Stochastic Dual Coordinate Ascent for training Conditional Random Fields

Rémi Le Priol

Montreal Institute of Learning Algorithms

remi.lp.17@gmail.com

September 15, 2017

Overview

1. Conditional Random Fields
2. SDCA for Max-likelihood
3. Non-Uniform Sampling
4. Leverage the Structure

Overview

1. Conditional Random Fields
2. SDCA for Max-likelihood
3. Non-Uniform Sampling
4. Leverage the Structure

Structured Prediction

$$
\begin{aligned}
\text{data point } x \in \mathcal{X} &\mapsto \text{ structured label } y \in \mathcal{Y} \\
\text{letter drawings} &\mapsto \text{word} \\
\text{sentence in English} &\mapsto \text{sentence in French} \\
\text{sentence} &\mapsto \text{parsing tree} \\
\text{natural image} &\mapsto \text{semantic segmentation}
\end{aligned}
$$

Structured Prediction

$$\begin{aligned} \text{data point } x \in \mathcal{X} &\mapsto \text{ structured label } y \in \mathcal{Y} \\ \text{letter drawings} &\mapsto \text{word} \\ \text{sentence in English} &\mapsto \text{sentence in French} \\ \text{sentence} &\mapsto \text{parsing tree} \\ \text{natural image} &\mapsto \text{semantic segmentation} \end{aligned}$$

Hypothesis

The conditional distribution $p(y|x)$ is Markov with respect to an undirected graphical model $G = (V, E)$.

Features

Feature extractor:

$$F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$$

Hypothesis

$$F(x, y) = \sum_{c \in \mathcal{C}} F_c(x, y_c)$$

where $\mathcal{C}$ is the set of maximal cliques of G.

The Model

Conditional probability of y given x:

$$p(y|x;w) := \frac{\exp(w^T F(x,y))}{\sum_{y' \in \mathcal{Y}} \exp(w^T F(x,y'))}$$

Standard approach to train CRF: Maximum Likelihood

$$\min_w \mathscr{P}(w) = \frac{\lambda}{2}\|w\|^2 - \frac{1}{n}\sum_{i=1}^n \log(p(y_i|x_i;w))$$

Reformulation

New notations:

Corrected features: $\psi_i(y) := F(x_i, y_i) - F(x_i, y) \in \mathbb{R}^d$

Corrected feature matrix: $A_i := (\psi_i(1), \psi_i(2), ..., \psi_i(|\mathcal{Y}_i|)) \in \mathbb{R}^{d \times |\mathcal{Y}_i|}$

Log-sum-exp function: $\phi_i(z) = \log\left(\sum_{y \in \mathcal{Y}_i} \exp(z_y)\right)$

Reformulation

New notations:

Corrected features: $\psi_i(y) := F(x_i, y_i) - F(x_i, y) \in \mathbb{R}^d$

Corrected feature matrix: $A_i := (\psi_i(1), \psi_i(2), ..., \psi_i(|\mathcal{Y}_i|)) \in \mathbb{R}^{d \times |\mathcal{Y}_i|}$

Log-sum-exp function: $\phi_i(z) = \log\Big(\sum_{y \in \mathcal{Y}_i} \exp(z_y)\Big)$

The log-likelihood becomes :

$$
\begin{aligned}
-\log(p(y_i|x_i; w)) &= \log\Big(\sum_y e^{w^T F(x_i, y)}\Big) - w^T F(x_i, y_i) \\
&= \log\Big(\sum_y e^{-w^T \psi_i(y)}\Big) \\
&= \phi_i(-A_i^T w)
\end{aligned}
$$

Primal Objective

$$\min_{w \in \mathbb{R}^d} \mathscr{P}(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \phi_i(-A_i^T w) \tag{1}$$

Primal Objective

$$\min_{w \in \mathbb{R}^d} \mathscr{P}(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \phi_i(-A_i^T w) \tag{1}$$

HARD ! A_i is huge.

Anterior Work

- Exponentiated Gradient by Collins in 2008[1]
- Non-Uniform Sampling – Stochastic Average Gradient (NUS-SAG) by Schmidt in 2014[2]

[1] Collins et al., "Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks".

[2] Schmidt et al., "Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields".

