly

repeat

$\quad \boldsymbol{\alpha}^* \leftarrow \mathrm{SolveSVM}(\mathbf{K}(\mathbf{d}^n), \epsilon)$

$\quad \lambda \leftarrow \mathrm{SpectralStepLength}$

$\quad \mathbf{p}^n \leftarrow \mathbf{d}^n - \mathbf{P}(\mathbf{d}^n - \boldsymbol{\lambda}\boldsymbol{\nabla} W(\mathbf{d}^n, \boldsymbol{\alpha}^*))$

$\quad \mathrm{s}^n \leftarrow \mathrm{Non} - \mathrm{Monotone}$

$\quad \epsilon \leftarrow \mathrm{TuneSVMPrecision}$

$\quad \mathbf{d}^{n+\mathbf{1}} \leftarrow \mathbf{d}^n - \mathrm{s}^n \mathbf{p}^n$

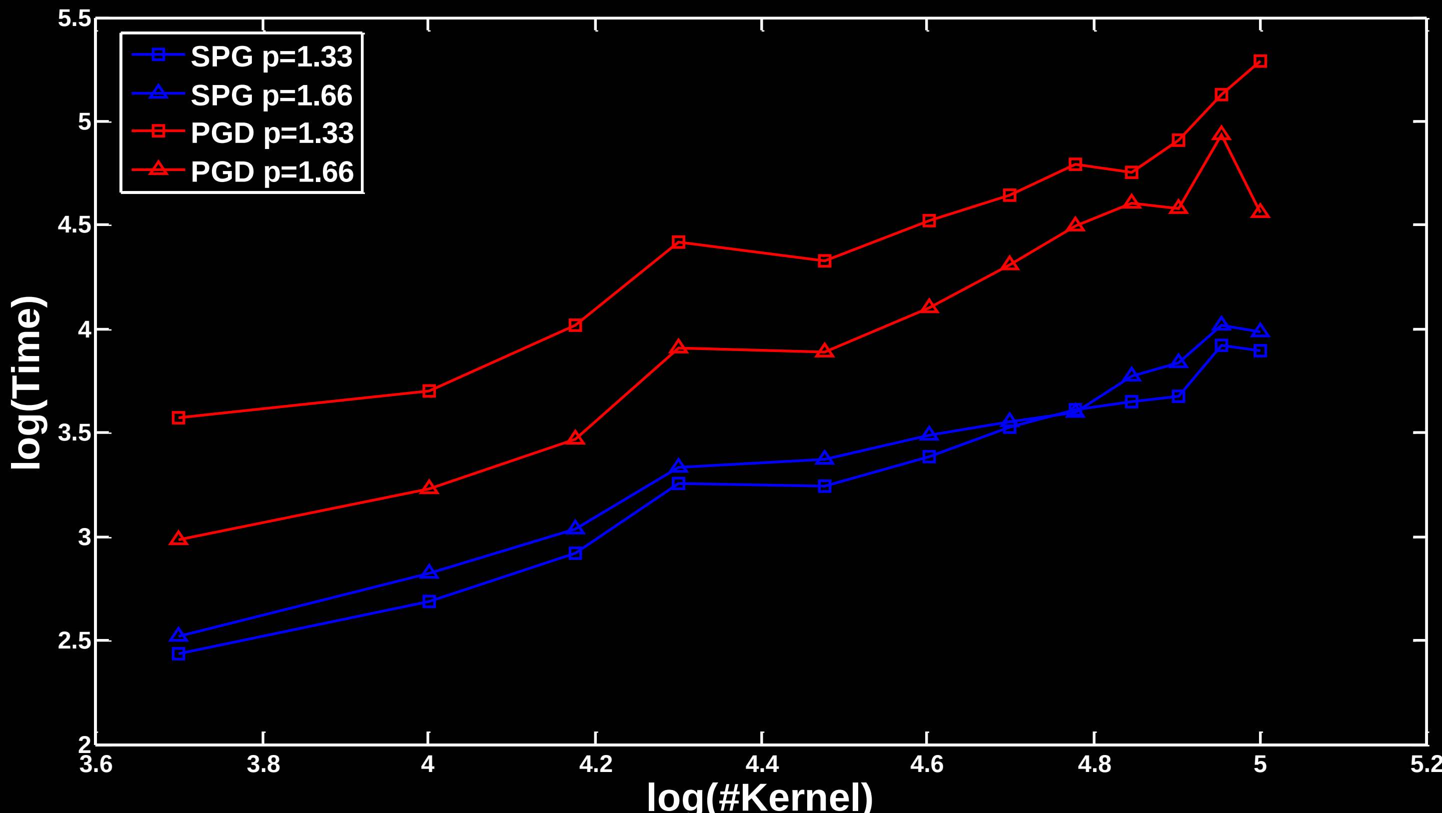
until converged

Results on Large Scale Data Sets

- Covertype: Sum of kernels subject to $l_{1.33}$ regularization
  - Number of training points 581,012
  - Number of Kernels 5
  - SPG time taken 64.46 hrs
- SPG took 26 SVM evaluations
- First SVM evaluation took 44 hours
- Only 0.19% of SV were cached

Results on Large Scale Data Sets

- Sonar: Sum of kernels subject to $l_{1.33}$ regularization
  - Number of training points 208
  - Number of Kernels 1 Million
  - SPG time taken 105.62 hrs



Results on Large Scale Data Sets

- Sum of kernels subject to $l_{p\geq1}$ regularization

Data Sets	# Train	# Kernels	p=1		p=1.33	
			PGD (hrs)	SPG (hrs)	PGD (hrs)	SPG (hrs)
Adult - 9	32,561	50	35.84	4.55	31.77	4.42
Cod - RNA	59,535	50	–	25.17	66.48	19.10
KDDCup04	50,000	50	–	40.10	–	42.20

