# A Tutorial on Primal-Dual Algorithm

Shenlong Wang

University of Toronto

March 31, 2016

# Energy minimization

## MAP Inference for MRFs

- Typical energies consist of a **regularization** term and a **data** term.

$$\min_{\mathbf{x}}\{E(\mathbf{x}) = \mathcal{R}(\mathbf{x}) + \mathcal{D}(\mathbf{x})\}$$

- Used for a wide range of problems.
- Minimizer provides the best configuration to the problem.
- The energy related to the posterior probability via a Gibbs distribution:

$$p(\mathbf{x}) = \frac{1}{Z}\exp(-E(\mathbf{x}))$$

# Energy minimization

## Discrete vs Continuous MRF setting

According to different output space, we have:

- **Discrete setting:** each variable $y_i$ can take a label from a discrete label set $\mathbb{U} \subset \mathbb{Z}$.
- **Continuous setting:** each variable $y_i$ is considered as a continuous value from $\mathbb{U} \subset \mathbb{R}$.

We will focus on continuous setting today.

# Why not Black box convex optimization?[1]

Black box convex optimization

Generic iterative algorithm for convex optimization:

1. Pick any initial vector $x^0 \in \mathbb{R}^n$, set $k = 0$
2. Compute search direction $d^k \in \mathbb{R}^n$
3. Choose step size $\tau^k$ such that $f(x^k + \tau^k d^k) < f(x^k)$
4. Set $x^{k+1} = x^k + \tau^k d^k$, $k = k + 1$
5. Stop if converged, else goto 2

- Plug-and-play, lots of choice: steepest descent, conjugate gradient, newton, quasi-newton (e.g. L-BFGS)
- Do not use the structure of the problems, thus may not be the most efficient choice.
- What if it's difficult to compute $d$?

[1] Image from Cremers (2014)

# Notations

- Variational imaging people use different notations:

$$\min_u \int_{x\in\Omega} |\nabla u| + \frac{\lambda}{2}\|k * u - f\|_2^2 dx$$

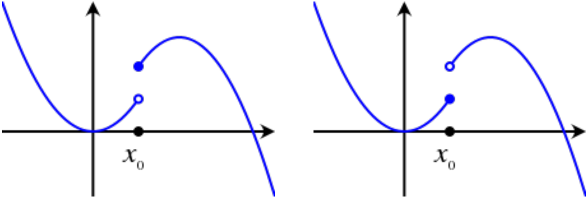
  where $u : \Omega \to \mathbb{R}$ is a function over the continuous image space $\Omega \subset \mathbb{R}^n$ and the objective is a functional of $u$.
- Keep in mind what it looks like in our language:

$$\min_{\mathbf{x}} \|G\mathbf{x}\|_1 + \frac{\lambda}{2}\|K\mathbf{x} - f\|_2^2$$

# Warm-up

## Semi-continuity[2]

- A function $f(x)$ is called lower(upper) **semi-continuous** for a point $\mathbf{x}_0$ if function values for arguments near $\mathbf{x}_0$ are either close to $f(\mathbf{x}_0)$ or greater than (less than) $f(\mathbf{x}_0)$




- Floor function $\lfloor x \rfloor$ is upper semi-continuous, $\lceil x \rceil$ is lower semi-continuous.
- The indicator function of any open set is lower semicontinuous. The indicator function of a closed set is upper semicontinuous.
- Used to convert inequality constraints to objective function.

[2] See https://en.wikipedia.org/wiki/Semi-continuity for a formal definition

- A function $f(x)$ is called **Lipschitz continuous** on $\mathcal{B}_n$ if :

  $$|f(x) - f(y)| \leq L\|x - y\|_\infty, \forall x, y \in \mathcal{B}_n$$

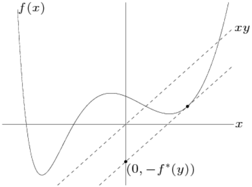
  where constant $L$ is an upper bound to the maximum steepness of $f(x)$
- Stronger than continuous, weaker than continuously differentiable.
- $f = |x|$ is Lipschitz continuous but not continuously differentiable.

