Regularized Frank-Wolfe for Dense CRFs: Generalizing Mean Field and Beyond

Đ.Khuê Lê-Huu **Karteek Alahari**
Inria

December 2021

Context and motivation

Dense conditional random fields (CRFs) [Krähenbühl and Koltun, 2011]:

- Once a *highly-successful* paradigm for semantic segmentation.

Context and motivation

Dense conditional random fields (CRFs) [Krähenbühl and Koltun, 2011]:

- Once a *highly-successful* paradigm for semantic segmentation.
- Used in most *top-performing systems* on PASCAL VOC (2011-2017).

Context and motivation

Dense conditional random fields (CRFs) [Krähenbühl and Koltun, 2011]:

- Once a *highly-successful* paradigm for semantic segmentation.
- Used in most *top-performing systems* on PASCAL VOC (2011-2017).
- *Fell out of favor since 2017* *[Lin et al., 2017]* as CNNs got stronger.

Context and motivation

Dense conditional random fields (CRFs) [Krähenbühl and Koltun, 2011]:

- Once a *highly-successful* paradigm for semantic segmentation.
- Used in most *top-performing systems* on PASCAL VOC (2011-2017).
- *Fell out of favor since 2017* *[Lin et al., 2017]* as CNNs got stronger.

We revisit dense CRFs with several contributions!

Context and motivation

Dense conditional random fields (CRFs) [Krähenbühl and Koltun, 2011]:

- Once a *highly-successful* paradigm for semantic segmentation.
- Used in most *top-performing systems* on PASCAL VOC (2011-2017).
- *Fell out of favor since 2017 [Lin et al., 2017]* as CNNs got stronger.

We revisit dense CRFs with several contributions!

1. *Algorithmic:* New algorithms & their connections to existing ones.

Context and motivation

Dense conditional random fields (CRFs) [Krähenbühl and Koltun, 2011]:

- Once a *highly-successful* paradigm for semantic segmentation.
- Used in most *top-performing systems* on PASCAL VOC (2011-2017).
- *Fell out of favor since 2017* *[Lin et al., 2017]* as CNNs got stronger.

We revisit dense CRFs with several contributions!

1. *Algorithmic:* New algorithms & their connections to existing ones.
2. *Theoretical:* Unified convergence & tightness analysis.

Context and motivation

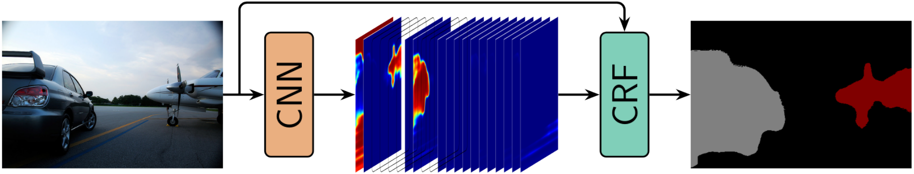
Dense conditional random fields (CRFs) [Krähenbühl and Koltun, 2011]:

- Once a *highly-successful* paradigm for semantic segmentation.
- Used in most *top-performing systems* on PASCAL VOC (2011-2017).
- *Fell out of favor since 2017* *[Lin et al., 2017]* as CNNs got stronger.

We revisit dense CRFs with several contributions!

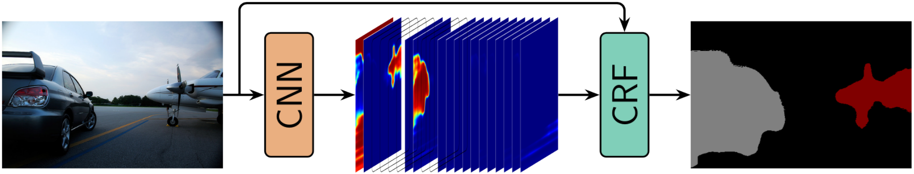
1. *Algorithmic:* New algorithms & their connections to existing ones.
2. *Theoretical:* Unified convergence & tightness analysis.
3. *Practical:* Encouraging results: 88.0 mIoU on PASCAL VOC $\rightarrow$ dense CRFs could still be relevant.

Background on CRFs



Given CNN output, CRF computes final prediction by *minimizing an energy*.

Background on CRFs



Given CNN output, CRF computes final prediction by *minimizing an energy*.

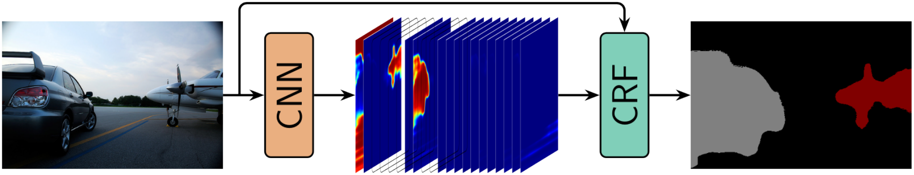
$$\min_{\mathbf{x}} E(\mathbf{x}) \triangleq \frac{1}{2}\mathbf{x}^\top \mathbf{P}\mathbf{x} + \mathbf{u}^\top \mathbf{x} \quad \text{s.t. } \mathbf{x} \in \{0,1\}^{nd}, \mathbf{1}^\top \mathbf{x}_i = 1 \; \forall i \in \mathcal{V}.$$