Results on Small Scale Data Sets

- Sum of kernels subject to l_1 regularization

Data Sets	SimpleMKL (s)	Shogun (s)	PGD (s)	SPG (s)
Wpbc	400 ± 128.4	15 ± 7.7	38 ± 17.6	6 ± 4.2
Breast - Cancer	676 ± 356.4	12 ± 1.2	57 ± 85.1	5 ± 0.6
Australian	383 ± 33.5	1094 ± 621.6	29 ± 7.1	10 ± 0.8
Ionosphere	1247 ± 680.0	107 ± 18.8	1392 ± 824.2	39 ± 6.8
Sonar	1468 ± 1252.7	935 ± 65.0	–	273 ± 64.0

Results on Large Scale Data Sets

- Product of kernels subject to $l_{p\geq 1}$ regularization

Data Sets	# Train	# Kernels	p=1		p=1.33	
			PGD (hrs)	SPG (hrs)	PGD (hrs)	SPG (hrs)
Letter	20,000	16	18.66	0.67	18.69	0.66
Poker	25,010	10	5.57	0.49	2.29	0.96
Adult - 8	22,696	42	–	1.73	–	3.42
Web - 7	24,692	43	–	0.88	–	1.33
RCV1	20,242	50	–	18.17	–	15.93
Cod - RNA	59,535	8	–	3.45	–	8.99

Effect of Individual Components

- Sum of kernels subject to $l_{1.1}$ regularization

Data Sets	PGD		PGD + N		PGD + S		PGD + N + S	
	Time (s)	# SVMs	Time (s)	# SVMs	Time (s)	# SVMs	Time (s)	# SVMs
Australian	39.4 ± 6.0	3230	32.7 ± 3.6	116	317.0 ± 49.1	5980	7.0 ± 1.6	621
Sonar	785.5 ± 471.1	209461	41.6 ± 17.1	3236	40.2 ± 24.6	3806	9.0 ± 1.8	2427
Breast - Cancer	237.3 ± 97.8	109599	42.2 ± 4.1	1187	14.9 ± 2.2	3537	8.6 ± 2.2	3006
Diabetes	73.6 ± 38.8	29347	26.3 ± 9.5	2966	10.5 ± 2.6	1239	4.1 ± 0.5	695
Wpbc	44.4 ± 11.6	14376	27.9 ± 13.6	9388	2.9 ± 0.8	340	1.2 ± 0.4	79