[3] Shalev-Shwartz and Zhang, "Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization".

Anterior Work

- Exponentiated Gradient by Collins in 2008[1]
- Non-Uniform Sampling – Stochastic Average Gradient (NUS-SAG) by Schmidt in 2014[2]
- SDCA[3]

[1] Collins et al., "Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks".

[2] Schmidt et al., "Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields".

[3] Shalev-Shwartz and Zhang, "Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization".

Anterior Work

- Exponentiated Gradient by Collins in 2008[1]
- Non-Uniform Sampling – Stochastic Average Gradient (NUS-SAG) by Schmidt in 2014[2]
- SDCA[3]

Variance reduced methods.

[1] Collins et al., "Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks".

[2] Schmidt et al., "Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields".

[3] Shalev-Shwartz and Zhang, "Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization".

Anterior Work

- Exponentiated Gradient by Collins in 2008[1]
- Non-Uniform Sampling – Stochastic Average Gradient (NUS-SAG) by Schmidt in 2014[2]
- SDCA[3]

Variance reduced methods.

Why SDCA?

[1] Collins et al., "Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks".

[2] Schmidt et al., "Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields".

[3] Shalev-Shwartz and Zhang, "Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization".

Anterior Work

- Exponentiated Gradient by Collins in 2008[1]
- Non-Uniform Sampling – Stochastic Average Gradient (NUS-SAG) by Schmidt in 2014[2]
- SDCA[3]

Variance reduced methods.

Why SDCA?

Exact line search for cheap!

[1] Collins et al., "Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks".

[2] Schmidt et al., "Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields".

[3] Shalev-Shwartz and Zhang, "Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization".

Overview

1. Conditional Random Fields
2. SDCA for Max-likelihood
3. Non-Uniform Sampling
4. Leverage the Structure

Dual Formulation

Primal:

$$\min_{w \in \mathbb{R}^d} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \phi_i(-A_i^T w)$$

Dual:

$$\max_{\alpha | \forall i, \alpha_i \in \Delta_i} \mathscr{D}(\alpha) = -\frac{1}{2\lambda}\|\frac{1}{n}\sum_i A_i \alpha_i\|^2 + \frac{1}{n}\sum_{i=1}^{n} H_i(\alpha_i)$$

Δ_i is the simplex of dimension $|\mathcal{Y}_i|$. H_i is the entropy over Δ_i.

Conjugate variables

Primal probabilities

$$\forall i, \boldsymbol{\alpha}_i(w) := \nabla \phi_i(-A_i^T w) = p(.|x_i; w) \propto \exp(-w^T \psi_i(.))$$

Dual weights

$$\begin{aligned} \boldsymbol{w}(\alpha) &= \frac{1}{\lambda n} \sum_i A_i \alpha_i = \frac{1}{\lambda n} \sum_i \mathbf{E}_{\alpha_i}[\psi_i] \\ &= \frac{1}{\lambda n} \sum_i F(x_i, y_i) - \frac{1}{\lambda n} \sum_i \mathbf{E}_{y \sim \alpha_i}[F(x_i, y)] \end{aligned}$$

Conjugate variables

Primal probabilities

$$\forall i, \boldsymbol{\alpha}_i(w) := \nabla \phi_i(-A_i^T w) = p(.|x_i; w) \propto \exp(-w^T \psi_i(.))$$

Dual weights

$$\begin{aligned}\boldsymbol{w}(\alpha) &= \frac{1}{\lambda n} \sum_i A_i \alpha_i = \frac{1}{\lambda n} \sum_i \mathbf{E}_{\alpha_i}[\psi_i] \\ &= \frac{1}{\lambda n} \sum_i F(x_i, y_i) - \frac{1}{\lambda n} \sum_i \mathbf{E}_{y \sim \alpha_i}[F(x_i, y)]\end{aligned}$$

Optimality condition

$$\alpha^* = \boldsymbol{\alpha}(w^*) \quad \text{and} \quad w^* = \boldsymbol{w}(\alpha^*)$$

Max-entropy

$$\max_{\alpha|\forall i, \alpha_i \in \Delta_i} \mathscr{D}(\alpha) = \underbrace{-\frac{\lambda}{2}\|\boldsymbol{w}(\alpha)\|^2}_{\text{data fitting}} + \underbrace{\frac{1}{n}\sum_{i=1}^{n} H_i(\alpha_i)}_{\text{regularization}}$$

Max-entropy

$$\max_{\alpha | \forall i, \alpha_i \in \Delta_i} \mathscr{D}(\alpha) = \underbrace{-\frac{\lambda}{2} \|\boldsymbol{w}(\alpha)\|^2}_{\text{data fitting}} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} H_i(\alpha_i)}_{\text{regularization}}$$

HARD ! α is huge.