Gaussian kernels

CNN output

one-hot encoding
(n pixels, d classes)

Background on CRFs



Given CNN output, CRF computes final prediction by *minimizing an energy*.

$$\min_{\mathbf{x}} E(\mathbf{x}) \triangleq \frac{1}{2}\mathbf{x}^\top \mathbf{P}\mathbf{x} + \mathbf{u}^\top \mathbf{x} \quad \text{s.t. } \mathbf{x} \in \{0,1\}^{nd}, \mathbf{1}^\top \mathbf{x}_i = 1 \ \forall i \in \mathcal{V}.$$

Gaussian kernels ($\mathbf{P}$)

CNN output ($\mathbf{u}$)

one-hot encoding ($\mathbf{x}$)
(n pixels, d classes)

Energy minimization is also known as *MAP inference*.

Solving MAP inference in dense CRFs

Continuous relaxation:

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathcal{X} \triangleq \left\{ \mathbf{x} \in [0,1]^{nd} : \mathbf{1}^\top \mathbf{x}_i = 1 \ \forall i \in \mathcal{V} \right\}.$$

Solving MAP inference in dense CRFs

Continuous relaxation:

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathcal{X} \triangleq \left\{\mathbf{x} \in [0,1]^{nd} : \mathbf{1}^\top \mathbf{x}_i = 1 \ \forall i \in \mathcal{V}\right\}.$$

→ **Parallel mean field** [Krähenbühl and Koltun, 2011]

$$\mathbf{x}^{k+1} = \text{softmax}(-\mathbf{P}\mathbf{x}^k - \mathbf{u}).$$

Solving MAP inference in dense CRFs

Continuous relaxation:

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathcal{X} \triangleq \left\{ \mathbf{x} \in [0,1]^{nd} : \mathbf{1}^\top \mathbf{x}_i = 1 \ \forall i \in \mathcal{V} \right\}.$$

→ **Parallel mean field** [Krähenbühl and Koltun, 2011]

$$\mathbf{x}^{k+1} = \text{softmax}(-\mathbf{P}\mathbf{x}^k - \mathbf{u}).$$

→ **Frank-Wolfe method** [Frank and Wolfe, 1956, Lê-Huu and Paragios, 2018]

$$\mathbf{p}^k \in \operatorname*{argmin}_{\mathbf{p} \in \mathcal{X}} \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle, \qquad \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k (\mathbf{p}^k - \mathbf{x}^k).$$

Solving MAP inference in dense CRFs

Continuous relaxation:

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathcal{X} \triangleq \left\{ \mathbf{x} \in [0,1]^{nd} : \mathbf{1}^\top \mathbf{x}_i = 1 \ \forall i \in \mathcal{V} \right\}.$$

→ **Parallel mean field** [Krähenbühl and Koltun, 2011]

$$\mathbf{x}^{k+1} = \text{softmax}(-\mathbf{P}\mathbf{x}^k - \mathbf{u}).$$

✓ Fast, differentiable, allowing CNN-CRF end-to-end training.
✗ Weak in terms of energy minimization.

→ **Frank-Wolfe method** [Frank and Wolfe, 1956, Lê-Huu and Paragios, 2018]

$$\mathbf{p}^k \in \underset{\mathbf{p} \in \mathcal{X}}{\text{argmin}} \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle, \qquad \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k(\mathbf{p}^k - \mathbf{x}^k).$$

Solving MAP inference in dense CRFs

Continuous relaxation:

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathcal{X} \triangleq \left\{ \mathbf{x} \in [0,1]^{nd} : \mathbf{1}^\top \mathbf{x}_i = 1 \ \forall i \in \mathcal{V} \right\}.$$

→ **Parallel mean field** [Krähenbühl and Koltun, 2011]

$$\mathbf{x}^{k+1} = \text{softmax}(-\mathbf{P}\mathbf{x}^k - \mathbf{u}).$$

✓ Fast, differentiable, allowing CNN-CRF end-to-end training.

✗ Weak in terms of energy minimization.

→ **Frank-Wolfe method** [Frank and Wolfe, 1956, Lê-Huu and Paragios, 2018]

$$\mathbf{p}^k \in \operatorname*{argmin}_{\mathbf{p} \in \mathcal{X}} \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle, \qquad \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k (\mathbf{p}^k - \mathbf{x}^k).$$

✓ Fast, stronger in terms of energy minimization.

✗ Backpropagation not possible.

Solving MAP inference in dense CRFs

Continuous relaxation:

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathcal{X} \triangleq \left\{ \mathbf{x} \in [0,1]^{nd} : \mathbf{1}^\top \mathbf{x}_i = 1 \ \forall i \in \mathcal{V} \right\}.$$

→ **Parallel mean field** [Krähenbühl and Koltun, 2011]

$$\mathbf{x}^{k+1} = \text{softmax}(-\mathbf{P}\mathbf{x}^k - \mathbf{u}).$$

✓ Fast, differentiable, allowing CNN-CRF end-to-end training.
✗ Weak in terms of energy minimization.