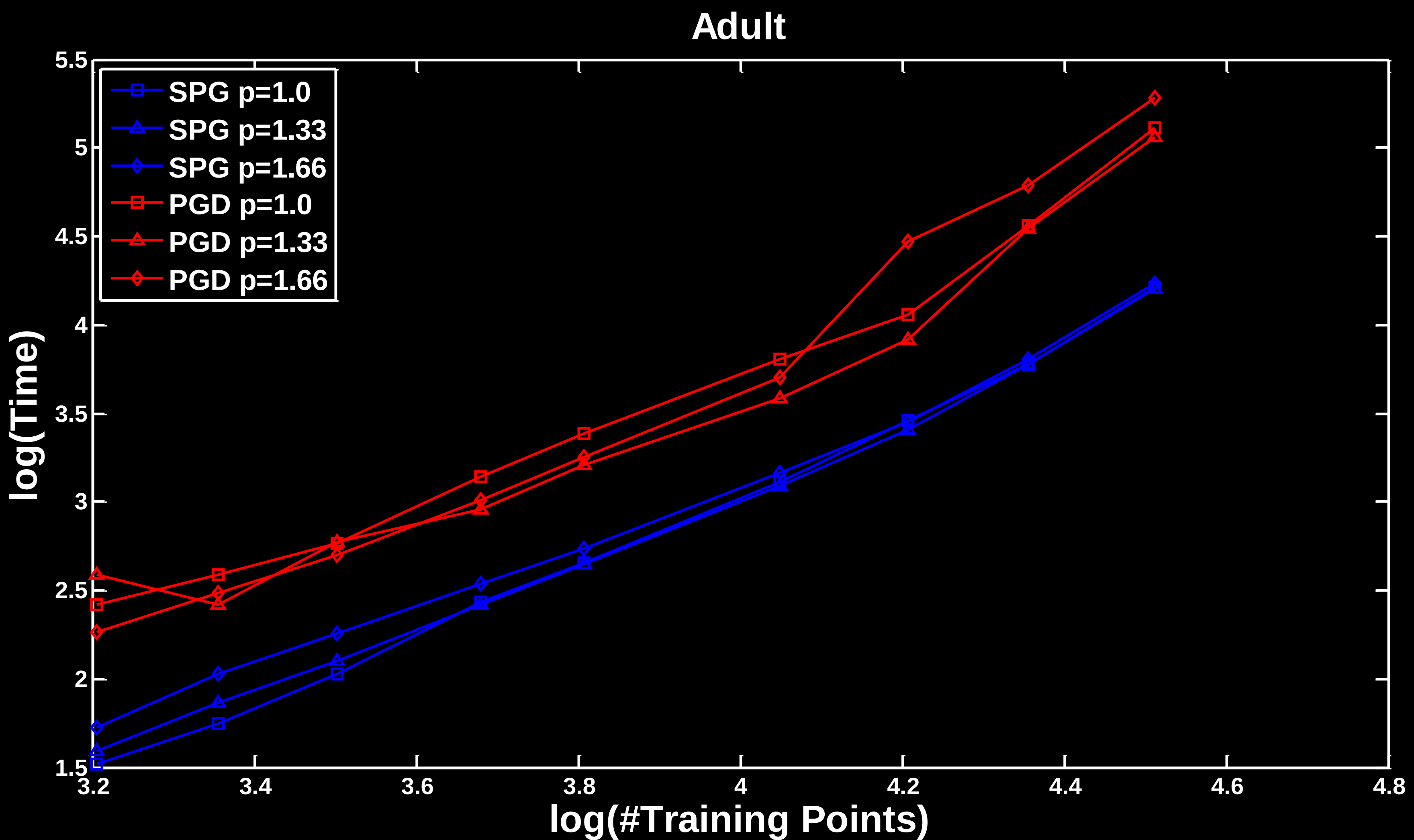
SVM Precision Tuning

- Sum of kernels subject to $l_{1.33}$ regularization

Data Sets	# Train	# Kernels	PGD (hrs)	PGD + N + S (hrs)	SPG (hrs)
Adult - 9	32,561	50	31.77	8.33	4.43
Web - 8	49,749	50	4.27	1.73	0.87
Sonar	208	100,000	53.91	3.35	2.19

SPG Scaling Properties

- Scaling with the number of training points



Conclusions

- Developed a generic and efficient MKL optimizer.
- Experimented with four different MKL formulations and solved both small and large scale problems.
- Combining spectral step length and non-monotone rule gives best performance.
- Quasi Newton methods not suitable for MKL problems due to noisy gradient.

Code: http://research.microsoft.com/en-us/um/people/manik/code/SPG-GMKL/download.html

Acknowledgements

- Kamal Gupta (IITD)
- Subhashis Banerjee (IITD)
- The Computer Services Center at IIT Delhi