# Warm-up

## Convex conjugate

The **convex conjugate** $f^*(\mathbf{y})$ of a function $f(x)$ is defined as:

$$f^*(y) = \sup_{\mathbf{x} \in \mathbf{dom} f} \langle x, y \rangle - f(x)$$




Each pair of $(\mathbf{y}^*, f(\mathbf{y}^*))$ is a tangent line of the function[3]

Examples:

- $f(x) = |x|$:

$$f^*(\mathbf{y}) = \sup_{\mathbf{x}} \langle x, y \rangle - |x| = \begin{cases} 0 & \text{if} \quad |y| < 1 \\ \infty & \text{else} \end{cases}$$

[3] Image from Wikipedia

# Warm-up

## Proximal operator

- The **proximal operator** (or proximal mapping) of a convex function $f$ is:
$$\mathbf{prox}_f(\mathbf{x}) = \arg\min_{\mathbf{u}} \left( f(\mathbf{u}) + \frac{1}{2}\|\mathbf{u} - \mathbf{x}\|_2^2 \right)$$
- $f$ can be nonsmooth, have embedded constraints, ...
- evaluating $\mathbf{prox}_f$ involves solving a convex optimization problem.
- but often has analytic solution, or simple linear-time algorithm.

# Warm-up

## Proximal operator examples

- **quadratic function** $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T P\mathbf{x} + \mathbf{q}^T\mathbf{x} + r$ ($P \succeq 0$):

$$\mathbf{prox}_f(\mathbf{x}) = (I + P)^{-1}(\mathbf{x} - \mathbf{q})$$

- $\ell_1$ **norm** $f(\mathbf{x}) = \|\mathbf{x}\|_1$:

$$\mathbf{prox}_f(x)_i = \begin{cases} x_i - 1 & x_i \geq 1 \\ 0 & |x_i| \leq 1 \\ x_i + 1 & x_i \leq -1 \end{cases} \quad \text{(soft thresholding)}$$

- **logarithmic barrier** $f(\mathbf{x}) = -\sum_{i=1}^n \log x_i$:

$$\mathbf{prox}_f(x)_i = \frac{x_i + \sqrt{x_i^2 + 4}}{2}$$

# Warm-up

## Useful properties of proximal operator

- If $f$ is closed and convex then $\mathbf{prox}_f$ **exists** and is **unique** for all $x$.
- **seperable sum**: if $f(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i)$, then

$$(\mathbf{prox}_f(\mathbf{x}))_i = \mathbf{prox}_{f_i}(x_i)$$

- **fixed point**: the point $\mathbf{x}^*$ minimizes $f$ if and only if $\mathbf{x}^*$ is a fixed point:

$$\mathbf{x}^* = \mathbf{prox}_f(\mathbf{x}^*)$$

- **scaling and translation**: define $h(\mathbf{x}) = f(t\mathbf{x} + \mathbf{b})$ with $t \neq 0$:

$$\mathbf{prox}_h(\mathbf{x}) = \frac{1}{t}(\mathbf{prox}_{t^2 f}(t\mathbf{x} + \mathbf{b}) - \mathbf{b})$$

- **conjugate**:

$$\mathbf{prox}_{tf^*}(\mathbf{x}) = \mathbf{x} - t\mathbf{prox}_{f/t}(\mathbf{x}/t)$$

- keys to design parrallel optimization method

# Proximal gradient method

**Problem**

$$\min_{\mathbf{x}} f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$$

- $g$ convex, differentiable, $\nabla g$ is Lipschitz continuous with constant $L$
- $h$ convex, possibly nondifferentiable; $\mathbf{prox}_h$ is inexpensive
- rules out many methods, e.g. conjugate gradient
- e.g. lasso:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\lambda}{2}\|A\mathbf{x} + \mathbf{b}\|_2^2$$

# Proximal gradient method



$$\min_{\mathbf{x}} f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$$

- $g$ convex, differentiable, $\nabla g$ is Lipschitz continuous with constant $L$
- $h$ convex, possibly nondifferentiable; $\mathbf{prox}_h$ is inexpensive

**proximal gradient algorithm**

$$\mathbf{x}^{(t)} = \mathbf{prox}_h \left( \mathbf{x}^{(t-1)} - \tau_t \nabla g(\mathbf{x}^{(t-1)}) \right)$$

- $O(1/N)$ convergence rate
- i.e. to get $f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq \epsilon$, need $O(1/\epsilon)$ iterations

# Proximal gradient method

## Accelerated gradient method

$$\min_{\mathbf{x}} f(\mathbf{x}) = g(\mathbf{x}) + h(\mathbf{x})$$

- $g$ convex, differentiable, $\nabla g$ is Lipschitz continuous with constant $L$
- $h$ convex, possibly nondifferentiable; $\mathbf{prox}_h$ is inexpensive