→ **Frank-Wolfe method** [Frank and Wolfe, 1956, Lê-Huu and Paragios, 2018]

$$\mathbf{p}^k \in \operatorname*{argmin}_{\mathbf{p} \in \mathcal{X}} \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle, \qquad \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k (\mathbf{p}^k - \mathbf{x}^k).$$

✓ Fast, stronger in terms of energy minimization.
✗ Backpropagation not possible.

Differentiable a.e.
but *the gradient is zero!*

Simple remedy for Frank-Wolfe

Our proposed solution to the zero-gradient issue of Frank-Wolfe.

Simple remedy for Frank-Wolfe

Our proposed solution to the zero-gradient issue of Frank-Wolfe.

- Zero gradient: $\mathbf{p}^k \in \operatorname{argmin}_{\mathbf{p}\in\mathcal{X}} \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle$.

Simple remedy for Frank-Wolfe

Our proposed solution to the zero-gradient issue of Frank-Wolfe.

- Zero gradient: $\mathbf{p}^k \in \operatorname{argmin}_{\mathbf{p}\in\mathcal{X}} \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle$.

$\rightarrow$ **Replacing with *approximate updates***

$$\mathbf{p}^k \in \operatorname*{argmin}_{\mathbf{p}\in\mathcal{X}} \left\{ \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle + r(\mathbf{p}) \right\},$$

regularizer

Simple remedy for Frank-Wolfe

Our proposed solution to the zero-gradient issue of Frank-Wolfe.

- Zero gradient: $\mathbf{p}^k \in \text{argmin}_{\mathbf{p}\in\mathcal{X}} \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle$.

→ **Replacing with *approximate updates***

$$\mathbf{p}^k \in \underset{\mathbf{p}\in\mathcal{X}}{\text{argmin}} \left\{ \left\langle \nabla E(\mathbf{x}^k), \mathbf{p} \right\rangle + r(\mathbf{p}) \right\},$$

regularizer

→ With suitable regularizers:
✓ Fast, strong in terms of energy minimization.
✓ Successful backpropagation.

General regularized Frank-Wolfe for MAP inference

We go further and propose an even more powerful algorithm!

General regularized Frank-Wolfe for MAP inference

We go further and propose an even more powerful algorithm!

1. Choose r, f, g such that $f + g = E + r + \delta_{\mathcal{X}}$.

General regularized Frank-Wolfe for MAP inference

We go further and propose an even more powerful algorithm!

1. Choose r, f, g such that $f + g = E + r + \delta_{\mathcal{X}}$.
2. Iterate until convergence:

$$\mathbf{p}^k \in \operatorname*{argmin}_{\mathbf{p}} \left\{ \left\langle \nabla f(\mathbf{x}^k), \mathbf{p} \right\rangle + g(\mathbf{p}) \right\}, \quad \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k(\mathbf{p}^k - \mathbf{x}^k).$$

General regularized Frank-Wolfe for MAP inference

We go further and propose an even more powerful algorithm!

1. Choose r, f, g such that $f + g = E + r + \delta_{\mathcal{X}}$.
2. Iterate until convergence:

$$\mathbf{p}^k \in \operatorname*{argmin}_{\mathbf{p}} \left\{ \left\langle \nabla f(\mathbf{x}^k), \mathbf{p} \right\rangle + g(\mathbf{p}) \right\}, \quad \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k(\mathbf{p}^k - \mathbf{x}^k).$$

3. Rounding: convert $\mathbf{x}$ to a discrete solution.

General regularized Frank-Wolfe for MAP inference

We go further and propose an even more powerful algorithm!

1. Choose r, f, g such that $f + g = E + r + \delta_{\mathcal{X}}$.
2. Iterate until convergence:

$$\mathbf{p}^k \in \operatorname*{argmin}_{\mathbf{p}} \left\{ \left\langle \nabla f(\mathbf{x}^k), \mathbf{p} \right\rangle + g(\mathbf{p}) \right\}, \quad \mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k (\mathbf{p}^k - \mathbf{x}^k).$$

3. Rounding: convert $\mathbf{x}$ to a discrete solution.

known as *generalized conditional gradient* for minimizing $f + g$ [Mine and Fukushima, 1981]

General regularized Frank-Wolfe for MAP inference

Why more powerful?

General regularized Frank-Wolfe for MAP inference

Why more powerful?

Flexibility in choosing r, f, g allows:

1. Easily obtaining new algorithms.

General regularized Frank-Wolfe for MAP inference

Why more powerful?

Flexibility in choosing r, f, g allows:

1. Easily obtaining new algorithms.
2. Making connections to existing ones.

General regularized Frank-Wolfe for MAP inference

Why more powerful?

Flexibility in choosing r, f, g allows:

1. Easily obtaining new algorithms.
2. Making connections to existing ones.
3. Unifying theoretical analysis for all these old and new algorithms.

Instantiations of regularized Frank-Wolfe

Our method leads to *new algorithms* for MAP inference by simple instantiations!

Instantiations of regularized Frank-Wolfe

Our method leads to *new algorithms* for MAP inference by simple instantiations!