Principle of SDCA[4]

- Store dual probabilities α and $\boldsymbol{w}(\alpha)$.
- Sample $i \in \{1, \ldots, n\}$
- Update $\alpha_i^+ \leftarrow (1-\gamma)\alpha_i + \gamma \boldsymbol{\alpha}_i(\boldsymbol{w}(\alpha))$ with $\gamma \in [0, 1]$

[4] Shalev-Shwartz and Zhang, "Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization".

Pseudo code

Algorithm 1 SDCA for Logistic Regression

$\forall i$, initialize $\alpha_i^{(0)}$ at random in Δ_K
Let $w^{(0)} = \frac{1}{\lambda n} A\alpha$
Let $\forall i,\ g_i = 1$ (optional)
for $k = 0 \dots K$ **do**
  Pick i at random in $\{1, \dots, n\}$ (optionally, proportional to g_i)
  Let $\beta_i := p(.|x; w^{(k)}) = \boldsymbol{\alpha}_i(\boldsymbol{w}(\alpha_i^{(k)}))$
  Let $g_i = D_{KL}(\alpha_i || \beta_i)$ (optional)
  Let $d_i = \beta_i - \alpha_i^{(k)}$ (dual ascent direction)
  Let $v_i = \frac{1}{\lambda n} A_i d_i$ (primal descent direction)
  Solve $\gamma^* = \arg\max_{\gamma \in [0,1]} H_i(\alpha_i^{(k)} + \gamma d_i) - \frac{\lambda n}{2} \|w^{(k)} + \gamma v_i\|^2$ (Line Search)
  Update $\alpha_i^{(k+1)} := \alpha_i^{(k)} + \gamma^* d_i$
  Update $w^{(k+1)} := w^{(k)} + \gamma^* v_i$

Overview

1. Conditional Random Fields
2. SDCA for Max-likelihood
3. Non-Uniform Sampling
4. Leverage the Structure

On duality gaps

Duality gap:

$$g(w, \alpha) = \mathscr{P}(w) - \mathscr{D}(\alpha)$$

Primal model gap:

$$g(w, \boldsymbol{\alpha}(w)) = \frac{\lambda}{2}\|w - \boldsymbol{w}(\boldsymbol{\alpha}(w))\|^2 = \frac{1}{2\lambda}\|\nabla \mathscr{P}(w)\|^2$$

Dual model gap:

$$g(\boldsymbol{w}(\alpha), \alpha) = \frac{1}{n}\sum_i D_{KL}(\alpha_i || \boldsymbol{\alpha}_i(\boldsymbol{w}(\alpha)))$$

Idea

Our Scheme

- At each step, update individual duality gap $g_i = D_{KL}(\alpha_i || \boldsymbol{\alpha}_i(\boldsymbol{w}(\alpha))$
- Sample i proportionally to g_i.

[a] Osokin et al., "Minding the Gaps for Block Frank-Wolfe Optimization of Structured SVMs".

Idea

Our Scheme

- At each step, update individual duality gap $g_i = D_{KL}(\alpha_i || \boldsymbol{\alpha}_i(\boldsymbol{w}(\alpha))$
- Sample i proportionally to g_i.

Scheme adapted from Block-Coordinate Franck-Wolfe[a].
Transposable to the exponentiated gradient.

[a] Osokin et al., "Minding the Gaps for Block Frank-Wolfe Optimization of Structured SVMs".

Idea

Our Scheme

- At each step, update individual duality gap $g_i = D_{KL}(\alpha_i || \boldsymbol{\alpha}_i(\boldsymbol{w}(\alpha))$
- Sample i proportionally to g_i.