**accelerated proximal gradient algorithm**[4]

$$\mathbf{x}^{(t)} = \mathbf{prox}_h \left(\mathbf{y}^{(t-1)} - \tau_t \nabla g(\mathbf{y}^{(t-1)})\right)$$
$$\mathbf{y}^{(t)} = \mathbf{x}^{(t)} + \frac{t-1}{t+2}(\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)})$$

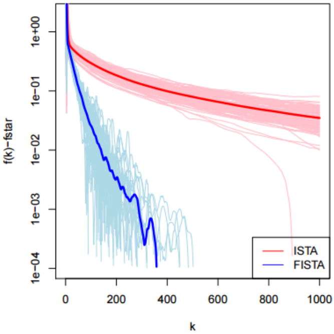
- $O(1/N^2)$ convergence rate for first-order method!
- i.e. to get $f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*) \leq \epsilon$, only need $O(1/\sqrt{\epsilon})$ iterations

[4] Nesterov (2004), Beck and Teboulle (2009)

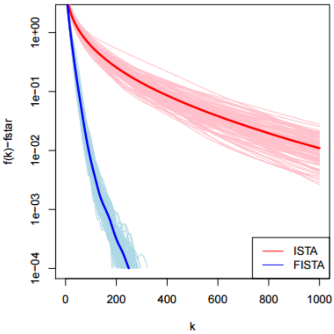
# Proximal gradient method

## Experiments

Lasso regression: 100 instances (with $n = 100$, $p = 500$):




Lasso logistic regression: 100 instances ($n = 100$, $p = 500$):




proximal gradient vs accelerated proximal gradient[5]

[5] Image from Gordon & Tibshirani (2012)

# Motivation

## Energy minimization

$$\min_{\mathbf{x}\in X} f(K\mathbf{x}) + g(\mathbf{x})$$

- $K$ is a linear and continuous operator
- $X$ is Hilbert space
- $f : X \to \mathbb{R} \cup \{\infty\}$, $g : X \to \mathbb{R} \cup \{\infty\}$, convex, not necessarily to be continuous and differentiable, $\mathbf{prox}$ is inexpensive

# Motivation

## Examples

Many problems are within this subclass:

- Lasso:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + \frac{\lambda}{2}\|A\mathbf{x} + \mathbf{b}\|_2^2$$

- Low-level vision:

$$\min_{\mathbf{u},\mathbf{v}} \|\nabla \mathbf{u}\|_1 + \|\nabla \mathbf{v}\|_1 + \lambda\|\rho(\mathbf{u},\mathbf{v})\|_1$$

- Linear programming:

$$\min_{\mathbf{x}} \langle \mathbf{c}, \mathbf{x}\rangle, \quad \text{subject to} \quad \begin{cases} A\mathbf{x} & = & \mathbf{b} \\ \mathbf{x} & \geq & 0 \end{cases}$$

- SVM:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} \max(0, 1 - y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b))$$

# Primal-dual algorithm

## Problem formulation

$$\min_{\mathbf{x}\in X} f(K\mathbf{x}) + g(\mathbf{x}) \quad \text{(primal)}$$

Recall the convex conjugate: $f^*(\mathbf{y}) = \langle K\mathbf{x}, \mathbf{y}\rangle - f(K\mathbf{x})$, we have:

$$\min_{\mathbf{x}\in X} \max_{\mathbf{y}\in Y} \langle K\mathbf{x}, \mathbf{y}\rangle + g(\mathbf{x}) - f^*(\mathbf{y}) \quad \text{(primal-dual)}$$

$$\max_{\mathbf{y}\in Y} -(f^*(\mathbf{y}) + g^*(-K^*\mathbf{y})) \quad \text{(dual)}$$

Primal-dual gap:

$$f(K\mathbf{x}) + g(\mathbf{x}) + f^*(\mathbf{y}) + g^*(-K^*\mathbf{y})$$

For convex function, primal-dual gap will be 0 at the optimal solution.

# Primal-dual algorithm

## Optimal condition

Today we focus on primal-dual problem:

$$\min_{\mathbf{x}\in X}\max_{\mathbf{y}\in Y}\langle K\mathbf{x},\mathbf{y}\rangle + g(\mathbf{x}) - f^*(\mathbf{y}) \quad \text{(primal-dual)}$$

Why primal-dual?