- *Euclidean Frank-Wolfe:*

$$\mathbf{p}^k = \operatorname*{argmin}_{\mathbf{p}\in\mathcal{X}} \left\{ \left\langle \mathbf{P}\mathbf{x}^k + \mathbf{u}, \mathbf{p} \right\rangle + \frac{\lambda}{2} \|\mathbf{p}\|_2^2 \right\} = \Pi_{\mathcal{X}} \left(-\frac{1}{\lambda}(\mathbf{P}\mathbf{x}^k + \mathbf{u}) \right).$$

Instantiations of regularized Frank-Wolfe

Our method leads to *new algorithms* for MAP inference by simple instantiations!

- *Euclidean Frank-Wolfe:*

$$\mathbf{p}^k = \operatorname*{argmin}_{\mathbf{p}\in\mathcal{X}} \left\{ \left\langle \mathbf{P}\mathbf{x}^k + \mathbf{u}, \mathbf{p} \right\rangle + \frac{\lambda}{2} \|\mathbf{p}\|_2^2 \right\} = \Pi_{\mathcal{X}} \left(-\frac{1}{\lambda}(\mathbf{P}\mathbf{x}^k + \mathbf{u}) \right).$$

- *Entropic Frank-Wolfe:*

$$\mathbf{p}^k = \operatorname*{argmin}_{\mathbf{p}\in\mathcal{X}} \left\{ \left\langle \mathbf{P}\mathbf{x}^k + \mathbf{u}, \mathbf{p} \right\rangle - \lambda H(\mathbf{p}) \right\} = \operatorname{softmax} \left(-\frac{1}{\lambda}(\mathbf{P}\mathbf{x}^k + \mathbf{u}) \right),$$

where $H(\mathbf{x}) = -\sum_{i,s} x_{is} \log x_{is}$ (entropy).

Instantiations of regularized Frank-Wolfe

Our method leads to *new algorithms* for MAP inference by simple instantiations!

- *Euclidean Frank-Wolfe:*

$$\mathbf{p}^k = \operatorname*{argmin}_{\mathbf{p}\in\mathcal{X}} \left\{ \left\langle \mathbf{P}\mathbf{x}^k + \mathbf{u}, \mathbf{p} \right\rangle + \frac{\lambda}{2} \|\mathbf{p}\|_2^2 \right\} = \Pi_{\mathcal{X}} \left(-\frac{1}{\lambda}(\mathbf{P}\mathbf{x}^k + \mathbf{u}) \right).$$

- *Entropic Frank-Wolfe:*

$$\mathbf{p}^k = \operatorname*{argmin}_{\mathbf{p}\in\mathcal{X}} \left\{ \left\langle \mathbf{P}\mathbf{x}^k + \mathbf{u}, \mathbf{p} \right\rangle - \lambda H(\mathbf{p}) \right\} = \operatorname{softmax} \left(-\frac{1}{\lambda}(\mathbf{P}\mathbf{x}^k + \mathbf{u}) \right),$$

  where $H(\mathbf{x}) = -\sum_{i,s} x_{is} \log x_{is}$ (entropy).

- *Other variants:* ℓ_p norm, lasso, binary entropy, etc.

Instantiations of regularized Frank-Wolfe

Multiple existing algorithms are also special cases!

Instantiations of regularized Frank-Wolfe

Multiple existing algorithms are also special cases!

- *Parallel mean field* *[Krähenbühl and Koltun, 2011]*

$$\mathbf{x}^{k+1} = \mathrm{softmax}(-\mathbf{P}\mathbf{x}^k - \mathbf{u}).$$

Instantiations of regularized Frank-Wolfe

Multiple existing algorithms are also special cases!

- *Parallel mean field* [Krähenbühl and Koltun, 2011]

$$\mathbf{x}^{k+1} = \mathrm{softmax}(-\mathbf{P}\mathbf{x}^k - \mathbf{u}).$$

- *Concave-convex procedure (CCCP)* [Yuille and Rangarajan, 2002]

$$-\nabla f(\mathbf{x}^k) \in \partial g(\mathbf{x}^{k+1}).$$

$\rightarrow$ CCCP-based CRF algorithms [Desmaison et al., 2016, Krähenbühl and Koltun, 2013] are instances of regularized Frank-Wolfe.

Instantiations of regularized Frank-Wolfe

Multiple existing algorithms are also special cases!

- *Parallel mean field* [Krähenbühl and Koltun, 2011]

$$\mathbf{x}^{k+1} = \mathrm{softmax}(-\mathbf{P}\mathbf{x}^k - \mathbf{u}).$$

- *Concave-convex procedure (CCCP)* [Yuille and Rangarajan, 2002]

$$-\nabla f(\mathbf{x}^k) \in \partial g(\mathbf{x}^{k+1}).$$

  $\rightarrow$ CCCP-based CRF algorithms [Desmaison et al., 2016, Krähenbühl and Koltun, 2013] are instances of regularized Frank-Wolfe.

- *Vanilla Frank-Wolfe:* Existing algorithms [Sontag and Jaakkola, 2007, Meshi et al., 2015, Tang et al., 2016, Desmaison et al., 2016, Lê-Huu and Paragios, 2018] are instances of regularized Frank-Wolfe.

Convergence analysis

Assumptions:

- f *differentiable* and L_f*-semi-concave* ($L_f \geq 0$).
- g *proper*, *closed*, and σ_g*-strongly-convex* ($\sigma_g \geq 0$).