Scheme adapted from Block-Coordinate Franck-Wolfe[a].
Transposable to the exponentiated gradient.

[a] Osokin et al., "Minding the Gaps for Block Frank-Wolfe Optimization of Structured SVMs".

Competing scheme

$$g_i = \|\boldsymbol{\alpha}_i(\boldsymbol{w}(\alpha)) - \alpha_i\| \sqrt{R_i^2 + 2\lambda n}$$

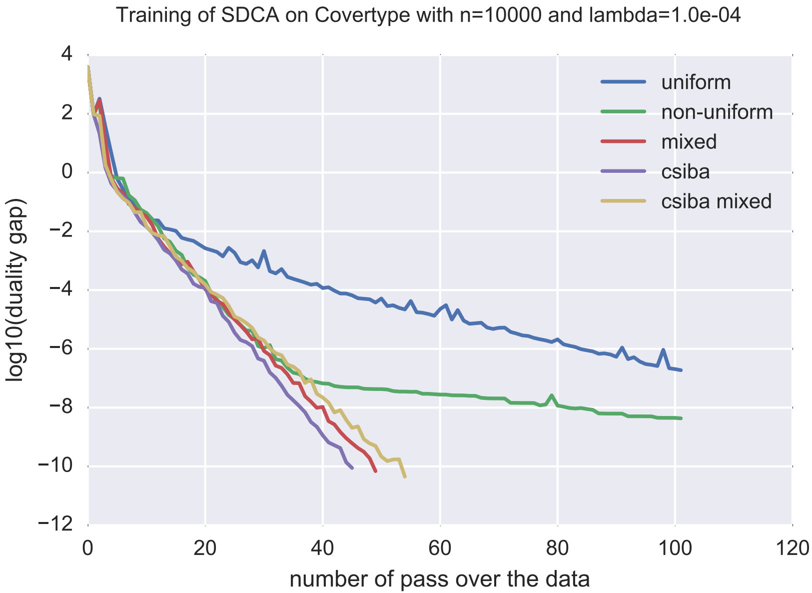
where R_i is the operator norm of A_i.[a]

[a] Csiba, Qu, and Richtarik, "Stochastic Dual Coordinate Ascent with Adaptive Probabilities".

Results 1


SDCA on a gaussian mixture
with n=1000, d=20, k =6, lambda=1.0e-03
log10(duality gap)
number of pass over the data
uniform
gaps
mixed gaps
csiba
mixed csiba
4
2
0
−2
−4
−6
−8
−10
−12
0
5
10
15
20
25
30

Results 2


Training of SDCA on Covertype with n=10000 and lambda=1.0e-04
log10(duality gap)
number of pass over the data
uniform
non-uniform
mixed
csiba
csiba mixed
4
2
0
−2
−4
−6
−8
−10
−12
0
20
40
60
80
100
120

Overview

1. Conditional Random Fields
2. SDCA for Max-likelihood
3. Non-Uniform Sampling
4. Leverage the Structure

Joints and Marginals[5]

Marginal probability on the clique $c \in \mathcal{C}$.

$$\mu_{i,c}(y_c) := \sum_{y' \in \mathcal{Y}_i \,|y'_c = y_{i,c}} \alpha_i(y')$$

The marginals of the sample i live in the local consistency polytope L_i.
If $\mathcal{T} = (\mathcal{C}, \mathcal{S})$ is a junction tree of G:

$$\alpha(y) = \frac{\prod_{c \in \mathcal{C}_{max}} \mu_c(y_c)}{\prod_{s \in \mathcal{S}} \mu_s(y_s)} \tag{2}$$

Junction Tree algorithm = marginalization oracle.

We can compute $p_c(y_c|x_i; w) = \mu'_{i,c}(y_c)$.

[5]Taskar, Guestrin, and Koller, "Max-Margin Markov Networks".