- proximal operator for $f(K\mathbf{x})$ is not trivial.
- but we can get proximal operator $f^*(\mathbf{x})$ and $g(\mathbf{x})$ easily

A saddle point $(\mathbf{x},\mathbf{y}) \in X \times Y$ of this min-max function should satisfy:

$$\begin{cases} K\hat{\mathbf{x}} - \partial f^*(\hat{\mathbf{y}}) \ni 0 \\ K^*\hat{\mathbf{y}} + \partial g(\hat{\mathbf{x}}) \ni 0 \end{cases}$$

We iterate according to this condition.

# Primal-dual algorithm

**The algorithm**[6]

$$\min_{\mathbf{x}\in X}\max_{\mathbf{y}\in Y}\langle K\mathbf{x},\mathbf{y}\rangle + g(\mathbf{x}) - f^*(\mathbf{y})$$

- Choose step size $\sigma > 0$ and $\tau > 0$, so that $\sigma\tau L^2 < 1$, where $L = \|K\|$, and $\theta \in [0,1]$.
- Choose initialization $(\mathbf{x}^0, \mathbf{y}^0)$
- For each iteration:

$$\begin{cases} \mathbf{y}^{(n+1)} &= \mathbf{prox}_{f^*}(\mathbf{y}^{(n)} + \sigma K\bar{\mathbf{x}}^{(n)}) & \text{(dual proximal)} \\ \mathbf{x}^{(n+1)} &= \mathbf{prox}_{g}(\mathbf{x}^{(n)} - \tau K^*\bar{\mathbf{y}}^{(n+1)}) & \text{(primal proximal)} \\ \bar{\mathbf{x}}^{(n+1)} &= \mathbf{x}^{(n+1)} + \theta(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) & \text{(extrapolation)} \end{cases}$$

- Essentially alternately do proximal gradient descent for $\mathbf{x}$ and $\mathbf{y}$.

[6] Chambolle and Pock (2011), Pock, Cremers, Bischof, Chambolle (2009)

# Primal-dual algorithm

## Convergence

The algorithm's convergence rate depending on different types of the problem[7]:

- Completely non-smooth problem: $O(1/N)$ for the duality gap.
- Sum of a smooth and non-smooth: $O(1/N^2)$ for $\|\mathbf{x} - \mathbf{x}^*\|^2$.
- Completely smooth problem: $O(\omega^N), \omega < 1$ for $\|\mathbf{x} - \mathbf{x}^*\|^2$.

[7] See Chambolle and Pock (2011) for a detailed proof.

# Discussion

## Parallel implementation

$$
\begin{cases}
\mathbf{y}^{(n+1)} & = & \mathbf{prox}_{f^*}(\mathbf{y}^{(n)} + \sigma K \bar{\mathbf{x}}^{(n)}) & \text{(dual proximal)} \\
\mathbf{x}^{(n+1)} & = & \mathbf{prox}_{g}(\mathbf{x}^{(n)} - \tau K^* \bar{\mathbf{y}}^{(n+1)}) & \text{(primal proximal)} \\
\bar{\mathbf{x}}^{(n+1)} & = & \mathbf{x}^{(n+1)} + \theta(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) & \text{(extrapolation)}
\end{cases}
$$

Problems we usually have in vision

- $\mathbf{x}$ and $\mathbf{y}$ are defined on a regular grid.
- $f$ and $g$ is usually in a **separable sum** format.
- Small number of variables involved gradient part $K\mathbf{x}$ (high-order potential)
- perfect for GPU parallel computing!

# Discussion

**Arrow-Hurwicz method ($\theta = 0$)**[8]

$$\begin{cases} \mathbf{y}^{(n+1)} & = & \mathbf{prox}_{f^*}(\mathbf{y}^{(n)} + \sigma K \bar{\mathbf{x}}^{(n)}) & \text{(dual proximal)} \\ \mathbf{x}^{(n+1)} & = & \mathbf{prox}_{g}(\mathbf{x}^{(n)} - \tau K^* \bar{\mathbf{y}}^{(n+1)}) & \text{(primal proximal)} \end{cases}$$

- Also tackles primal-dual method
- Without the ‘momentum’ step
- Theoretically lose $O(1/N^2)$ convergence guarantee (or people haven’t proved it yet)
- In practice, for some method it is still fast