Convergence analysis

Assumptions:

- f *differentiable* and *L_f-semi-concave* ($L_f \geq 0$).
- g *proper*, *closed*, and *σ_g-strongly-convex* ($\sigma_g \geq 0$).

Main results: Upper bound on *conditional gradient norm* [Beck, 2017].

Convergence analysis

Assumptions:

- f *differentiable* and *L_f-semi-concave* ($L_f \geq 0$).
- g *proper*, *closed*, and *σ_g-strongly-convex* ($\sigma_g \geq 0$).

Main results: Upper bound on *conditional gradient norm* [Beck, 2017].

	constant stepsize $\alpha_k = \alpha > 0 \ \forall k$	constant step length $\alpha_k = \frac{\alpha}{\lVert \mathbf{p}^k - \mathbf{x}^k \rVert} \ \forall k$	non-summable $\sum_{k=0}^{+\infty} \alpha_k = \infty$	adaptive or line search
convex g	$\frac{\Delta_0}{\alpha(k+1)} + \frac{L_f \Omega^2 \alpha}{2}$	$\frac{\Delta_0 \Omega}{\alpha(k+1)} + \frac{L_f \Omega \alpha}{2}$	$\frac{\Delta_0 + \frac{L_f \Omega^2}{2} \sum_{i=0}^k \alpha_i^2}{\sum_{i=0}^k \alpha_i}$	$\max\left(\frac{2\Delta_0}{k+1}, \frac{\mu \Omega}{\sqrt{k+1}}\right)$
strongly convex g	$\frac{\Delta_0}{\alpha(k+1)} + \eta(\alpha)\Omega^2 \ \forall \alpha \geq 2\omega$ $\frac{\Delta_0}{\rho(\alpha)(k+1)} \ \forall \alpha < 2\omega$	$\left(\frac{\Delta_0}{\alpha\sqrt{2\sigma_g}(k+1)} + \frac{(L_f + \sigma_g)\alpha}{2\sqrt{2\sigma_g}}\right)^2$	$\frac{\Delta_{k(\omega)}}{\sum_{i=k(\omega)}^k \alpha_i}$	$\frac{\Delta_0}{\omega(k+1)}$
concave f	$\frac{\Delta_0}{\alpha(k+1)}$	$\frac{\Delta_0 \Omega}{\alpha(k+1)}$	$\frac{\Delta_0}{\sum_{i=0}^k \alpha_i}$	$\frac{2\Delta_0}{k+1}$

Convergence analysis

Assumptions:

- f *differentiable* and *L_f-semi-concave* ($L_f \geq 0$).
- g *proper*, *closed*, and *σ_g-strongly-convex* ($\sigma_g \geq 0$).

Main results: Upper bound on *conditional gradient norm* [Beck, 2017].

	constant stepsize $\alpha_k = \alpha > 0\ \forall k$	constant step length $\alpha_k = \frac{\alpha}{\lVert \mathbf{p}^k - \mathbf{x}^k \rVert}\ \forall k$	non-summable $\sum_{k=0}^{+\infty} \alpha_k = \infty$	adaptive or line search
convex g	$\frac{\Delta_0}{\alpha(k+1)} + \frac{L_f \Omega^2 \alpha}{2}$	$\frac{\Delta_0 \Omega}{\alpha(k+1)} + \frac{L_f \Omega \alpha}{2}$	$\frac{\Delta_0 + \frac{L_f \Omega^2}{2} \sum_{i=0}^{k} \alpha_i^2}{\sum_{i=0}^{k} \alpha_i}$	$\max\left(\frac{2\Delta_0}{k+1}, \frac{\mu \Omega}{\sqrt{k+1}}\right)$
strongly convex g	$\frac{\Delta_0}{\alpha(k+1)} + \eta(\alpha)\Omega^2\ \forall \alpha \geq 2\omega$ $\frac{\Delta_0}{\rho(\alpha)(k+1)}\ \forall \alpha < 2\omega$	$\left(\frac{\Delta_0}{\alpha\sqrt{2\sigma_g}(k+1)} + \frac{(L_f+\sigma_g)\alpha}{2\sqrt{2\sigma_g}}\right)^2$	$\frac{\Delta_{k(\omega)}}{\sum_{i=k(\omega)}^{k} \alpha_i}$	$\frac{\Delta_0}{\omega(k+1)}$
concave f	$\frac{\Delta_0}{\alpha(k+1)}$	$\frac{\Delta_0 \Omega}{\alpha(k+1)}$	$\frac{\Delta_0}{\sum_{i=0}^{k} \alpha_i}$	$\frac{2\Delta_0}{k+1}$

- Best rate of convergence: $\mathcal{O}(1/k)$.

Convergence analysis

Assumptions:

- f *differentiable* and L_f*-semi-concave* ($L_f \geq 0$).
- g *proper*, *closed*, and σ_g*-strongly-convex* ($\sigma_g \geq 0$).