Marginals to the Weights

$$
\begin{aligned}
\boldsymbol{w}(\alpha) &= \frac{1}{\lambda n} \sum_i \mathbf{E}_{y \sim \alpha_i}[\psi_i(y)] \\
&= \frac{1}{\lambda n} \sum_i \sum_{c \in \mathcal{C}_i} \mathbf{E}_{y \sim \alpha_i}[\psi_{i,c}(y_c)] \\
&= \frac{1}{\lambda n} \sum_i \sum_{c \in \mathcal{C}_i} \mathbf{E}_{y_c \sim \mu_{i,c}}[\psi_{i,c}(y_c)]
\end{aligned}
$$

Marginal weights

$$
\mathcal{W}(\mu) := \frac{1}{\lambda n} \sum_i \sum_c B_{i,c} \mu_{i,c}
$$

where $B_{i,c}$ has size $d \times |\mathcal{Y}_c|$, with $\psi_{i,c}(y_c)$ in the column y_c.

Marginals to the Entropy

We express the entropy of the joint probability as a function of the marginals with equation 2.

$$\mathcal{H}(\mu) := H_{|\mathcal{Y}|}(\alpha) = \sum_c H_{|c|}(\mu_c) - \sum_s H_{|s|}(\mu_s)$$

$$\mathcal{D}(\mu||\mu') := D_{KL}(\alpha||\alpha') = \sum_c D(\mu_c||\mu'_c) - \sum_s D(\mu_s||\mu'_s)$$

Remark: We can directly transpose our non-uniform sampling scheme. This is not true for Cisba's scheme.

A New Dual Objective

$$\max_{\forall i, \mu_i \in \mathcal{L}_i} -\frac{\lambda}{2}\|\mathcal{W}(\mu)\|^2 + \frac{1}{n}\sum_i \mathcal{H}_i(\mu_i) \tag{3}$$

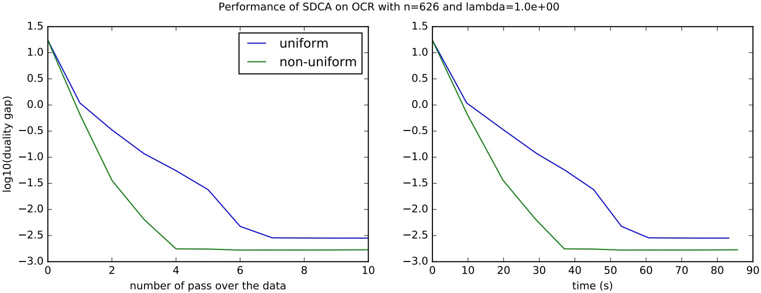
We apply the coordinate ascent directly on this objective

Algorithm 2 SDCA for CRF

Let $\forall i, c, \mu_{i,c}^{(0)} := \frac{1}{|c|}$ and $w^{(0)} := \frac{1}{\lambda n} B\mu^{(0)}$
Let $\forall i g_i = 1$ (optional)
for $k = 0 \dots K$ **do**
  Pick i at random in $\{1, \dots, n\}$ (optionally, proportional to g_i)
  Compute $\forall c, \mu'_{i,c}(y_c) := p(y_c|x; w^{(k)})$ (marginalization oracle)
  Let $g_i = \mathcal{D}(\mu_i||\mu'_i)$ (optional)
  Let $d_i = \mu'_i - \mu_i^{(k)}$ (ascent direction)
  Let $v_i = \frac{1}{\lambda} B_i d_i$ (primal direction)
  Solve $\gamma^* = \arg\max_{\gamma \in [0,1]} \mathcal{H}_i(\mu_i^{(k)} + \gamma d_i) - \frac{\lambda n}{2} \|w^{(k)} + \frac{\gamma}{n} v_i\|^2$ (Line Search)
  Update $\mu_i^{(k+1)} := \mu_i^{(k)} + \gamma^* d_i$
  Update $w^{(k+1)} := w^{(k)} + \frac{\gamma^*}{n} v_i$

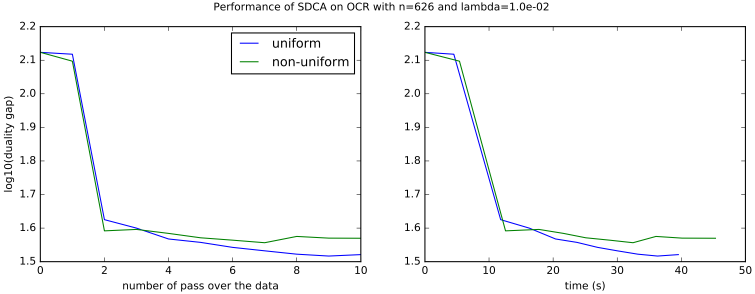
Results 1

No convergence yet.