[8] Arrow, Hurwicz, Uzawa (1958)

# Discussion

## Alternating direction method of multipliers (ADMM)

$$\min_{\mathbf{x}\in X} f(K\mathbf{x}) + g(\mathbf{x}) \quad \text{(primal)}$$

We can conduct a decomposition:

$$\min_{\mathbf{x}\in X} f(\mathbf{y}) + g(\mathbf{x}) \;\; \text{subject to} \quad K\mathbf{x} - \mathbf{y} = 0$$

Solving the following augmented Lagrangian multipliers problem:

$$L_\tau(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{y}) + g(\mathbf{x}) + \mathbf{z}^T(K\mathbf{x} - \mathbf{y}) + \frac{\tau}{2}\|K^*\mathbf{y} - \mathbf{x}\|_2^2$$

$$\begin{cases} \mathbf{y}^{(n+1)} & = & \arg\min_{\mathbf{y}} f(\mathbf{y}) + \langle \mathbf{y}, \mathbf{z}^{(n)} \rangle + \frac{\tau}{2}\|K\mathbf{x}^{(n)} - \mathbf{y}\|_2^2 & \text{(primal)} \\ \mathbf{x}^{(n+1)} & = & \arg\min_{\mathbf{x}} g(\mathbf{x}) - \langle K\mathbf{x}, \mathbf{z}^{(n)} \rangle + \frac{\tau}{2}\|K\mathbf{x} - \mathbf{y}^{(n+1)}\|_2^2 & \text{(primal)} \\ \mathbf{z}^{(n+1)} & = & \mathbf{z}^{(n)} + \tau(K\mathbf{x} - \mathbf{y}) & \text{(dual)} \end{cases}$$

- Primal-dual method is equivalent to ADMM if $K = I$.
- But in the general case primal-dual is usually faster, since solving the subproblems of ADMM is harder.

# Discussion

## Experimental results[9]

| | $\lambda = 16$ | | $\lambda = 8$ | |
|---|---|---|---|---|
| | $\varepsilon = 10^{-4}$ | $\varepsilon = 10^{-6}$ | $\varepsilon = 10^{-4}$ | $\varepsilon = 10^{-6}$ |
| PD | 108 (1.95s) | 937 (14.55s) | 174 (2.76s) | 1479 (23.74s) |
| AHZC | 65 (0.98s) | 634 (9.19s) | 105 (1.65s) | 1001 (14.48s) |
| FISTA | 107 (2.11s) | 999 (20.36s) | 173 (3.84s) | 1540 (29.48s) |
| NEST | 106 (3.32s) | 1213 (38.23s) | 174 (5.54s) | 1963 (58.28s) |
| ADMM | 284 (4.91s) | 25584 (421.75s) | 414 (7.31s) | 33917 (547.35s) |
| PGD | 620 (9.14s) | 58804 (919.64s) | 1621 (23.25s) | – |
| CFP | 1396 (20.65s) | – | 3658 (54.52s) | – |

ROF-model, 500×375 grayscale image, $\epsilon$ error tolerance

- AHZC: Arrow-Hurwicz primal-dual method
- FISTA: Fast iterative shrinkage threshold, $O(1/N^2)$ convergence rate
- NEST: Nesterov's method on dual problem, $O(1/N^2)$ convergence rate
- ADMM: Alternating direction method of multipliers
- PGM: Proximal gradient method , $O(1/N)$ convergence rate
- CFP: Fixed point method on dual

[9] Table from Chambolle and Pock (2010)

# Example

## Image denoising

ROF[10] model for image denoising:

$$\min_{\mathbf{x}} \|\nabla \mathbf{x}\|_1 + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{u}\|_2^2$$

$f(\mathbf{x}) = \sum_i \|\nabla x_i\|_1$, where $\nabla x_i$ is a two-dimensional intensity gradient vector at image pixel $i$

$$f^*(\mathbf{y}) = \delta_{\ell_\infty}(\mathbf{y}) = \begin{cases} 0 & \mathbf{y} \in \mathcal{P} \\ \infty & \mathbf{y} \notin \mathcal{P} \end{cases}, \quad \mathcal{P} = \{\mathbf{p} : \forall i, |p_i| < 1\}$$