Main results: Upper bound on *conditional gradient norm* [Beck, 2017].

	constant stepsize $\alpha_k = \alpha > 0 \; \forall k$	constant step length $\alpha_k = \frac{\alpha}{\lVert \mathbf{p}^k - \mathbf{x}^k \rVert} \; \forall k$	non-summable $\sum_{k=0}^{+\infty} \alpha_k = \infty$	adaptive or line search
convex g	$\frac{\Delta_0}{\alpha(k+1)} + \frac{L_f \Omega^2 \alpha}{2}$	$\frac{\Delta_0 \Omega}{\alpha(k+1)} + \frac{L_f \Omega \alpha}{2}$	$\frac{\Delta_0 + \frac{L_f \Omega^2}{2} \sum_{i=0}^{k} \alpha_i^2}{\sum_{i=0}^{k} \alpha_i}$	$\max\left(\frac{2\Delta_0}{k+1}, \frac{\mu\Omega}{\sqrt{k+1}}\right)$
strongly convex g	$\frac{\Delta_0}{\alpha(k+1)} + \eta(\alpha)\Omega^2 \; \forall \alpha \geq 2\omega$ $\frac{\Delta_0}{\rho(\alpha)(k+1)} \quad \forall \alpha < 2\omega$	$\left(\frac{\Delta_0}{\alpha\sqrt{2\sigma_g(k+1)}} + \frac{(L_f + \sigma_g)\alpha}{2\sqrt{2\sigma_g}}\right)^2$	$\frac{\Delta_{k(\omega)}}{\sum_{i=k(\omega)}^{k} \alpha_i}$	$\frac{\Delta_0}{\omega(k+1)}$
concave f	$\frac{\Delta_0}{\alpha(k+1)}$	$\frac{\Delta_0 \Omega}{\alpha(k+1)}$	$\frac{\Delta_0}{\sum_{i=0}^{k} \alpha_i}$	$\frac{2\Delta_0}{k+1}$

- Best rate of convergence: $\mathcal{O}(1/k)$.
- Byproduct: *convergent parallel mean field* variants.

Tightness analysis

We were solving a *discrete* optimization problem through a (regularized) *continuous relaxation*. **How good could the final discrete solution be?**

Tightness analysis

We were solving a *discrete* optimization problem through a (regularized) *continuous relaxation*. **How good could the final discrete solution be?**

Main results: *Upper bound on energy*.

Tightness analysis

We were solving a *discrete* optimization problem through a (regularized) *continuous relaxation*. **How good could the final discrete solution be?**

Main results: *Upper bound on energy*.

$$E(\bar{\mathbf{x}}_r^*) \leq E^* + M - m + C,$$

where:

- $\bar{\mathbf{x}}_r^*$: discrete solution rounded from $\text{argmin}_{\mathbf{x}\in\mathcal{X}} \{E(\mathbf{x}) + r(\mathbf{x})\}$.
- E^*: minimum discrete energy.
- m, M: lower and upper bounds of r on $\mathcal{X}$.
- C: constant depending on selected rounding scheme.

Tightness analysis

We were solving a *discrete* optimization problem through a (regularized) *continuous relaxation*. **How good could the final discrete solution be?**

Main results: *Upper bound on energy*.

$$E(\bar{\mathbf{x}}_r^*) \leq E^* + M - m + C,$$

where:

- $\bar{\mathbf{x}}_r^*$: discrete solution rounded from $\operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \{E(\mathbf{x}) + r(\mathbf{x})\}$.
- E^*: minimum discrete energy.
- m, M: lower and upper bounds of r on $\mathcal{X}$.
- C: constant depending on selected rounding scheme.

→ *Recovering previous results as special cases [Berthod, 1982, Ravikumar and Lafferty, 2006, Lê-Huu and Paragios, 2018].*

Experiments: Models and datasets

- **Task:** Semantic image segmentation.

Experiments: Models and datasets

- **Task:** Semantic image segmentation.
- **Datasets:** *PASCAL VOC* and *Cityscapes*.

Experiments: Models and datasets

- **Task:** Semantic image segmentation.
- **Datasets:** *PASCAL VOC* and *Cityscapes*.
- **Models:** Standard *CNN-CRF* with Gaussian potentials [Krähenbühl and Koltun, 2011, Zheng et al., 2015]. Use *DeepLabv3* [Chen et al., 2017] and *DeepLabv3+* [Chen et al., 2018] for CNN.

Experiments: Methods

Euclidean Frank-Wolfe (ℓ_2**FW**) and *Entropic Frank-Wolfe* (e**FW**) against:

Experiments: Methods

Euclidean Frank-Wolfe (ℓ_2**FW**) and *Entropic Frank-Wolfe* (e**FW**) against:

- Mean field (**MF**) [Krähenbühl and Koltun, 2011, 2013] (main baseline).

Experiments: Methods

Euclidean Frank-Wolfe (ℓ_2**FW**) and *Entropic Frank-Wolfe* (e**FW**) against:

- Mean field (**MF**) [Krähenbühl and Koltun, 2011, 2013] (main baseline).
- Nonconvex vanilla Frank-Wolfe (**FW**) [Lê-Huu and Paragios, 2018].

Experiments: Methods

Euclidean Frank-Wolfe (ℓ_2**FW**) and *Entropic Frank-Wolfe* (e**FW**) against:

- Mean field (**MF**) [Krähenbühl and Koltun, 2011, 2013] (main baseline).
- Nonconvex vanilla Frank-Wolfe (**FW**) [Lê-Huu and Paragios, 2018].
- Projected gradient (**PGD**) [Larsson et al., 2017, Lê-Huu and Paragios, 2018].