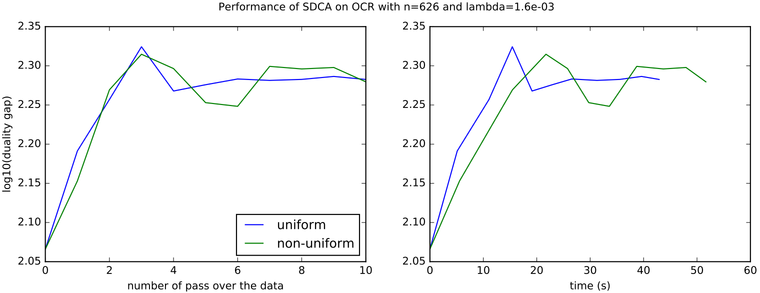
Results 1

No convergence yet.



Results 1

No convergence yet.



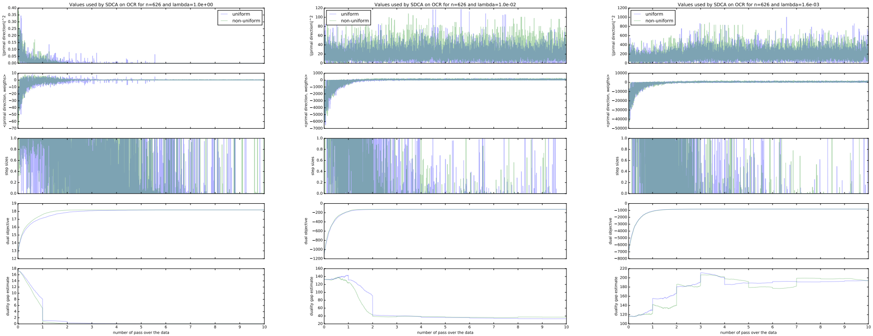


Figure: Some values of interest tracked along the run of SDCA.

Questions?

Appendix 1

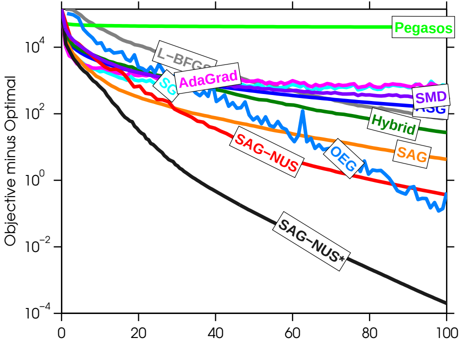


Figure: Training curves for various optimization algorithms on the OCR dataset. The x axis is the number of epochs, while the y axis is the primal suboptimality.

Appendix 1

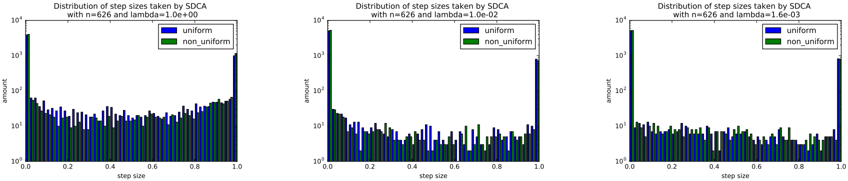


Figure: Distributions of step sizes taken by SDCA. The y axis is a log-scale. A large majority of steps are either taken with full size 1, either not taken at all. When the algorithm works better, with λ large, there are more intermediate step sizes.

Appendix 1

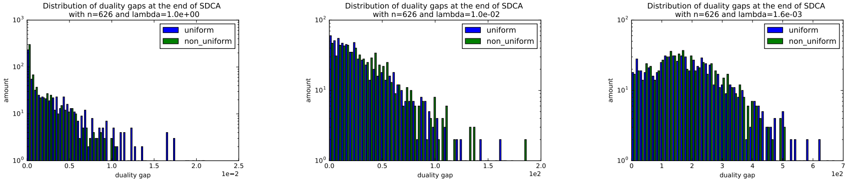


Figure: Distribution of individual duality gaps after the run of SDCA.