Proximal operator for convex-set indicator function is just euclidean projecting onto the feasible closed set $\mathcal{P}$. Thus:

$$\mathbf{prox}_{f^*}(\mathbf{y}) = \frac{\mathbf{y}}{\max(\|\mathbf{y}\|, 1)}$$

Proximal operator for $\frac{\lambda}{2}\|\mathbf{x} - \mathbf{u}\|_2^2$ is in closed form:

$$\mathbf{prox}_g(\mathbf{x}) = \frac{\mathbf{x} + \lambda\tau\mathbf{u}}{1 + \lambda\tau}$$

[10] Rudin, Osher, Fatemi (1992)

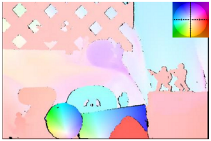
# Example

## TV-$L^1$ Optical flow[11]

$$\min_{\mathbf{u},\mathbf{v}} \|\nabla \mathbf{u}\|_1 + \|\nabla \mathbf{v}\|_1 + \lambda \|\rho(\mathbf{u},\mathbf{v})\|_1$$

- $\mathbf{u}, \mathbf{v}$: horizontal and vertical motion field
- $\rho(\mathbf{u}, \mathbf{v})$ first-order Taylor approximation of photometric error
- $\mathbf{u}_k$: estimation of inverse depth from single view

(a) Input



(b) Ground truth

(c) Estimated motion

[11] Zach, Pock, Bischof (2007)

# Example

## Linear Programming

$$\min_{\mathbf{x}} \langle \mathbf{c}, \mathbf{x} \rangle, \quad \text{subject to} \quad \begin{cases} A\mathbf{x} & = & \mathbf{b} \\ \mathbf{x} & \geq & 0 \end{cases}$$

Introducing Lagrange multipliers $\mathbf{y}$

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \langle A\mathbf{x} - b, \mathbf{y} \rangle + \langle \mathbf{c}, \mathbf{x} \rangle, \quad \text{subject to} \quad \mathbf{x} \geq 0$$

Applying primal-dual algorithm:

$$\begin{cases} \mathbf{y}^{(n+1)} & = & \mathbf{y}^{(n)} + \sigma A \bar{\mathbf{x}}^{(n)} \\ \mathbf{x}^{(n+1)} & = & \mathbf{proj}_{[0,\infty)}(\mathbf{x}^{(n)} - \tau(A^T \mathbf{y}^{(n+1)} + \mathbf{c})) \\ \bar{\mathbf{x}}^{(n+1)} & = & \mathbf{x}^{(n+1)} + \theta(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) \end{cases}$$

where $\mathbf{proj}_{[0,\infty)}$ is the simply truncation function.

# Example

## Discrete MRF inference

$$\min_{\forall i, x_i \in \mathcal{L}} \sum_i \theta_i(x_i) + \sum_f \theta_f(\mathbf{x}_f)$$

LP relaxation:

$$\begin{aligned}
\min_{\forall i, x_i \in \mathcal{L}} & \sum_i \sum_{x_i} \theta_i(x_i)\mu_i(\mathbf{x}_i) + \sum_i \sum_{\mathbf{x}_f} \theta_f(\mathbf{x}_f)\mu_f(\mathbf{x}_f) \\
\text{subject to } & \sum_{x_i} \mu_i(x_i) = 1, \forall i \\
& \sum_{\mathbf{x}_f} \mu_f(\mathbf{x}_f) = 1, \forall f \\
& \sum_{\mathbf{x}_{f1}} \mu_f(\mathbf{x}_f) = \mu_i(x_i), \forall f, i, x_i
\end{aligned}$$

# Example

## Binary labeling[12]

Potts-model over 2D grid can be written as following convex relaxation:

$$\min_{\mathbf{x}} \|D\mathbf{x}\|_1 + \langle \mathbf{x}, \mathbf{w} \rangle \quad \text{subject to} \quad 0 \le x_i \le 1, \forall i$$

Preconditioned primal-dual algorithm:

$$\begin{cases} \mathbf{y}^{(n+1)} & = & \mathbf{proj}_{[-1,1]}(\mathbf{y}^{(n)} + \Sigma A \bar{\mathbf{x}}^{(n)}) \\ \mathbf{x}^{(n+1)} & = & \mathbf{proj}_{[0,\infty)}(\mathbf{x}^{(n)} - T(A^T \mathbf{y}^{(n+1)} + \mathbf{w})) \\ \bar{\mathbf{x}}^{(n+1)} & = & \mathbf{x}^{(n+1)} + \theta(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) \end{cases}$$

| MAXFLOW | PD | P-PD | P-PD-GPU |
|---|---|---|---|
| 0.160s | 15.75s | 8.56s | 0.045s |

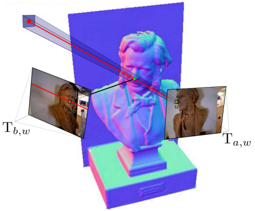
[12] Image from Cremers (2014)