Experiments: Methods

Euclidean Frank-Wolfe (ℓ_2**FW**) and *Entropic Frank-Wolfe* (e**FW**) against:

- Mean field (**MF**) [Krähenbühl and Koltun, 2011, 2013] (main baseline).
- Nonconvex vanilla Frank-Wolfe (**FW**) [Lê-Huu and Paragios, 2018].
- Projected gradient (**PGD**) [Larsson et al., 2017, Lê-Huu and Paragios, 2018].
- Fast proximal gradient method (**PGM**) [Beck and Teboulle, 2009].

Experiments: Methods

Euclidean Frank-Wolfe (ℓ_2**FW**) and *Entropic Frank-Wolfe* (e**FW**) against:

- Mean field (**MF**) [Krähenbühl and Koltun, 2011, 2013] (main baseline).
- Nonconvex vanilla Frank-Wolfe (**FW**) [Lê-Huu and Paragios, 2018].
- Projected gradient (**PGD**) [Larsson et al., 2017, Lê-Huu and Paragios, 2018].
- Fast proximal gradient method (**PGM**) [Beck and Teboulle, 2009].
- Alternating direction method of multipliers (**ADMM**) [Lê-Huu and Paragios, 2017, 2018].

Experiments: Methods

Euclidean Frank-Wolfe (ℓ_2**FW**) and *Entropic Frank-Wolfe* (*e***FW**) against:

- Mean field (**MF**) [Krähenbühl and Koltun, 2011, 2013] (main baseline).
- Nonconvex vanilla Frank-Wolfe (**FW**) [Lê-Huu and Paragios, 2018].
- Projected gradient (**PGD**) [Larsson et al., 2017, Lê-Huu and Paragios, 2018].
- Fast proximal gradient method (**PGM**) [Beck and Teboulle, 2009].
- Alternating direction method of multipliers (**ADMM**) [Lê-Huu and Paragios, 2017, 2018].

Exclusion due to poor performance:

- Convex vanilla Frank-Wolfe [Desmaison et al., 2016].
- Entropic mirror descent [Nemirovskij and Yudin, 1983, Beck and Teboulle, 2003].

Experiments: Inference performance

No CRF learning in this experiment!

Experiments: Inference performance

No CRF learning in this experiment!

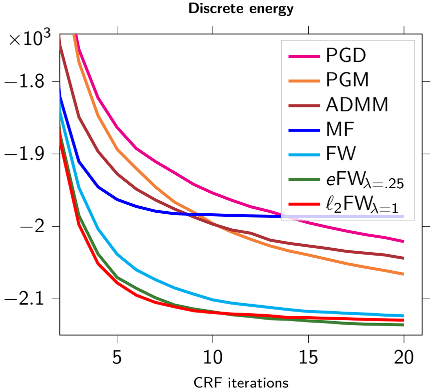
- Use pre-trained DeepLabv3 and DeepLabv3+.
- Use *Potts model* for CRF.

Experiments: Inference performance

No CRF learning in this experiment!

- Use pre-trained DeepLabv3 and DeepLabv3+.
- Use *Potts model* for CRF.

Average discrete energy on PASCAL VOC validation set:

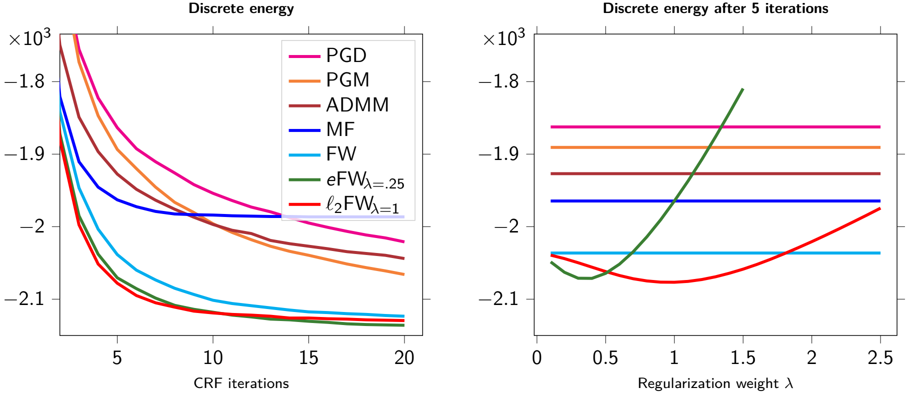


Experiments: Inference performance

No CRF learning in this experiment!

- Use pre-trained DeepLabv3 and DeepLabv3+.
- Use *Potts model* for CRF.

Average discrete energy on PASCAL VOC validation set:



Experiments: Inference performance

Validation mIoU using Potts dense CRF on top of pre-trained CNN

		CNN	PGD	PGM	ADMM	MF	FW	*e*$\mathrm{FW}_{.7}$	*e*$\mathrm{FW}_{.3}$	ℓ_2FW
VOC	DeepLabv3	81.83	82.23	82.23	82.22	82.21	82.27	82.26	**82.29**	**82.29**
	DeepLabv3+	82.89	83.36	83.37	83.38	83.45	83.43	83.45	83.48	**83.50**
CITY	DeepLabv3	76.73	76.88	76.86	76.95	76.97	76.86	76.99	76.99	**77.03**
	DeepLabv3+	79.55	79.64	79.63	**79.66**	79.63	79.64	79.65	**79.66**	**79.66**

Experiments: Inference performance

Validation mIoU using Potts dense CRF on top of pre-trained CNN

		CNN	PGD	PGM	ADMM	MF	FW	eFW$_{.7}$	eFW$_{.3}$	ℓ_2FW
VOC	DeepLabv3	81.83	82.23	82.23	82.22	82.21	82.27	82.26	**82.29**	**82.29**
VOC	DeepLabv3+	82.89	83.36	83.37	83.38	83.45	83.43	83.45	83.48	**83.50**
CITY	DeepLabv3	76.73	76.88	76.86	76.95	76.97	76.86	76.99	76.99	**77.03**
CITY	DeepLabv3+	79.55	79.64	79.63	**79.66**	79.63	79.64	79.65	**79.66**	**79.66**

- Improvement of 0.1–0.6% by CRF over CNN.
- Similar performance between CRF solvers, *ℓ_2FW consistently best*.

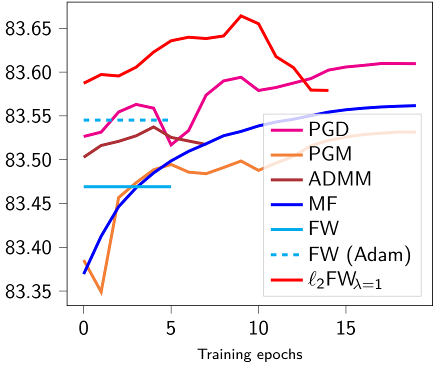
Experiments: Learning performance

Joint training of CNN and CRF in this experiment!

Experiments: Learning performance

Joint training of CNN and CRF in this experiment!

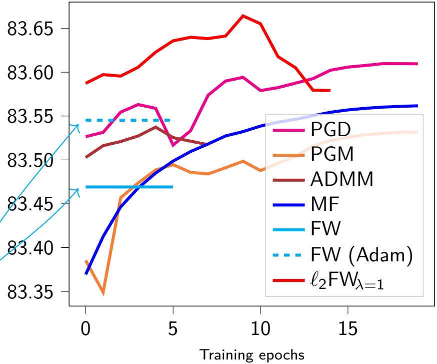
Validation mIoU on PASCAL VOC



Experiments: Learning performance

Joint training of CNN and CRF in this experiment!

Validation mIoU on PASCAL VOC



Vanilla FW fails to learn
(zero-gradient issue)

Experiments: Learning performance

Validation mIoU under **joint training**

		CNN	PGD	PGM	ADMM	MF	$eFW_{.7}$	$eFW_{.3}$	ℓ_2FW
VOC	DeepLabv3	81.83	83.69 ±0.20	**83.75** ±0.23	83.68 ±0.06	83.69 ±0.10	83.50 ±0.10	83.25 ±0.20	**83.75** ±0.13
	DeepLabv3+	82.89	84.82 ±0.23	84.79 ±0.20	84.83 ±0.06	84.87 ±0.17	84.64 ±0.23	84.50 ±0.16	**85.14** ±0.09
CITY	DeepLabv3+	79.55	79.80	79.62	79.62	79.74	79.70	79.58	**79.95**

Experiments: Learning performance

Validation mIoU under **joint training**

		CNN	PGD	PGM	ADMM	MF	$eFW_{.7}$	$eFW_{.3}$	ℓ_2FW
VOC	DeepLabv3	81.83	83.69 ±0.20	**83.75** ±0.23	83.68 ±0.06	83.69 ±0.10	83.50 ±0.10	83.25 ±0.20	**83.75** ±0.13
	DeepLabv3+	82.89	84.82 ±0.23	84.79 ±0.20	84.83 ±0.06	84.87 ±0.17	84.64 ±0.23	84.50 ±0.16	**85.14** ±0.09
CITY	DeepLabv3+	79.55	79.80	79.62	79.62	79.74	79.70	79.58	**79.95**

- *Joint training yields larger improvements* by CRF over CNN: 1.9–2.3% on PASCAL VOC, 0.4% on Cityscapes.
- Again, *ℓ_2FW consistently best*.

Conclusion

- Regularized Frank-Wolfe: General MAP inference method.

Conclusion

- Regularized Frank-Wolfe: General MAP inference method.
- This generalized perspective allows a unified analysis of many new and existing algorithms.

Conclusion

- Regularized Frank-Wolfe: General MAP inference method.
- This generalized perspective allows a unified analysis of many new and existing algorithms.
- ℓ_2FW and eFW are two strong instances.

Conclusion

- Regularized Frank-Wolfe: General MAP inference method.
- This generalized perspective allows a unified analysis of many new and existing algorithms.
- ℓ_2FW and eFW are two strong instances.
- Dense CRFs could still be relevant for semantic segmentation.

Conclusion

- Regularized Frank-Wolfe: General MAP inference method.
- This generalized perspective allows a unified analysis of many new and existing algorithms.
- ℓ_2FW and *e*FW are two strong instances.
- Dense CRFs could still be relevant for semantic segmentation.

Thank you for your attention!

Please read our paper for more details.

Code available at https://github.com/netw0rkf10w/CRF.