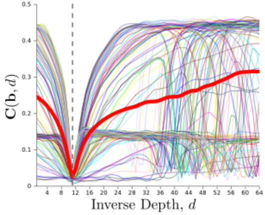
# Example

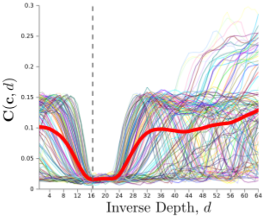
## Multi-view stereo reconstruction[13]

$$\min_{\mathbf{x}} \lambda(\mathbf{x})\|\nabla \mathbf{x}\|_\epsilon + C(\mathbf{x})$$

- $\mathbf{x}$: inverse depth estimation
- $\lambda(\mathbf{x})$: element-wise weighting function
- $\|\cdot\|_\epsilon$ robust Huber loss function
- $C(\mathbf{x})$ non-convex matching function




[13] Newcombie et al. (2011)

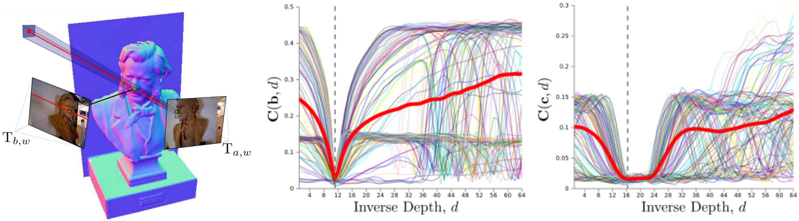
# Example

## Multi-view stereo reconstruction[14]

Introducing auxiliary variable $\mathbf{u}$:

$$\min_{\mathbf{x}} \lambda(\mathbf{x})\|\nabla \mathbf{x}\|_\epsilon + C(\mathbf{u}) + \frac{1}{2\theta}\|\mathbf{x}-\mathbf{u}\|_2^2$$

- Minimize $C(\mathbf{u}) + \frac{1}{2\theta}\|\mathbf{x}-\mathbf{u}\|_2^2$ wrt. $\mathbf{u}$ by smart brute-force search
- Minimize $\|\nabla \mathbf{x}\|_\epsilon + \frac{1}{2\theta}\|\mathbf{x}-\mathbf{u}\|_2^2$ wrt. $\mathbf{x}$ by primal-dual
- Reducing $\theta$




[14] Newcombie et al. (2011)

# Summary

- First-order primal-dual algorithm for a class of structured convex optimization problems
- Objective function can be non-differentiable
- Easy to implement (we just need to derive the proximal operators)
- Optimal convergence rate on multiple sub-classes

# Reference


- Chambolle, Antonin, and Thomas Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging." Journal of Mathematical Imaging and Vision 40.1 (2011): 120-145.
- Nesterov, Yurii. Introductory lectures on convex optimization: A basic course. Vol. 87. Springer Science & Business Media, 2013.
- Nesterov, Yurii. Gradient methods for minimizing composite objective function. No. CORE Discussion Papers (2007/76). UCL, 2007.
- Parikh, Neal, and Stephen P. Boyd. "Proximal Algorithms." Foundations and Trends in optimization 1.3 (2014): 127-239.
- Newcombe, Richard A., Steven J. Lovegrove, and Andrew J. Davison. "DTAM: Dense tracking and mapping in real-time." Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.
- Pock, Thomas, et al. "Global solutions of variational models with convex regularization." SIAM Journal on Imaging Sciences 3.4 (2010): 1122-1145.
- Nieuwenhuis, Claudia, Eno TÃűppe, and Daniel Cremers. "A survey and comparison of discrete and continuous multi-label optimization approaches for the Potts model." International journal of computer vision 104.3 (2013): 223-240.
- Zach, Christopher, Thomas Pock, and Horst Bischof. "A duality based approach for realtime TV-L 1 optical flow." Pattern Recognition. Springer Berlin Heidelberg, 2007. 214-223.
- http://gpu4vision.icg.tugraz.at/